DataScientest

CO2 Car Emission Prediction

Final Report

Start Date: 05/06/2024

by

Azangue Pavel: pavelazangue@gmail.com
Abd Akdim: pavelazangue@gmail.com
Halimeh: pavelazangue@gmail.com
email here

Under the supervision of

Sarah Lenet:

school
DataScientest

Contents

1 Introduction

The impacts of climate change have become a global focal point in recent decades. A significant contributor to greenhouse gas emissions, which drive climate change, is CO2 emissions from the transportation sector. Cars, in particular, are a major source of CO2 emissions. Therefore, understanding the factors that influence vehicle CO2 emissions and developing methods to predict and ultimately reduce these emissions is crucial.

Predicting CO2 emissions is critical for shaping environmental strategies and policy measures. Governments and regulatory bodies can use such models to set stricter emission standards and promote the development of low-emission vehicles. Furthermore, consumers who are aware of the environmental impact of their vehicle choices can make more informed decisions.

The objective of this project is to develop a model to predict the CO2 emissions of cars based on various vehicle characteristics such as engine size, weight, fuel type, and other relevant factors. Such a model can not only help assess the environmental impact of the transportation sector but also assist policymakers and manufacturers in developing more environmentally friendly vehicles.

In this project, various machine learning techniques will be used to predict the CO2 emissions of cars. The data source consists of publicly available datasets containing information about various vehicle parameters and their CO2 emissions. Modeling techniques include both linear and non-linear methods to capture the relationships between vehicle characteristics and CO2 emissions.

2 Methodology

The methodology followed for this project is the following:

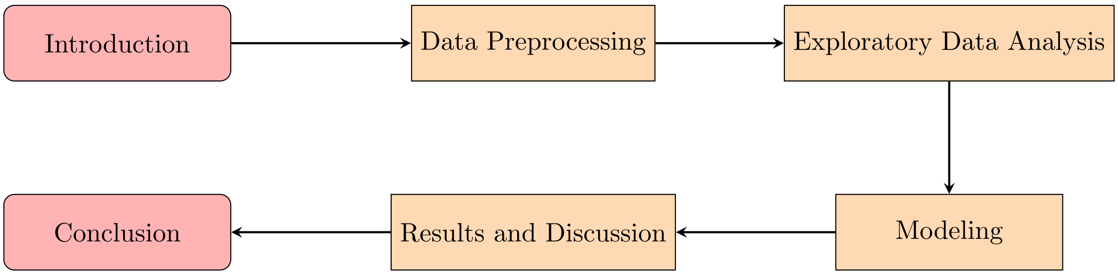

Figure 1: Workflow of the CO2 Emission Prediction by vehicles Project

3 Data Preprocessing

The dataset provides a comprehensive overview of the technical characteristics and environmental impact of vehicles marketed in France in 2013. Analyzing this data can help identify trends, patterns, and insights related to vehicle emissions, fuel efficiency, and other factors influencing air quality and environmental sustainability.

3.1 Initial Data Exploration

This section provides an overview of the dataset.

- **Size:** The dataset consists of 44,850 entries and 26 columns.
- **Columns:** It includes a range of attributes, including: (see Table 1).

Table 1: Overview of Dataset Attributes

Column	**Description**
Marque	The brand or manufacturer of the vehicle.
Modèle dossier	Model name assigned by UTAC (Union Technique de l'Automobile du Motocycle et du Cycle), a French organization responsible for vehicle approval.
Modèle UTAC	Model name assigned by UTAC.
Désignation commerciale	Commercial designation of the vehicle.
CNIT	Identification number assigned by the French government for each vehicle model.
TVV	Type, variant, and version of the vehicle.
Carburant	Type of fuel used by the vehicle (e.g., gasoline, diesel, hybrid).
Hybride	Indicates whether the vehicle is hybrid (yes/no).
Puissance administrative	Administrative horsepower of the vehicle.
Puissance maximale (kW)	Maximum power of the vehicle in kilowatts.
Boîte de vitesse	Type of transmission (e.g., manual, automatic).
Consommation urbaine (l/100km)	Urban fuel consumption in liters per 100 kilometers.
Consommation extra-urbaine (l/100km)	Extra-urban fuel consumption in liters per 100 kilometers.
Consommation mixte (l/100km)	Mixed fuel consumption in liters per 100 kilometers.
CO2 (g/km)	CO2 emissions of the vehicle in grams per kilometer.
CO type I (g/km)	CO emissions of type I (e.g., carbon monoxide) in grams per kilometer.
HC (g/km)	Hydrocarbon emissions in grams per kilometer.
NOX (g/km)	Nitrogen oxide emissions in grams per kilometer.
HC+NOX (g/km)	Combined hydrocarbon and nitrogen oxide emissions in grams per kilometer.
Particules (g/km)	Particulate emissions in grams per kilometer.
Masse vide euro min (kg)	Minimum empty weight of the vehicle in kilograms according to Euro standards.
Masse vide euro max (kg)	Maximum empty weight of the vehicle in kilograms according to Euro standards.
Champ V9	Additional field.
Date de mise à jour	Date of the last update for the vehicle.
Carrosserie	Body type of the vehicle (e.g., sedan, hatchback, SUV).
Gamme	Range or series of the vehicle.

3.2 Data Types

The dataset contains a mix of data types including object, int64, and float64. Figure 1 shows the type of each column.

data_types.png

Figure 2: An overview of data types in the dataset

3.3 Missing Values

Some columns have missing values such as Consommation urbaine, Consommation extra-urbaine, CO2 emissions, and others. Figure 2 shows the amount of missing values for each column.

3.4 Hybrid Vehicles

The dataset indicates whether a vehicle is hybrid (Hybride), with most vehicles being non-hybrid.

3.5 Last Updated

The dataset includes a column indicating the date of the last update for each vehicle.

Figure 3: Percentage of missing values in the dataset

3.6 Target Variable

One of the potential target variables for analysis could be CO2 emissions (CO2 (g/km)), which is essential for assessing the environmental impact of vehicles.

4 Potential Challenges

This section presents the distribution of values for each attribute and potential outliers.

4.1 Numerical Variables

4.1.1 Puissance administrative:

Statistics of this attribute are:

- Count: 44,850
- Mean: 11.018997
- Std: 5.554475
- Min: 1.000000

- 25%: 9.000000
- 50%: 10.000000
- 75%: 11.000000
- Max: 81.000000

Figure 3 shows the distribution of the data for this attribute.
There are several outliers that need to be handled. The calculation of outliers shows that:

- Number of outliers: 7,813
- Lower whisker: 6.0
- Upper whisker: 14.0

4.1.2 Puissance maximale:

Statistics of this attribute are:

- Count: 44,850
- Mean: 124.780834
- Std: 49.158804
- Min: 10.000000
- 25%: 100.000000
- 50%: 120.000000
- 75%: 125.000000
- Max: 559.300000

Figure 4 shows the distribution of the data for this attribute.
There are 7,259 data points that are outliers. We need to select an appropriate method to handle these outliers.

4.1.3 Consommation urbaine:

Statistics of this attribute are:

- Count: 44,808
- Mean: 9.706744
- Std: 2.366181
- Min: 0.000000
- 25%: 8.800000
- 50%: 9.800000
- 75%: 10.700000
- Max: 41.100000

There are 44,808 non-null values out of a total of 44,850 entries in the dataset, suggesting some missing values. Furthermore, the min and max values show that there may be some erroneous or missing data entries. Figure 5 shows the distribution of data in this column. We identified 4,210 data points that are outliers. We need to handle these outliers.

consommation_urbaine.png

Figure 4: Distribution of Urban fuel consumption in liters per 100 kilometers

4.1.4 Consommation extra-urbaine:

Statistics for this attribute are:

- Count: 44,808
- Mean: 6.567634
- Std: 1.196234
- Min: 2.800000
- 25%: 6.300000
- 50%: 6.700000
- 75%: 7.100000
- Max: 14.900000

There are 44,808 non-null values out of a total of 44,850 entries in the dataset, suggesting some missing values. Figure 6 shows some representations of this attribute. Analysis of the dataset shows that there are 8,546 outlier or extreme values. We need to find a suitable approach to manage them.

consommation_extra_urbaine.png

Figure 5: Distribution of Extra-Urban fuel consumption

4.1.5 Consommation mixte:

Statistics of this attribute are:

- Count: 44,811
- Mean: 7.716254
- Std: 1.597110
- Min: 1.200000
- 25%: 7.200000
- 50%: 7.700000
- 75%: 8.400000
- Max: 24.500000

There are 44,811 non-null values out of a total of 44,850 entries in the dataset, suggesting some missing values. Figure 7 shows the distribution of the data for this attribute and also the boxplot to analyze the outliers or extreme values. Analysis of data shows that there are 6,254 outlier or extreme values. We need to find a suitable approach to manage them.

Figure 6: Representation of Combined Fuel Consumption

4.1.6 CO2:

Statistics of this attribute are:

- Count: 44,811
- Mean: 198.910892
- Std: 39.014678
- Min: 27.000000
- 25%: 187.000000
- 50%: 203.000000
- 75%: 221.000000
- Max: 572.000000

There are 44,811 non-null values out of a total of 44,850 entries in the dataset, suggesting some missing values. Figure 8 shows two representations of data. Analysis of data shows that there are 5,458 outlier or extreme values. We need to find a suitable approach to manage them.

co2.png

Figure 7: Representations of CO2 emissions

4.1.7 CO type I (g/km):

Statistics of this attribute are:

- Count: 44,547
- Mean: 0.153461
- Std: 0.138984
- Min: 0.005000
- 25%: 0.046000
- 50%: 0.093000
- 75%: 0.222000
- Max: 0.968000

There are 44,547 non-null values out of a total of 44,850 entries in the dataset, suggesting some missing values. Figure 9 shows two representations of this attribute. The boxplot shows that most of the data are concentrated on the right side after the median line inside of the box. It suggests that the distribution of the data is positively skewed. Furthermore, outliers are concentrated on the right side of the boxplot. We have these statistics for outliers:

- Number of outliers: 1,009
- Lower whisker: -0.21800000000000003
- Upper whisker: 0.486

There are 303 NaN values in this column and therefore it affects the calculation of outliers. We need to handle these missing values.

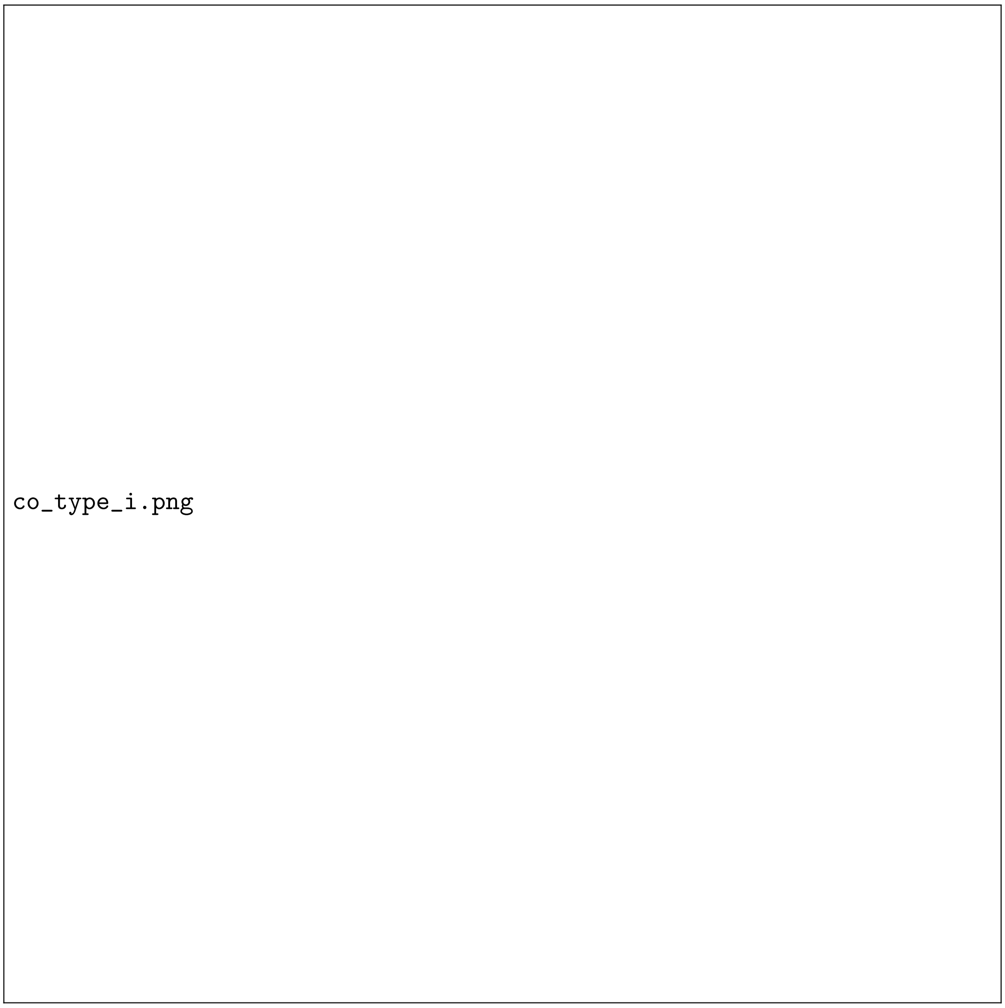

Figure 8: Distribution of CO type 1 emissions

4.1.8 HC (g/km):

Statistics of this attribute are:

- Count: 10,403
- Mean: 0.030499
- Std: 0.018408
- Min: 0.008000

- 25%: 0.008000
- 50%: 0.031000
- 75%: 0.044000
- Max: 0.143000

As we can see there are too many missing values in this column (34,447). We need to decide how can we manage this issue. Figure 10 shows the distribution of the data. As we have a bunch of missing values, the results for the calculation of outliers and lower and upper whiskers are:

- Number of outliers: 4
- Lower whisker: -0.045999999999999999
- Upper whisker: 0.097999999999999999

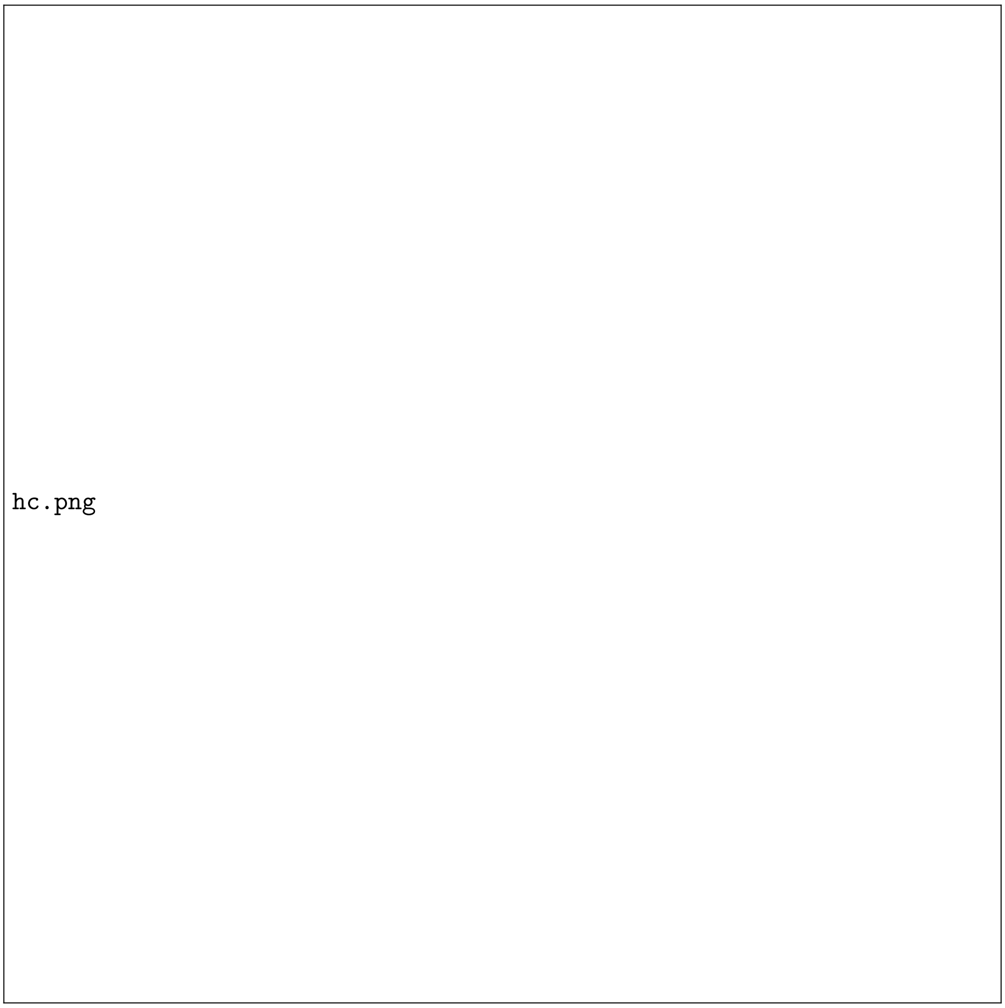

Figure 9: Distribution of Hydrocarbon emissions

4.1.9 NOX (g/km):

Statistics of this attribute are:

- Count: 44,547
- Mean: 0.311837
- Std: 0.463112
- Min: 0.001000
- 25%: 0.158000
- 50%: 0.197000
- 75%: 0.228000
- Max: 1.846000

This column has 303 missing values. Figure 11 shows the distribution of the data for this attribute. We calculated the number of outliers for this attribute:

- Number of outliers: 10,426
- Lower whisker: 0.05299999999999999
- Upper whisker: 0.333

Therefore, we need to handle these outliers.

4.1.10 HC+NOX (g/km):

Statistics of this attribute are:

- Count: 34,191
- Mean: 0.224788
- Std: 0.041681
- Min: 0.038000
- 25%: 0.201000
- 50%: 0.220000
- 75%: 0.248000
- Max: 0.306000

There are 10,659 missing values in this column and we need to decide about the approach to handle them. Figure 12 shows the distribution of the data in this column. The title of this column says that this attribute is the combination of two previous columns. We need to examine whether they are directly concluded from the two previous columns or they are new data. We calculated the number of outliers:

- Number of outliers: 883
- Lower whisker: 0.13050000000000003
- Upper whisker: 0.3185

Therefore, we need to decide on which approach is suitable to handle these outliers.

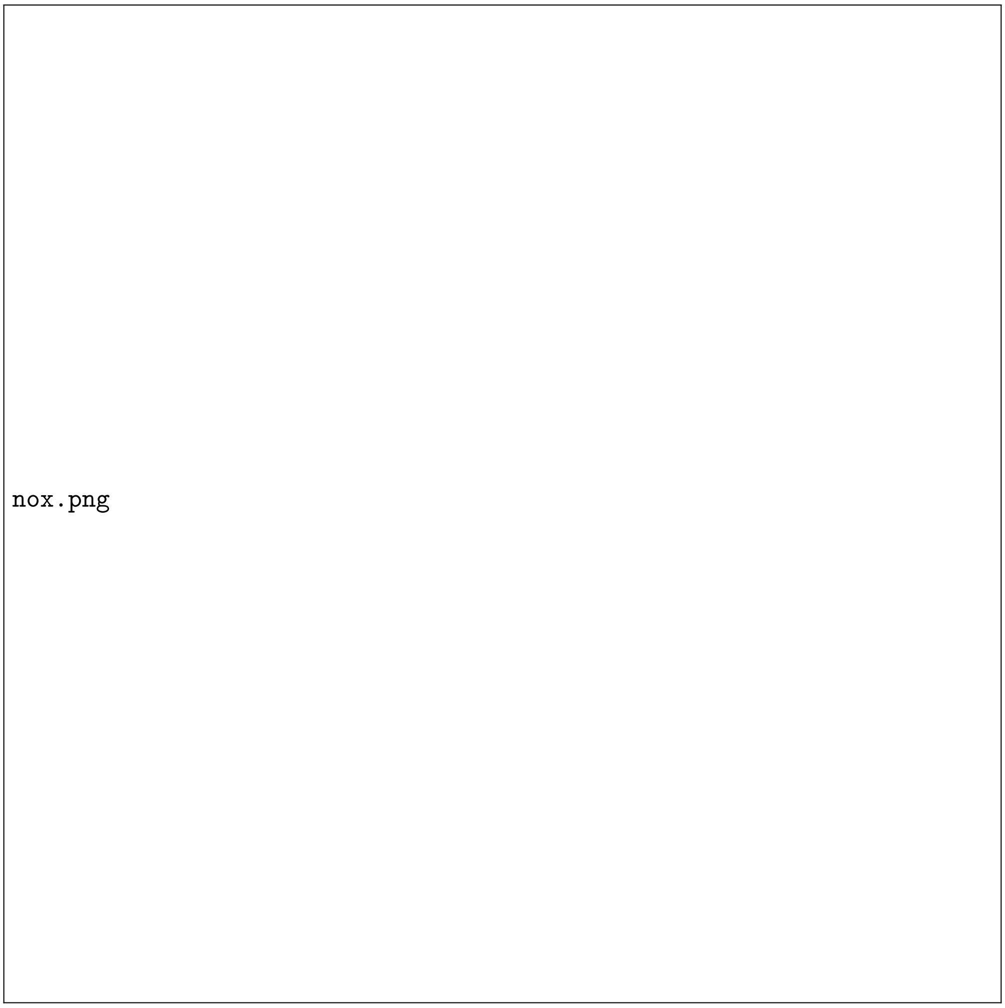

Figure 10: Distribution of data for Nitrogen oxide emissions

4.1.11 Particules (g/km):

Statistics of this attribute are:

- Count: 41,708
- Mean: 0.000961
- Std: 0.006469
- Min: 0.000000
- 25%: 0.000000
- 50%: 0.001000
- 75%: 0.001000
- Max: 0.610000

There are 3,142 missing values in this column. Figure 13 shows the distribution of data in this column. This figure shows that there are some anomalies in the data. This is the results of calculation for outliers:

- Number of outliers: 3,696

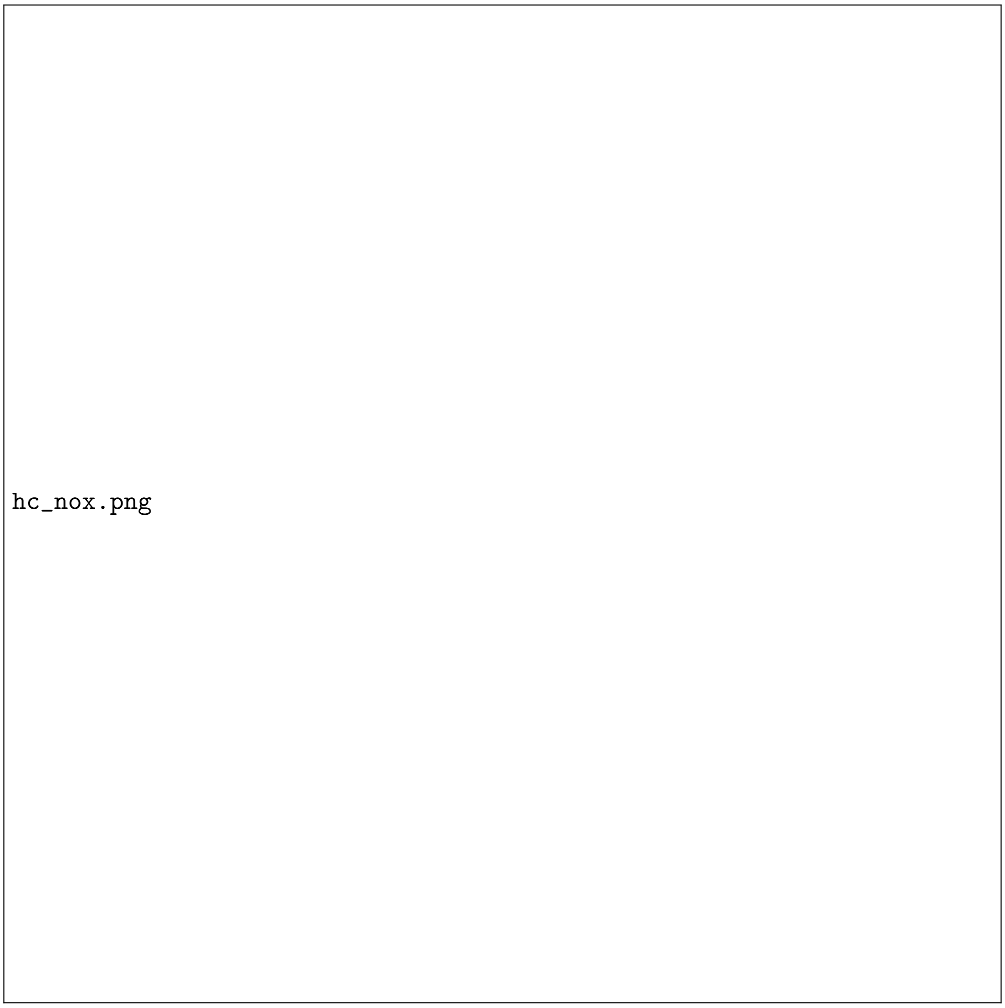

Figure 11: Distribution of Combined hydrocarbon and nitrogen oxide emissions

- Lower whisker: -0.0015
- Upper whisker: 0.0025

We need to decide how can we handle these anomalies.

4.1.12 Masse vide euro min (kg):

Statistics of this attribute are:

- Count: 44,850
- Mean: 2,070.961650
- Std: 342.872975
- Min: 825.000000
- 25%: 1,976.000000
- 50%: 2,076.000000
- 75%: 2,256.000000

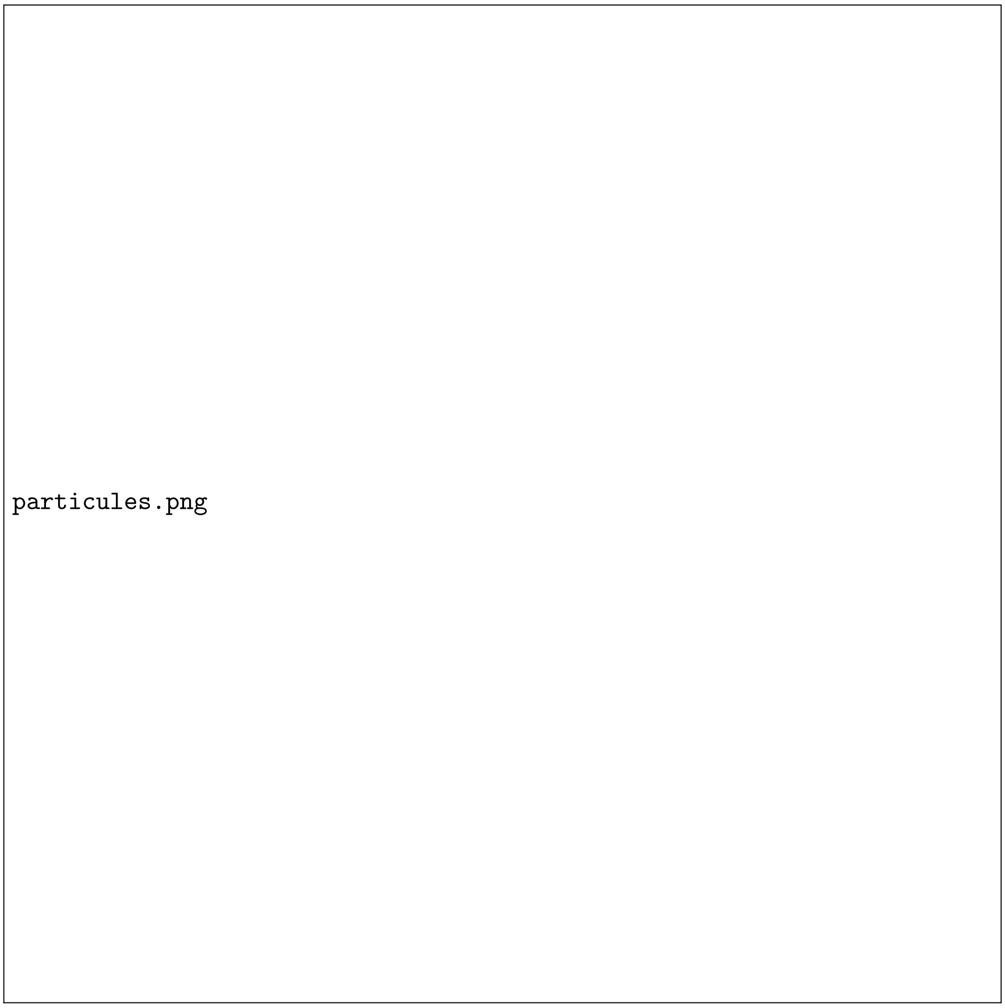

Figure 12: Distribution of Particulate emissions

- Max: 3,115.000000

There is no missing value in this column. Figure 14 shows the distribution of data. This is the results of calculations for outliers; we need to manage them.

- Number of outliers: 4,397
- Lower whisker: 1,556.0
- Upper whisker: 2,676.0

4.1.13 Masse vide euro max (kg):

Statistics of this attribute are:

- Count: 44,850
- Mean: 2,169.545284
- Std: 410.600541
- Min: 825.000000

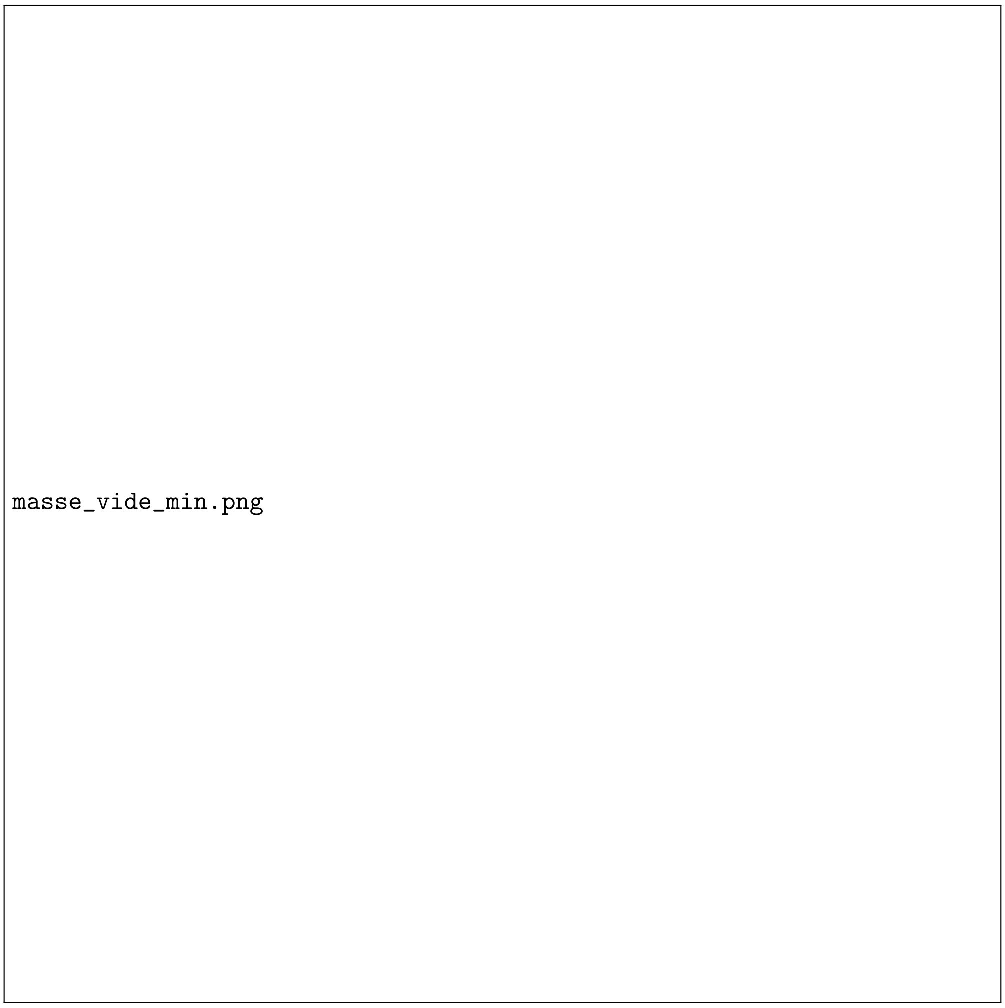

Figure 13: Distribution of Minimum empty weight of the vehicle

- 25%: 2,043.500000
- 50%: 2,185.000000
- 75%: 2,355.000000
- Max: 3,115.000000

There is no missing value in this column. Figure 15 shows the distribution of data. As we can see in the boxplot, there are many outliers that we need to handle:

- Number of outliers: 8,240
- Lower whisker: 1,576.25
- Upper whisker: 2,822.25

4.2 Categorical Variables

There are 13 categorical attributes in the dataset. The number of categories for each variable are:

- Marque: 51

`masse_vide_max.png`

Figure 14: Distribution of Maximum empty weight of the vehicle

- Modèle dossier: 458
- Modèle UTAC: 419
- Désignation commerciale: 3,582
- CNIT: 44,191
- Type Variante Version (TVV): 28,781
- Carburant: 13
- Hybride: 2
- Boîte de vitesse: 16
- Champ V9: 13
- Date de mise à jour: 3
- Carrosserie: 10
- Gamme: 7

As we can see, there are variables with so many categories. When preparing data for machine learning models, categorical variables need to be transformed into numerical representations. There are several approaches to handle categorical variables with many categories including One-Hot Encoding, Frequency Encoding, and Target Encoding. We need to decide which approach is suitable for encoding the mentioned categorical variables.

5 Data Visualization

In this section, five representations of the data are presented.

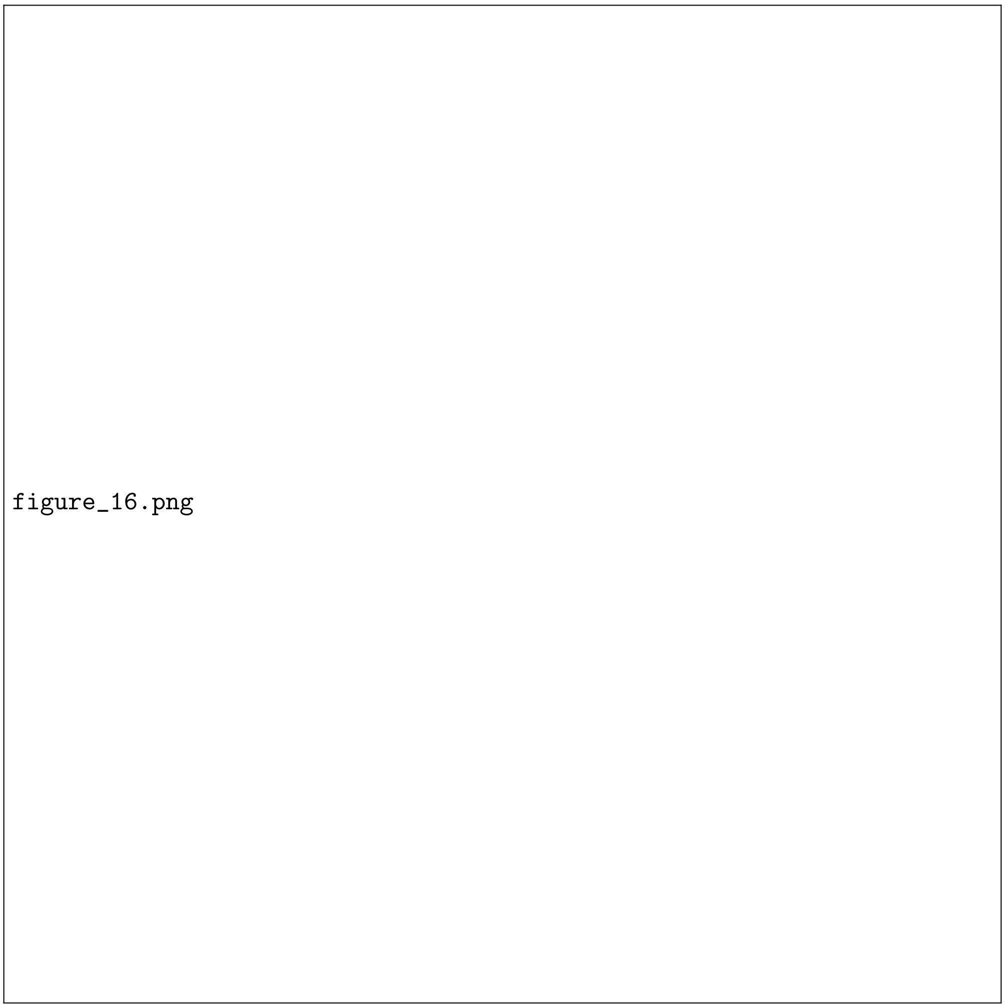

Figure 15: Relationship between Consommation Mixte and CO2 emissions

The scatterplot illustrates a clear positive correlation between fuel consumption ('Consommation mixte') and CO2 emissions. The main trendline slopes from the bottom left to the top right, indicating that vehicles with higher fuel consumption tend to have higher CO2 emissions. There are two additional lines observed at approximately 70 l/100 km. One line runs below the main trendline up to around 300 l/100 km while the other runs at the same height but above the main trendline. These additional lines suggest potential subgroups or distinct characteristics within the data:

Below the main line up to around 300 l/100 km: Vehicles in this range may possess specific attributes leading to lower CO2 emissions despite higher fuel consumption. Above the main line up to around 300 l/100 km: Vehicles in this range may have characteristics resulting in higher CO2 emissions even with relatively lower fuel consumption. Further exploration and subgroup analysis could provide insights into the unique features influencing CO2 emissions within these fuel consumption ranges.

figure_17.png

Figure 16: Influence of weight and hybrid on CO2 emissions

For vehicles with a weight below approximately 1,800 kg (minimum weight), the green points (Hybrid vehicles) seem to have generally lower CO2 emissions compared to the orange points (Non-Hybrid vehicles). This could suggest that Hybrid vehicles may be more efficient in terms of CO2 emissions at lighter weights. From a weight of about 1,800 kg onwards, there appears to be a mixing of green and orange points indicating that the influence of vehicle weight on CO2 emissions becomes more similar between Hybrid and Non-Hybrid vehicles. It would be interesting to conduct additional analyses such as dividing the data into weight ranges and examining average CO2 emissions in these ranges. This could help further explore the relationship between weight, hybrid properties, and CO2 emissions.

The horizontal bar plot shows the mean CO2 emissions for different car fuel types. The order of the bars from top to bottom represents the ascending order of mean CO2 emissions. Here's an interpretation based on our observation:

- ES/GN (Ethanol/Gasoline): This category has the highest mean CO2 emissions among the displayed fuel types. It indicates that vehicles using a combination of Ethanol and Gasoline tend to emit more CO2 on average.
- EE (Electric/Electric): This category represents vehicles that are fully electric (Electric) and use an electric powertrain exclusively. The "EE" category has the lowest mean CO2 emissions among the

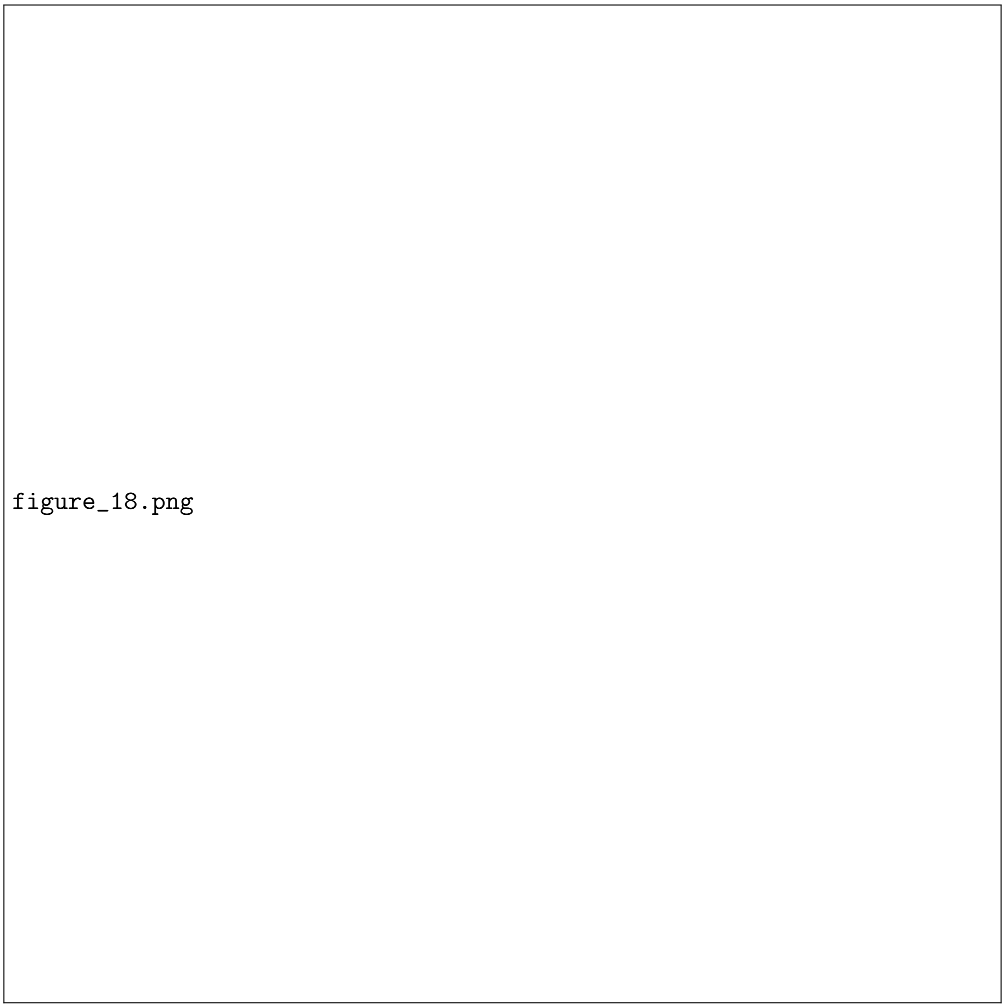

Figure 17: Distribution of CO2 emissions by car fuel type

displayed fuel types. The low CO2 emissions in this category indicate a reduced carbon footprint compared to vehicles with other fuel types.

- EL (Electric/LPG): The absence of a bar for this category in the plot suggests that there might be very few or no vehicles with the combination of Electric and LPG in the dataset. As a result, the dataset does not provide sufficient information to calculate a meaningful mean CO2 emissions value for this particular fuel type.

Information about other fuel types:

- FE (Electric)
- ES (Electric/Gasoline) and Go (Gasoline)
- GN/ES (Natural Gas/Electric)
- EH (Electric/Hybrid)
- GN (Natural Gas)
- ES/GP (Electric/LPG)
- GP/ES (LPG/Electric)

- GH (Gasoline/Hybrid)
- GL (LPG/Hybrid)

These interpretations are based on the mean values, and individual vehicles within each category may vary in their CO2 emissions.

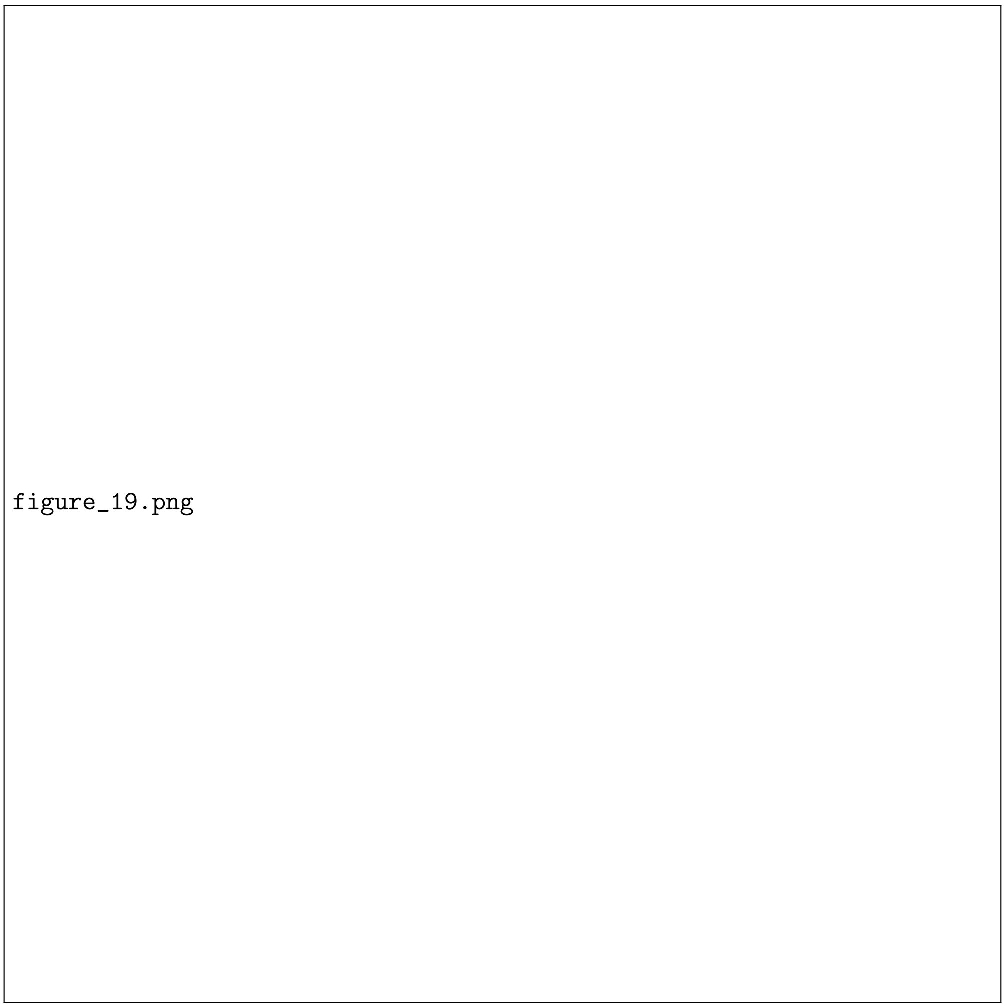

Figure 18: Relationship between power and CO2 emissions

We can identify two observations based on this diagram:

- Vehicles with more power tend to have higher CO2 emissions and vice versa. This observation aligns with general expectations as vehicles with higher power often have larger engines and may consume more fuel.
- A large number of data points are concentrated within a specific range of power values such as between approximately 50 and 250. The concentration of data points in a specific power range may indicate that many vehicles in the dataset share similar power characteristics. Vehicles with extremely high-power values (e.g., greater than 400 kW) are often high-performance or specialty vehicles. These could include sports cars, luxury vehicles, or other niche segments. The limited number of data points in this range might be reflective of the relatively lower production volume of such vehicles compared to mainstream models.

Based on this figure, we can make some observations about which car makes are performing relatively well or poorly in terms of CO2 emissions:

figure_20.png

Figure 19: Distribution of CO2 emissions by Car make

Low CO2 emissions: Car makes with lower mean and median CO2 emissions compared to the overall average of approximately 191.6 g/km can be considered as performing relatively well in terms of emissions efficiency. Examples of such car makes include:

- ALFA-ROMEO: Mean = 134.78 g/km, Median = 134.0 g/km
- CITROEN: Mean = 133.01 g/km, Median = 133.5 g/km
- DACIA: Mean = 130.93 g/km, Median = 131.5 g/km
- KIA: Mean = 133.21 g/km, Median = 134.5 g/km
- PEUGEOT: Mean = 137.76 g/km, Median = 139.0 g/km

High CO2 Emissions: Car makes with higher mean and median CO2 emissions compared to the overall average indicate relatively poorer emissions performance. Examples of such car makes include:

- BENTLEY: Mean = 313.55 g/km, Median = 338.0 g/km
- LAMBORGHINI: Mean = 339.75 g/km, Median = 341.0 g/km
- MAYBACH: Mean = 358.31 g/km, Median = 350.0 g/km

- MERCEDES-BENZ: Mean = 205.04 g/km, Median = 208.0 g/km
- ROLLS-ROYCE: Mean = 342.75 g/km, Median = 347.0 g/km

Various factors need to be considered while categorizing car makes as “doing good” or “poorly” such as the vehicle type, market segment, and the size and performance of vehicles.

6 Sprint 2 (Pre-processing and Feature Engineering)

In this section, we describe the steps we undertook to prepare the dataset, rendering it suitable for ML models.

6.1 Handling Missing Values

As explained in the previous section, some of the columns have missing values.

HC (g/km): Given that this attribute has a high percentage of missing values (76.8%), we decided to drop this column from the dataset.

HC+NOX (g/km): With nearly 24% missing values in this column, using mean or median imputation might not be the best approach, especially if the missingness is not completely at random. Therefore, we used K-Nearest Neighbors (KNN) Imputation to handle missing values in this column. We used this approach to handle missing values in other numerical columns including Particules (g/km), NOX (g/km), CO type I (g/km), Consommation urbaine (l/100km), Consommation extra-urbaine (l/100km), Consommation mixte (l/100km), and CO2 (g/km).

Champ V9: This is a categorical column and has approximately 0.52% missing values. We used the mode imputation approach to replace missing values with the most frequent category (mode) in the column.

Other Numeric Columns (e.g., Puissance maximale (kW), Puissance administrative, etc.): For these columns with no missing values, no action is needed.

Other Categorical Columns (e.g., Carrosserie, Carburant, etc.): Since there are no missing values in these columns, no action is needed.

6.2 Handling Duplicates

In the next step, we removed the duplicate values.

6.3 Handling Categorical Variables

The frequency of categorical variables in the dataset is not the same. The number of categories for each categorical variable is as follows:

- Marque: 51 categories
- Modèle dossier: 458 categories
- Modèle UTAC: 419 categories
- Désignation commerciale: 3,582 categories
- CNIT: 44,191 categories
- Type Variante Version (TVV): 28,781 categories
- Carburant: 13 categories
- Hybride: 2 categories
- Boîte de vitesse: 16 categories
- Champ V9: 13 categories

- Date de mise à jour: 3 categories
- Carrosserie: 10 categories
- Gamme: 7 categories

As it can be seen, the number of categories for some of the variables are very high. Therefore, we need to decide how to handle each categorical variable based on the number of categories and our specific requirements for the analysis. We used the following approaches to handle categorical variables:

One-Hot Encoding (OHE): This approach creates binary columns for each category indicating the presence or absence of the category. However, it can lead to a significant increase in the dimensionality of the dataset, which might not be feasible with extremely large categories. Label Encoding is another approach that we could use to handle categorical variables. However, as the values in the categorical variables are alphanumeric codes rather than ordinal categories, label encoding might not be appropriate for such data because it would imply an order that may not exist. Therefore, we decided to apply OHE to the following variables:

- Marque
- Modèle UTAC
- Carburant
- Hybride
- Boîte de vitesse
- Date de mise à jour
- Carrosserie
- Gamme

Removing some of the columns: Analyzing the dataset shows that "Modèle dossier" and "Modèle UTAC" are both indicating the model name which is, for example, "RANGE ROVER." Therefore, we decided to drop the Modèle dossier column and keep the Modèle UTAC column. Furthermore, some of the columns are not meaningful to us (domain knowledge needed); therefore, we decided to remove them. These columns are:

- Désignation commerciale
- CNIT
- Type Variante Version (TVV)
- Champ V9

6.4 Normalizing Numerical Variables

Normalizing Numerical Variables refers to the process of adjusting numerical data to be on a consistent scale or brought into a specific distribution. This step is crucial to ensure that numerical features are comparable and to avoid biases due to different scales or units. There are two common methods for normalizing numerical variables:

Scaling to a Unit Interval (Min-Max Scaling): This method transforms the data by scaling each value to a range between 0 and 1. The formula is:

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Here X is the original value, X_{new} is the scaled value, and X_{min} and X_{max} are the minimum and maximum values of the variable.

Standardization (Z-Score Normalization): This method transforms the data to have a standard normal distribution (mean 0, standard deviation 1). The formula is:

$$X_{new} = \frac{X - mean(X)}{std(X)}$$

In this case, each value X is divided by the mean of the variable and then by the standard deviation of the variable.

Normalization is particularly important in algorithms based on distance measures such as k-nearest neighbors (k-NN) or gradient descent in neural networks. It ensures that all features are equally weighted and improves the convergence speed of optimization algorithms. We used the Min-Max Scaling method to normalize the numerical variables. Min-Max Scaling ensures the preservation of relative distances between data points. This preservation of relationships is particularly meaningful when the connections between values play a crucial role, especially in applications like k-NN (k-nearest neighbors).

6.5 Outlier Detection and Treatment

We used the Z-score method to identify outliers and handled them via the winsorization technique. The Z-score, also known as the standard score, measures how many standard deviations a data point is from the mean of a distribution. It is calculated using the formula:

$$Z = \frac{X - \mu}{\sigma}$$

where Z is the Z-score, X is the individual data point, μ is the mean of the distribution, and σ is the standard deviation of the distribution.

Winsorization is a statistical technique used to handle outliers in a dataset. Instead of removing extreme values, Winsorization involves replacing values beyond a certain threshold with the nearest value within that threshold. The idea is to limit the impact of extreme values on statistical analyses while retaining the information they provide. This method is named after the concept of "trimming" or "capping" the tails of a distribution and it's particularly useful when a dataset has outliers that might skew the analysis. In the optimization phase of the project, we handled outliers for two columns (NOX (g/km) and HC+NOX (g/km)) differently than others. Since some cars may have zero emissions, we computed the upper quantiles specifically for these columns and addressed them separately by assigning outliers a value of zero.

The number of outliers in each column after treatment:

Column Name	# Outliers	Column Name	# Outli
Puissance administrative	1041	Marque_QUATTRO	
Puissance maximale (kW)	1049	Marque_RENAULT	
Consommation urbaine (l/100km)	504	Marque_RENAULT TECH	
Consommation extra-urbaine (l/100km)	302	Marque_ROLLS-ROYCE	
Consommation mixte (l/100km)	394	Marque_SEAT	
CO2 (g/km)	263	Marque_SKODA	
CO type I (g/km)	573	Marque_SMART	
NOX (g/km)	3644	Marque_SSANGYONG	
HC+NOX (g/km)	617	Marque_SUBARU	
Particules (g/km)	59	Marque_SUZUKI	
masse vide euro min (kg)	298	Marque_TESLA	
masse vide euro max (kg)	122	Marque_TOYOTA	
gamme_ECONOMIQUE	219	Marque_VOLKSWAGEN	
gamme_INFERIEURE	1622	Marque_VOLVO	
gamme_LUXE	0	Carburant_EE	
gamme_MOY-INF	2	Carburant_EH	
gamme_MOY-INFER	0	Carburant_EL	
gamme_MOY-SUPER	0	Carburant_ES	
gamme_SUPERIEURE	1956	Carburant_ES/GN	
Marque_ALFA-ROMEO	103	Carburant_ES/GP	
Marque_ASTON MARTIN	78	Carburant_FE	
Marque_AUDI	242	Carburant_GH	
Marque_BENTLEY	22	Carburant_GL	
Marque_BMW	525	Carburant_GN	
Marque_CADILLAC	44	Carburant_GN/ES	
Marque_CHEVROLET	63	Carburant_GO	
Marque_CITROEN	207	Carburant_GP/ES	
Marque_DACIA	30	Boîte de vitesse_A 0	
Marque_DANGEL	40	Boîte de vitesse_A 4	
Marque_FERRARI	21	Boîte de vitesse_A 5	
Marque_FIAT	415	Boîte de vitesse_A 6	1
Marque_FORD	296	Boîte de vitesse_A 7	
Marque_HONDA	51	Boîte de vitesse_A 8	
Marque_HYUNDAI	52	Boîte de vitesse_D 5	
Marque_INFINITI	26	Boîte de vitesse_D 6	
Marque_JAGUAR	43	Boîte de vitesse_D 7	
Marque_JAGUAR LAND ROVER LIMITED	55	Boîte de vitesse_M 5	1
Marque_JEEP	74	Boîte de vitesse_M 6	
Marque_KIA	78	Boîte de vitesse_M 7	
Marque_LADA	9	Boîte de vitesse_N 0	
Marque_LAMBORGHINI	16	Boîte de vitesse_N 1	
Marque_LANCIA	37	Boîte de vitesse_S 6	
Marque_LEXUS	175	Boîte de vitesse_V 0	
Marque_LOTUS	26	Carrosserie_BERLINE	
Marque_MASERATI	8	Carrosserie_BREAK	2
Marque_MAYBACH	13	Carrosserie_CABRIOLET	
Marque_MAZDA	46	Carrosserie_COMBISPACE	
Marque_MERCEDES AMG	174	Carrosserie_COUPE	1
Marque_MERCEDES-BENZ	0	Carrosserie_MINIBUS	
Marque_MIA	21	Carrosserie_MINISPACE	
Marque_MINI	79	Carrosserie_MONOSPACE	
Marque_MITSUBISHI	39	Carrosserie_MONOSPACE COMPACT	
Marque_NISSAN	173	Carrosserie_TS TERRAINS/CHEMINS	1

6.6 Splitting Data

We split the dataset into training, validation, and test sets to evaluate the performance of our machine learning models properly in the next sprints.

7 Sprint 3

In this sprint, we want to train the baseline models with the preprocessed data and use it to predict the CO2 emissions from different types of cars. The target feature to estimate is CO2 (g/km). In the following sections, we present the results of baseline models.

7.1 Linear Regression

7.1.1 Linear Regression without Dimensionality Reduction

The results of applying Linear Regression without reducing the dimension are shown in Table 2. A low MSE, MAE, and RMSE indicate that the model's predictions are close to the actual values while a high R-squared value suggests that the model explains a significant proportion of the variance in the target variable. Overall, these results indicate that the model performs well in predicting CO2 emissions.

7.1.2 Linear Regression with Dimensionality Reduction

We applied PCA to reduce the dimensionality of the dataset. The results of evaluation metrics are shown in Table 2. Reducing the dimensionality with PCA while retaining 95% of the variance has led to faster model application which is a common advantage of dimensionality reduction techniques. However, the evaluation metrics show that the model's performance slightly decreased after applying PCA compared to the previous model without dimensionality reduction. The mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE) have increased indicating higher prediction errors. Additionally, the R-squared (R2) value decreased suggesting that the model explains less variance in the target variable.

7.2 Ridge Regression

Ridge Regression is a type of linear regression that incorporates regularization to address high correlation between predictor variables and reduce the model's sensitivity to noisy input data. The results of applying this model are presented in the next sections.

7.2.1 Ridge Regression without Dimensionality Reduction

The results of applying Ridge Regression without reducing the dimension are presented in Table 2. This model seems to perform well based on these metrics, with low errors and a high R-squared value indicating a good fit to the data. To identify the optimal value of alpha, we used cross-validation. These are the results of the application of the model with the optimal value for alpha: 0.1. It seems that using the optimal alpha value has resulted in improvements in most of the evaluation metrics. The Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R2) values are slightly better with the optimal alpha indicating that the model's performance improved after tuning the regularization parameter.

7.2.2 Ridge Regression with Dimensionality Reduction

We used PCA to reduce the dimensionality of the dataset. The results of the application of Ridge Regression model after reducing the dimensions are shown in Table 2. It appears that before applying PCA, the Ridge Regression model with an optimal alpha value of 0.1 achieved slightly better performance compared to after applying PCA. The R-squared value was higher (approximately 0.99), indicating that the model explained more variance in the target variable before PCA. Additionally, the MSE, MAE, and RMSE values were lower indicating lower prediction errors. This suggests that the original features without dimensionality reduction might have contained more predictive power for estimating CO2 emissions. However, it's essential to consider the trade-offs between model performance and computational efficiency as dimensionality reduction techniques like PCA can help reduce the complexity of the model and speed up computation especially with larger datasets.

7.3 Lasso Regression

Lasso Regression, also known as L1 regularization, is a linear regression technique that adds a penalty term to the ordinary least squares objective function. This penalty term is the absolute value of the coefficients multiplied by a regularization parameter (alpha), which controls the strength of regularization. The main objective of Lasso Regression is to minimize the sum of the squared residuals between the observed and predicted values, similar to ordinary linear regression. However, it also aims to minimize the sum of the absolute values of the coefficients, thereby encouraging sparsity in the coefficient matrix. This means that Lasso Regression tends to produce models with fewer coefficients, effectively performing feature selection by shrinking less important coefficients towards zero.

The regularization parameter alpha controls the balance between fitting the data well and keeping the model simple. Higher values of alpha result in more regularization, leading to more coefficients being set to zero and a simpler model.

In summary, Lasso Regression is a useful technique for feature selection and regularization in linear regression models, particularly when dealing with high-dimensional datasets with potentially redundant or irrelevant features. Using Lasso Regression alongside Ridge Regression is a common practice, especially when dealing with high-dimensional datasets or when you want to perform feature selection. Lasso Regression tends to produce sparse models by setting some coefficients to zero, effectively performing feature selection, whereas Ridge Regression tends to shrink the coefficients towards zero without necessarily setting them exactly to zero.

The results of the application of this model on our dataset are presented in Table 2. These results show a high mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE) as well as a negative R-squared value. This indicates that the model is performing poorly and may not be capturing the relationship between the features and the target variable effectively. We tried to tune the value of alpha. The results of this model with the optimal value of alpha (0.01) are shown in Table 2. These results show that the performance of the Lasso Regression model with the optimal alpha value (0.01) improved compared to the previous application. However, the model's performance still seems suboptimal as indicated by the relatively low R-squared value (0.3729) and the error metrics. Therefore, this model is not a good model for our project.

7.4 ElasticNet Regression

ElasticNet Regression is a combination of Ridge and Lasso regression techniques which combines their penalties. The results of applying this method to the dataset are presented in Table 2. The results show that the values of MSE, MAE, and RMSE are low. But the negative value of R-squared indicates that the model performs worse than a horizontal line. This suggests that the model is not able to explain the variance in the data and is performing poorly. We tried to perform hyperparameter tuning for ElasticNet regression using cross-validation with GridSearchCV. The results are:

Best Hyperparameters: 'alpha': 0.1, 'l1$_ratio' : 0.1MeanSquaredError(MSE) : 0.0033337131647687627R-$ $squared(R2) : 0.30028459159136967Withthesehyperparameters, theElasticNetmodelachievedanMSEofapproximatel$ $squaredvalue.Therefore, thismodelisnotagoodmodelforourproject.$

7.5 Decision Trees

Decision Trees are a popular algorithm for regression tasks; therefore, we decided to apply this model to our dataset. We applied this model with and without dimensionality reduction. The results are presented in the next two sections.

7.5.1 Decision Trees without Dimensionality Reduction

The evaluation results of this model are presented in Table 2. A very low MSE, high R-squared, and small MAE and RMSE indicate that this model is performing exceptionally well and is able to make accurate predictions with very small errors. Additionally, the explained variance score of nearly 1 suggests that the model is able to explain almost all of the variance in the target variable. Overall, these metrics indicate that the Decision Tree Regression model is highly effective for the task at hand.

7.5.2 Decision Trees with Dimensionality Reduction

The results of this model are shown in Table 2. These results show that the Decision Tree model without dimensionality reduction performed better across all metrics compared to the one with dimensionality reduction. It should be noted that applying PCA before a decision tree (or ensemble methods like Random Forest) is less common compared to using it with linear models. Decision trees, including Random Forest, are capable of handling high-dimensional data and nonlinear relationships without the need for dimensionality reduction techniques like PCA.

7.6 Random Forest

Random Forest is a popular ensemble learning technique that combines multiple decision trees to improve predictive performance and reduce overfitting. It's known for its robustness and ability to handle large datasets with high dimensionality. The results of applying this model are shown in Table 2. These results show that the Random Forest model seems to perform very well, achieving low values for MSE, MAE, and RMSE, and a high value for R2 indicating a good fit to the data. The Explained Variance Score is also very high, suggesting that the model explains most of the variance in the target variable.

7.7 Bagging Algorithms

Bagging is an ensemble technique where multiple models (often of the same type) are trained on different subsets of the training data. The final prediction is typically made by averaging the predictions of all models (for regression) or using voting (for classification). Random Forest is a popular bagging algorithm based on decision trees. It builds multiple decision trees and merges their predictions to improve accuracy and reduce overfitting. We first applied this model to the dataset. Although the results of this model were satisfactory, we wanted to try some other Bagging models including Bagged Decision Trees and Bagged SVM. The results of these models are presented in the following sections.

7.7.1 Random Forest

Random Forest is a popular ensemble learning technique that combines multiple decision trees to improve predictive performance and reduce overfitting. It's known for its robustness and ability to handle large datasets with high dimensionality. The results of applying this model are presented in Table 2. These results show that the Random Forest model seems to perform very well, achieving low values for MSE, MAE, and RMSE, and a high value for R2 indicating a good fit to the data. The Explained Variance Score is also very high, suggesting that the model explains most of the variance in the target variable.

7.7.2 Bagged Decision Trees

In Bagged Decision Trees, multiple decision tree models are trained on random subsets of the training data with replacement. Each tree is built independently, meaning that they can have different splits and structures. During prediction, the output of the bagged ensemble is typically the average (for regression tasks) or the majority vote (for classification tasks) of the predictions made by individual trees. Bagged decision trees are particularly effective when dealing with high variance models as they reduce overfitting and improve generalization. The results of applying this model are presented in Table 2. The results show that both Random Forest and Bagged Decision Trees perform quite well across the metrics, with very small values for most error metrics and high values for the R2 and Explained Variance Score. It seems like Random Forest slightly outperforms Bagged Decision Trees in terms of most metrics, but the differences are minimal.

7.7.3 Bagged SVM

Bagged SVM applies the bagging technique to Support Vector Machines. In this approach, multiple SVM models are trained on different subsets of the training data. SVMs are known for their ability to find the optimal decision boundary, but they can be sensitive to the choice of hyperparameters and the specific training data. Bagging helps to reduce the variance of the model by training multiple SVMs on different subsets of the data and averaging their predictions. This can lead to improved performance and robustness, especially in scenarios where the data is noisy or contains outliers. The results of applying this model are presented in Table 2. The results show that compared to the results of Random Forest and Bagged Decision Trees, Bagged SVM seems to have higher error metrics (MSE, MAE, RMSE) and lower scores for R2 and Explained Variance Score. Therefore, for our project, Random Forest and Bagged Decision Trees might be more effective models than Bagged SVM.

7.8 Boosting Algorithms

Boosting is another ensemble technique where multiple weak learners (often shallow decision trees) are trained sequentially and each subsequent model tries to correct the errors made by the previous one. Gradient Boosting is one of the most popular boosting algorithms. It builds trees sequentially, and each tree tries to correct the errors of the previous one. We applied three boosting algorithms on the dataset including AdaBoost, Gradient Boosting, and XGBoost.

7.8.1 AdaBoost

AdaBoost is a boosting algorithm that works by iteratively training a series of weak learners (typically decision trees) and combining their predictions to create a strong learner. It adjusts the weights of incorrectly classified instances in each iteration to focus on the difficult examples. The results are presented in Table 2. The results show that AdaBoost performs quite well across most metrics with low error metrics (MSE, MAE, RMSE) and high scores for R2 and Explained Variance Score. It seems to be a strong performer based on these metrics.

7.8.2 Gradient Boosting

Gradient Boosting is another boosting algorithm that builds a series of decision trees sequentially. Each tree corrects the errors made by the previous ones with a focus on minimizing the residual errors. The results are presented in Table 2. Comparing the results for AdaBoost and Gradient Boosting show that both algorithms perform exceptionally well across these metrics. Gradient Boost, however, appears to have slightly lower error metrics and higher scores for R2 and Explained Variance Score compared to AdaBoost, indicating that Gradient Boost might be slightly more accurate for our project.

7.8.3 XGBoost

XGBoost is a scalable and efficient implementation of Gradient Boosting. It's known for its speed and performance improvements over traditional Gradient Boosting methods. The results of this model are presented in Table 2. These results show that among XGBoost and the two previous algorithms, XGBoost performs the best across all metrics with the lowest error metrics and the highest scores for R2 and Explained Variance Score. Gradient Boost also performs very well while AdaBoost performs slightly lower compared to the other two.

7.9 Deep Learning

Deep learning models such as neural networks can capture complex patterns in data and are suitable for various types of tasks including regression and classification. We applied different DL techniques that have been presented in the following sections.

7.9.1 Multi-layer Perceptron (MLP)

A Multi-layer Perceptron is a basic type of neural network with multiple layers of neurons (nodes) and non-linear activation functions. The results of applying this technique on the dataset are shown in Table 2. Based on these results, the MLP model appears to perform very well on the task. It exhibits low errors, high explanatory power, and effectively captures the underlying patterns in the data. We first trained the model through 100 epochs. Then, to reduce the training time and improve performance, we used early stopping. Early stopping halts training when the model's performance on a validation dataset stops improving, which prevents overfitting and reduces training time. With this technique, training the model stopped after 53 epochs (4 min). Evaluation results of this model by using early stopping are shown in Table 2.

7.9.2 Convolutional Neural Networks (CNNs)

CNNs are primarily used for image recognition and processing tasks. They are very effective at capturing spatial patterns in data due to their unique architecture which includes convolutional layers, pooling layers, and fully connected layers. Although CNNs are suitable for image data including tasks such as image classification, object detection, and image segmentation, we conducted experiments with different techniques and architectures.

References