DataScientest

CO2 Car Emission Prediction

By

Abd Akdim - akdimabd@gmail.com
Halimeh Agh - agh.halime@gmail.com
Azangue Pavel - pavelazangue@gmail.com

Supervisor

Sarah Lenet - sarah.l@datascientest.com

School
DataScientest Paris

Table of 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

Any data science project, including our CO2 Emission Prediction project, must follow the technique shown in the graphic to be executed successfully. This organized method guarantees a thorough and methodical investigation, leading us through every crucial stage of the undertaking.

The process begins with Data Mining and Visualization, where data is collected and exploratively analyzed to gain insights. Following this, Pre-processing involves cleaning, transforming, and preparing data for further analysis.

Next, Feature Engineering extracts and combines features to enhance the predictive models' performance. This step is crucial for improving model accuracy and effectiveness.

Modeling applies machine learning algorithms to identify patterns and make predictions based on the prepared data. Various models are evaluated and optimized during this stage.

Interpretation of results analyzes the model outcomes to derive conclusions and actionable insights. This step helps in understanding the implications of the model findings.

Finally, the Conclusion summarizes the findings and draws final conclusions from the entire analysis process, providing a clear overview of the insights gained.

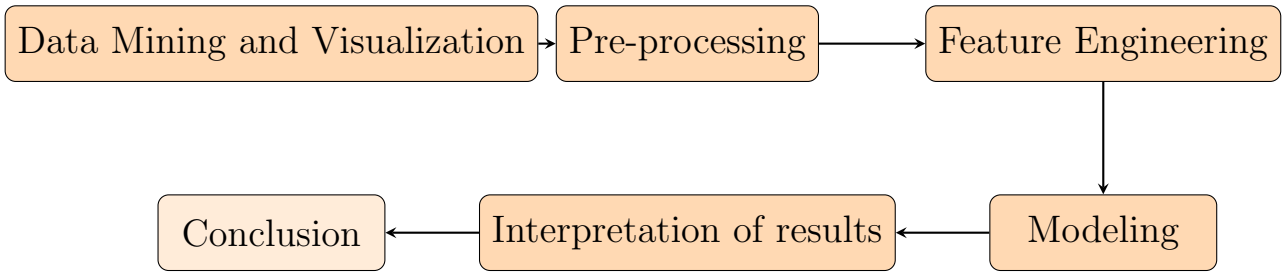

Figure 1: Workflow of the CO2 Emission Prediction by vehicles Project

3 Data Mining and Visualization

Our data science project, which aims to comprehend CO2 emissions by vehicles, heavily relies on the "Data Mining and Visualization" component. In this stage, the dataset is thoroughly examined to find any hidden structures, challenges, or biases. This information serves as a basis for additional analysis and modeling.
This phase's main goals are to perform a thorough analysis of the dataset in order to emphasize its structure, spot any challenges and biases, and produce understandable visual representations of the data. During the modeling phase, these procedures facilitate improved understanding of the data and assist make well-informed judgments.

3.1 Data Mining

Chapter 3.1 introduces fundamental concepts and methods of data mining. It explores various attributes such as brand, model, fuel type, and emissions. Special attention is given to numerical variables, which are analyzed using statistical measures to identify patterns and outliers. Additionally, categorical variables are discussed, highlighting their diversity and presenting the most common values. This chapter provides a comprehensive exploration of data mining techniques, covering both quantitative and qualitative aspects of data analysis.

3.2 Data Exploration

In our dataset, there are 44,850 items with 26 columns, each of which represents a distinct property of a car that was sold in France in 2013. Important characteristics include the fuel type (petrol, extra-urban, and mixed), vehicle model name, fuel consumption (urban, extra-urban, and mixed), CO2 emissions, and other technical details.
The following ***table 1*** displays the variables contained in the dataset along with their descriptions.

- **Size and structure:** 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.

3.2.1 Data Types

The dataset contains a mix of data types including object, int64, and float64. Table 2 shows the type of each column.

- **Object**: A data structure that contains data and methods; Example: `obj = {}`
- **Int64**: A 64-bit integer data type; Example: `num_int = 42`.
- **Float64**: A 64-bit floating point number data type; Example: `num_float = 3.14`

The tabl 2 2displays all data for each column, including those with missing

Column	Non-Null Count	type	Missing Data (%)
Marque	44850	object	0.00
Modèle dossier	44850	object	0.00
Modèle UTAC	44850	object	0.00
Désignation commerciale	44850	object	0.00
CNIT	44850	object	0.00
Type Variante Version (TVV)	44850	object	0.00
Carburant	44850	object	0.00
Hybride	44850	object	0.00
Puissance administrative	44850	int64	0.00
Puissance maximale (kW)	44850	float64	0.00
Boîte de vitesse	44850	object	0.00
Consommation urbaine (l/100km)	44808	float64	0.09
Consommation extra-urbaine (l/100km)	44808	float64	0.09
Consommation mixte (l/100km)	44811	float64	0.09
CO2 (g/km)	44811	float64	0.09
CO type I (g/km)	44547	float64	0.68
HC (g/km)	10403	float64	76.81
NOX (g/km)	44547	float64	0.68
HC+NOX (g/km)	34191	float64	23.77
Particules (g/km)	41708	float64	7.01
Masse vide euro min (kg)	44850	int64	0.00
Masse vide euro max (kg)	44850	int64	0.00
Champ V9	44615	object	0.52
Date de mise à jour	44850	object	0.00
Carrosserie	44850	object	0.00
Gamme	44850	object	0.00

Table 2: Vehicle Attributes, Non-Null Count, Data Types and Missing Data Percentage

data. Displaying missing data helps to verify the completeness and quality of the datasets. Columns such as HC (g/km), HC+NOX (g/km), Particules (g/km), and others exhibit the highest percentages of missing data. Identifying such columns allows prioritization for data cleaning to ensure the accuracy and usability of the information.

3.2.2 Handling of Missing Values

During our investigation, we found that various columns including **CO2 emissions, consumption of urban areas and extra-urban areas** had missing numbers. It was essential to handle those missing values for precise modeling and analysis.

3.2.3 Numerical Variables

Numerical variables in data analysis represent numeric values, including integers and floats. They are used for quantitative measurements such as mean,

standard deviation (σ), minimum, maximum, and percentiles. These values are essential for understanding central tendency, variability, and distribution of data, which are fundamental for statistical analysis and modeling [1].

Mean ($\bar{x}$) is the value obtained by summing all data points x_i divided by the number of data points n.

$$\text{Mean} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Standard Deviation (Std or σ) measures the average deviation of data points x_i from the mean.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Minimum (Min) is the smallest value in the data set $x_1, x_2, \ldots, x_n$.

$$\text{Min} = \min(x_1, x_2, \ldots, x_n)$$

Max (Maximum): The largest value in the dataset.

$$\text{Max} = \max(x_1, x_2, \ldots, x_n)$$

Figure 2 illustrates a Boxplot, a common tool in data science for visualizing data distribution. It effectively identifies outliers, which are data points that lie significantly outside the range of

$$Q1 - 1.5 \times IQR \quad \text{to} \quad Q3 + 1.5 \times IQR$$

Outliers can skew statistical analyses and machine learning models, emphasizing the importance of their detection for accurate insights.

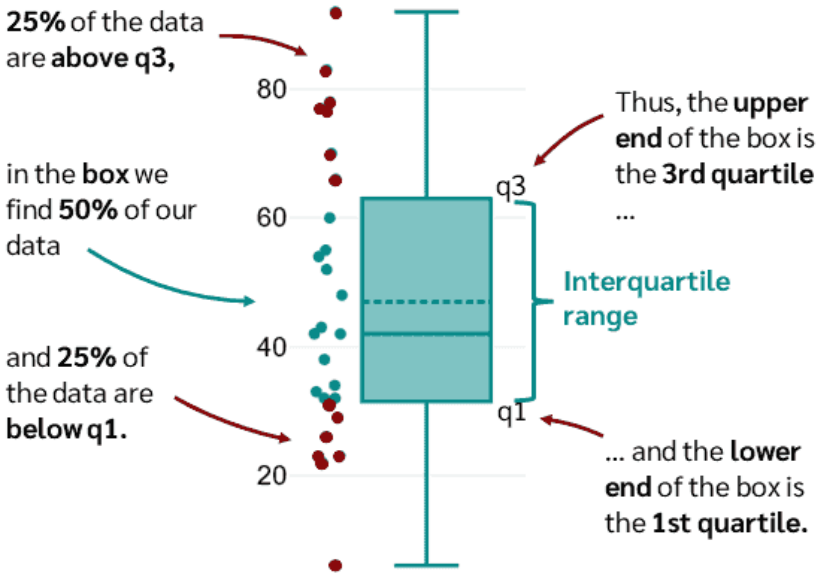

Figure 2: Boxplot description [2]

25% (Q1, First Quartile): The value below which 25% of the data fall and can be found using:

$$P_{25} = 0.25 \times (n + 1).$$

50% (Median, Q2): The value below which 50% of the data fall and is computed as:

$$P_{50} = 0.5 \times (n + 1).$$

75% (Q3, Third Quartile): The value below which 75% of the data fall and is calculated as follows:

$$P_{75} = 0.75 \times (n + 1).$$

Our dataset contains seven different numerical attributes. Table 3 summarizes the calculated statistical values of these attributes. This summary provides a clear overview of the distribution and variability of the data, enabling a thorough analysis of emissions and vehicle mass data.

Attribute	Mean	Std	Min	25%	50%	75%	Max
CO type I (g/km)	0.153461	0.138984	0.005	0.046	0.093	0.222	0.968
HC (g/km)	0.030499	0.018408	0.008	0.008	0.031	0.044	0.143
NOX (g/km)	0.311837	0.463112	0.001	0.158	0.197	0.228	1.846
HC+NOX (g/km)	0.224788	0.041681	0.038	0.201	0.220	0.248	0.306
Particules (g/km)	0.000961	0.006469	0.000	0.000	0.001	0.001	0.610
Masse vide euro min (kg)	2070.961650	342.872975	825.000	1976.000	2076.000	2256.000	3115.000
Masse vide euro max (kg)	2169.545284	410.600541	825.000	2043.500	2185.000	2355.000	3115.000

Table 3: Attribute Types and Statistics

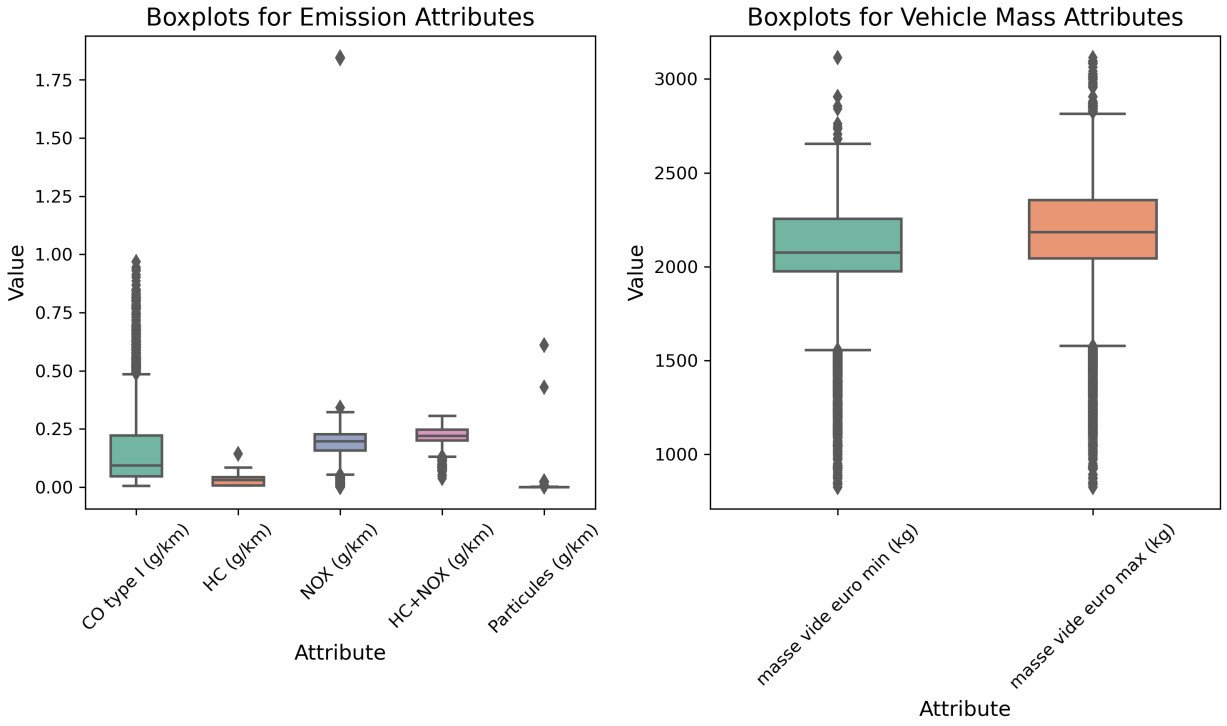

Figure 3: Boxplots for numerical variables

3.2.4 Categorical Variables

Categorical variables such as model, vehicle type, and fuel type are crucial for analysis as they represent qualitative data. Each variable can encompass multiple categories describing various vehicle characteristics. These data allow for identifying patterns and groups to examine their impact on CO2 emissions more closely.

Attribute	Number of Outliers	Lower Whisker	Upper Whisker
CO type I (g/km)	1009 (2.26%)	-0.218	0.486
HC (g/km)	4 (0.04%)	-0.046	0.098
NOX (g/km)	10426 (23.41%)	0.053	0.333
HC+NOX (g/km)	883 (2.58%)	0.131	0.319
Particules (g/km)	3696 (8.86%)	-0.0015	0.0025
Masse vide euro min (kg)	4397 (9.80%)	1556.0	2676.0
Masse vide euro max (kg)	8240 (18.38%)	1576.25	2822.25

Table 4: Outliers and their percentage for each attribute

Variable	Number of Categories	Example Values/Mode
Modèle dossier	458	208, 305, 421
Modèle UTAC	419	A25, B12, C36
Désignation commerciale	3582	Compact, Sedan, SUV
CNIT	44191	123456, 789012, 345678
Type Variante Version (TVV)	28781	Type1, Variant2, Version3
Carburant	13	Diesel, Petrol, Hybrid
Hybride	2	Non, Oui
Boîte de vitesse	16	Manual, Automatic, CVT
Champ V9	13	V9A, V9B, V9C
Carrosserie	10	Hatchback, Sedan, SUV
Gamme	7	Economy, Mid-range, Luxury

Table 5: Overview of categorical variables in the dataset

3.3 Data Visualization

Making difficult data more readable, comprehensible, and useful is one of the main functions of visualization in data analysis. In order to understand the dataset better, we produced a number of visuals. These five important visualizations are accompanied by thorough commentary.

3.3.1 Distribution of CO2 Emissions

By emphasizing the most prevalent emission levels, this representation aids in determining the range and frequency of emissions.

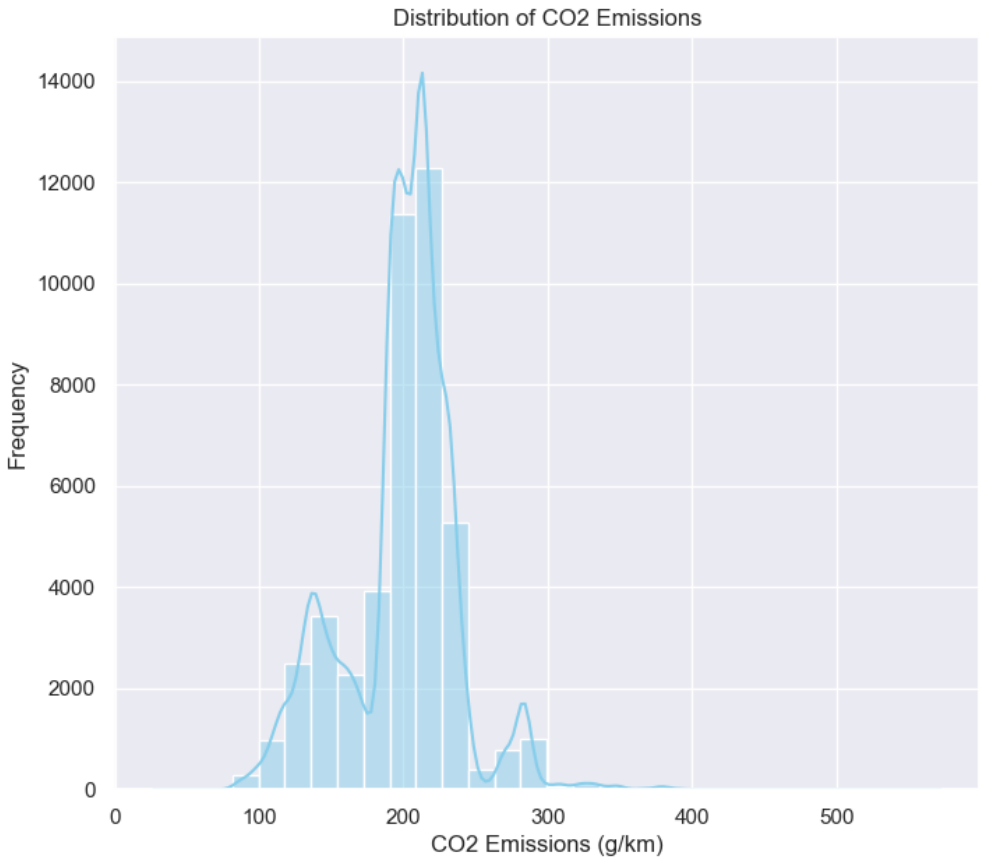

Figure 4: a histogram that displays the CO2 emissions (g/km) distribution for every car.

3.3.2 Relationship between Fuel Consumption and CO2 Emissions

Vehicles that use more gasoline typically emit more CO2 emissions, according to the positive correlation. The scatter (Figure 5) plot illustrates the relationship between mixed fuel consumption (x-axis in liters per 100 kilometers) and CO2 emissions (y-axis in grams per kilometer). The points form three distinct lines with varying slopes, suggesting different groups or clusters of vehicles based on their fuel efficiency and resulting CO2 emissions. This variation in slopes also hints at the possibility of other factors, such as vehicle weight, influencing the linear relationship with fuel consumption and CO2 emissions.

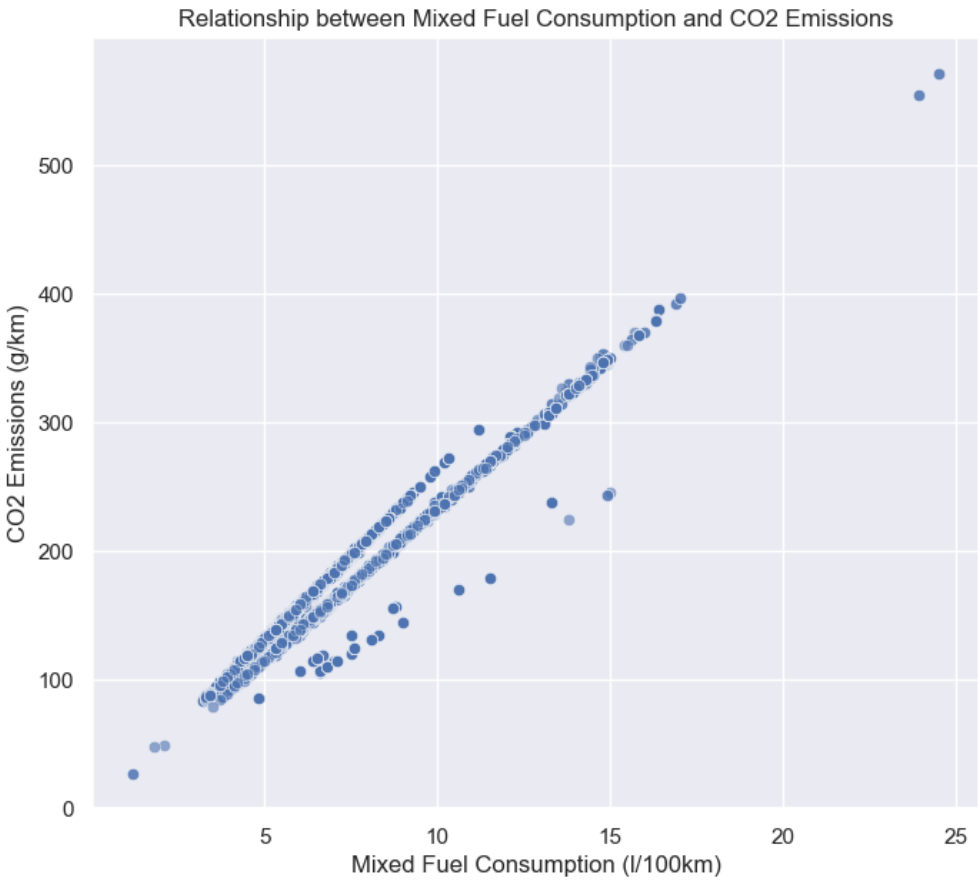

Figure 5: A scatter plot illustrating the relationship between mixed fuel consumption (l/100km) and CO2 emissions.

To better understand the pattern in Diagram 5, we have included in Diagram 6 the linear regression of CO2 emissions against the average vehicle weight. An R2 value of 0.48 indicates that approximately 48% of the variations in CO2 emissions can be explained by the average vehicle weight (calculated as the average between 'masse vide euro min' (kg) and 'masse vide euro max' (kg)). This suggests a moderate explanatory power of vehicle weight on CO2 emissions. This result shows that vehicle weight has a noticeable impact on CO2 emissions, although this influence is not entirely comprehensive.

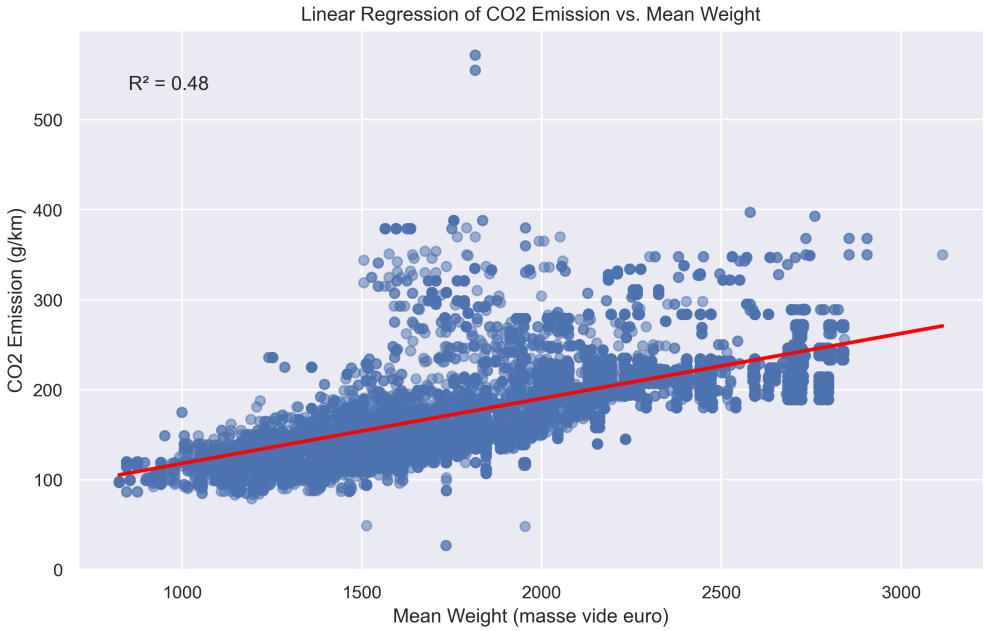

Figure 6: Linear Regression of CO2 Emission vs. Mean Weight.

An R^2 value of 0.48 indicates that approximately 48% of the variations in CO2 emissions can be explained by the average vehicle weight (calculated as the average between "masse vide euro min" (kg) and "masse vide euro max" (kg)). This suggests a moderate explanatory power of vehicle weight on CO2 emissions.This result indicates that vehicle weight has a notable impact on CO2 emissions, although its influence is not entirely comprehensive.

3.3.3 Impact of Vehicle Weight 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 plot differentiates between hybrid and non-hybrid vehicles, revealing

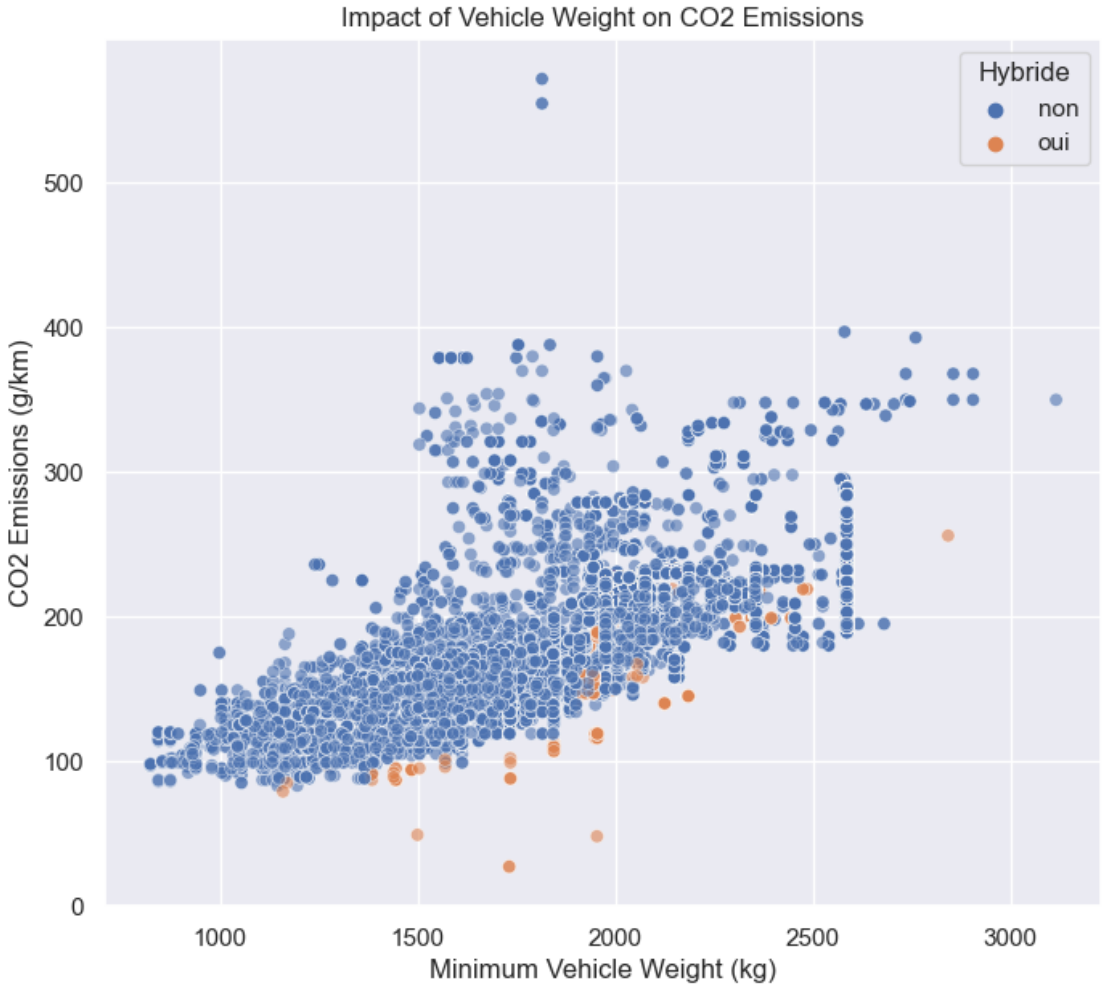

Figure 7: A scatter plot showing CO2 emissions against the minimum vehicle weight (kg).

that hybrid vehicles generally have lower CO2 emissions, particularly for lighter vehicles.

3.3.4 CO2 Emissions by Fuel Type

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 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.

This visualization shows that vehicles using a combination of ethanol and gasoline (ES/GN) have the highest mean CO2 emissions, while fully electric vehicles (EE) have the lowest.

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.

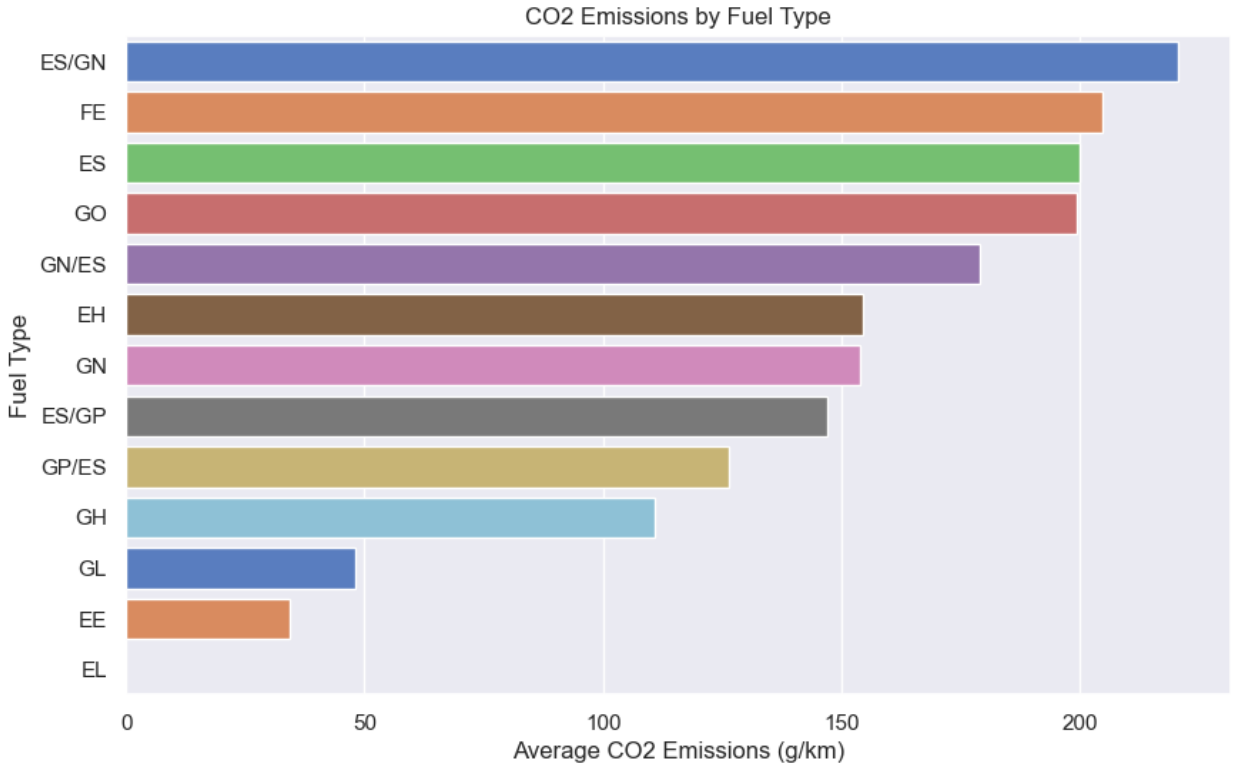

Figure 8: A bar plot displaying the average CO2 emissions for different fuel types.

- 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.

3.3.5 CO2 Emissions by Vehicle Brand

This visualization helps identify which brands are performing better or worse in terms of emissions efficiency.

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

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.

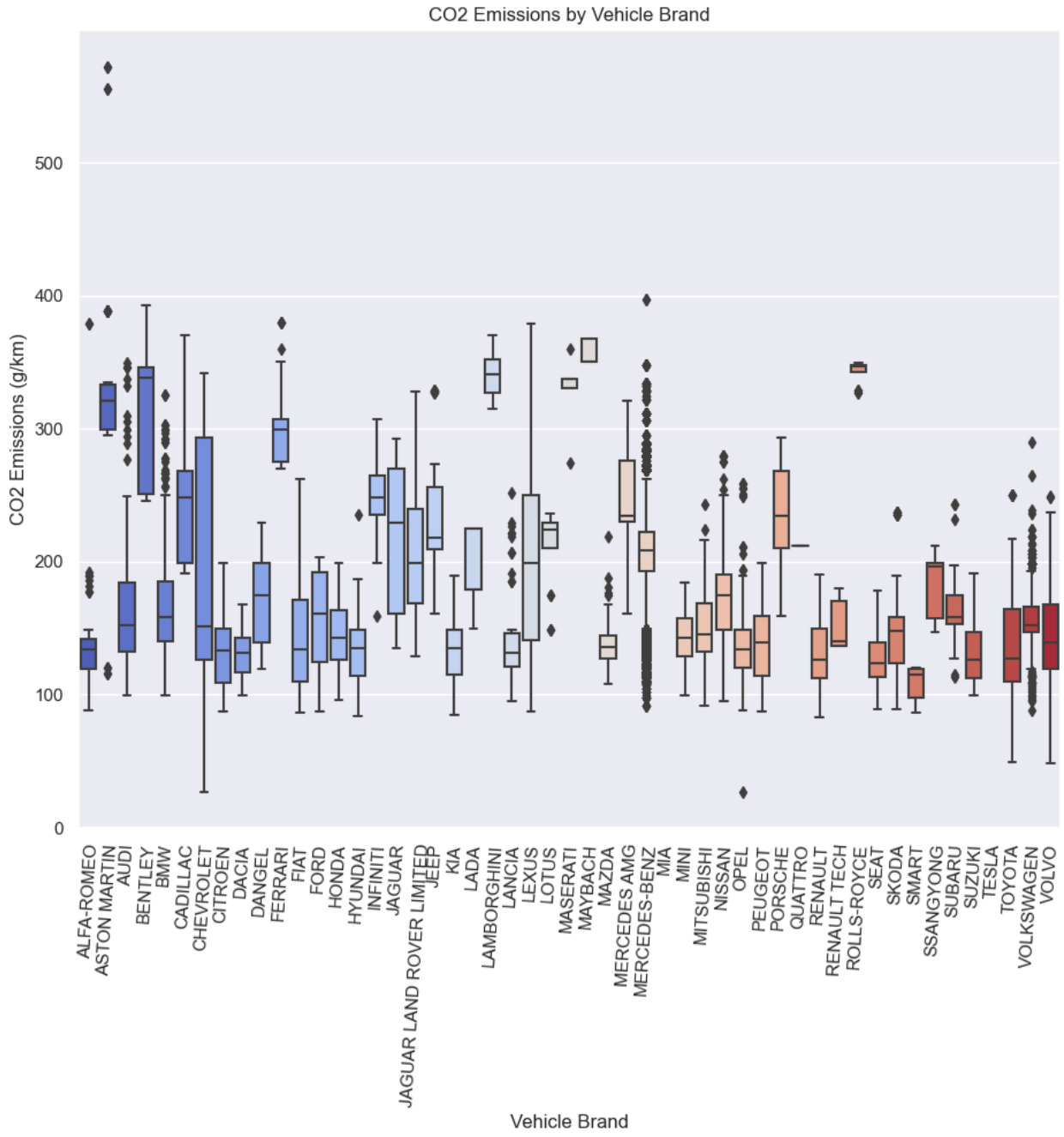

Figure 9: A box plot comparing the CO2 emissions across different vehicle brands (Marque).

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.

3.3.6 Analysis of Visualizations

Every visualization offered insightful information about the dataset:

Distribution Analysis: Knowing how CO2 emissions are distributed makes it easier to identify outliers and establish reasonable benchmarks.
Correlation Analysis: The projected relationship between fuel consumption and CO2 emissions is confirmed by the positive correlation, which also directs future research on fuel efficiency enhancements.
Effect of Vehicle Weight: Assessing how well hybrid technology reduces emissions requires distinguishing between hybrid and non-hybrid automobiles.
Fuel Type Comparison: When suggesting alternative fuels or technologies, it helps to identify the fuel types with higher emissions.
Brand Performance: By comparing emissions by brand, leaders and laggards are identified, laying the groundwork for improvements throughout the whole industry.

The process of data extraction and visualization provided important new information about car CO2 emissions. We established a strong basis for additional analysis and modeling by spotting trends, connections, and outliers.

These understandings are crucial for formulating practical plans to lower car emissions, enhancing environmental sustainability, and directing design and policy choices within the automotive sector.

4 Pre-processing and Feature Engineering

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

4.1 Pre-processing

A critical stage in the data science pipeline is data preparation, which entails preparing and cleaning the dataset to make it fit for modeling and analysis. The purpose of this stage is to deal with problems like noise in the data, missing values, and discrepancies. The preprocessing stage of our CO2 Emissions by Vehicles project comprised the following crucial procedures, which are described in full below.

4.1.1 Handling Missing Values and Duplicates

Multiple columns in the dataset had missing values. If these missing values were not properly handled, the model may become biased and inaccurate. To deal with missing values, we used several tactics tactics.

- **Imputation:**

We utilized mean imputation to fill in the missing values for columns like **CO2 (g/km), Consommation extra-urbaine (l/100km), and Consommation urbaine (l/100km)**. By substituting the column mean for missing values, this technique makes sure that the data's overall distribution is preserved. This technique was mostly used for numerical columns.

The most frequent category in the column is used to replace missing values in categorical columns like **Carburant and Boîte de vitesse** through the use of mode imputation. for instance,

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.

- **Advanced Imputation Techniques:**

We took into consideration more advanced imputation methods like K-Nearest Neighbors (KNN) imputation for columns like HC (g/km) that had a high percentage of missing data. By filling in the missing values using the values of the closest neighbors, this approach maintains the relationships present in the data. As example,

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 missing data is not completely random. Therefore, we used K-Nearest Neighbors (KNN) imputation to handle the missing values in this column. We applied this approach to other numerical columns as well, 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).

Finally, given the fact that this attribute **HC (g/km),** has a high percentage of missing values (76.8%), we decided to drop this column from our dataset since it impacts will irrelevant or simply not weighs much during the modelling part.
In order to completely clean our dataset, duplicate values were removed.

4.1.2 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:

As can be seen, the number of categories for some of the variables is 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 we could use to handle categorical variables. However, since the values in the categorical variables are alphanumeric codes rather than ordinal categories, label encoding might not be appropriate for such

Column Name	number of categories
Marque	51
Modèle dossier	458
Modèle UTAC	419
Désignation commerciale	3582
CNIT	44191
Type Variante Version (TVV)	28781
Carburant	13
Hybride	2
Boîte de vitesse	16
Champ V9	13
Date de mise à jour	3
Carrosserie	10
Gamme	7

Table 6: Categorical variables with their categories

data because it would imply an order that may not exist. Therefore, we decided to apply One-Hot Encoding (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** both indicate the model name, such as **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 due to lack knowledge or insightful data on which we could used; therefore, we decided to remove them. These columns are:

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

4.1.3 Normalizing Numerical Variables

Normalization and standardization are techniques used to scale numerical features to a similar range, which helps improve the performance of machine learning algorithms.

Normalizing numerical variables involves adjusting numerical data to a consistent scale or transforming it 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:

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

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.
This method was applied to columns such as **Consommation urbaine (l/100km)** and **Consommation extra-urbaine (l/100km)**.

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

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

In this case, each value X is divided by the mean of the variable and then by the standard deviation of the variable.
We applied this method to columns such as **Puissance administrative** and **Puissance maximale (kW)**.

To sum up this important section, distance-measuring algorithms, like k-nearest neighbors (k-NN) or gradient descent in neural networks, normalization is very crucial. It guarantees equal weighting of all features and accelerates optimization algorithm convergence. To standardize the numerical variables, we applied the Min-Max Scaling approach. Relative distances between data points are preserved thanks to min-max scaling. This relationship preservation is especially important when relationships between values

are important, like in k-NN applications.

4.1.4 Handling of Outliers

Outliers can distort the results of the analysis and modeling. We used the following methods to detect and handle outliers:

Z-score Method: Data points that deviate from the mean by more than a specific number of standard deviations are classified as outliers according to this methodology. To find outliers in numerical columns, we set a threshold of three standard deviations. It is calculated using the formula below.

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

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.

Interquartile Range (IQR): We determined the IQR for every numerical column, and data points that fell below Q1 - 1.5IQR or above Q3 + 1.5IQR were considered outliers. These anomalies were eliminated or had their values capped at the threshold.

A statistical method for handling outliers in a dataset is called winsorization from **Z-score method** mostly. Winsorization replaces values over a threshold with the closest value falling inside that threshold as opposed to eliminating extreme values. The goal is to preserve the information that extreme values offer while minimizing their influence on statistical studies. The term 'trimming' or 'capping' the tails of a distribution is the inspiration behind the name of this procedure, which comes in handy when a dataset contains outliers that could distort the results. We treated outliers for two columns (NOX (g/km) and HC+NOX (g/km)) differently from other columns during the project's optimization phase. We calculated the top quantiles especially for these columns because certain cars might have zero emissions, and we addressed each one independently by assigning outliers a value of zero.

The number of outliers in each column after handling was recorded in the table below for some of them.

Column Name	#Outliers
Puissance administrative	1041
Puissance maximale (kW)	1049
Consommation urbaine (l/100km)	504
Consommation extra-urbaine (l/100km)	302
Consommation mixte (l/100km)	394
CO2 (g/km)	263
CO type I (g/km)	573
NOX (g/km)	3644
HC+NOX (g/km)	617
Particules (g/km)	59
masse vide euro min (kg)	298
masse vide euro max (kg)	122
gamme_ECONOMIQUE	219
gamme_INFERIEURE	1622
gamme_LUXE	0
gamme_MOY-INF	2
gamme_MOY-INFER	0
gamme_MOY-SUPER	0
gamme_SUPERIEURE	1956

Table 7: Portion of the number of outliers for different columns

4.2 Feature Engineering

A crucial phase of our project, is feature engineering. It entails converting unstructured data into useful features that improve our machine learning models' performance. We develop and choose features that capture the underlying patterns and relationships in the dataset by utilizing domain expertise and data analysis methodologies. This procedure helps to make the models more actionable and comprehensible in addition to increasing their accuracy. This section goes into detail about the methods and approaches we utilized to design features that make our prediction models successful.

4.2.1 Feature Selection

The dataset is now clean, consistent, and ready for the next stage of feature engineering and modeling after the preparation processes have been completed. We have improved the quality of the dataset by changing categorical variables, fixing scaling errors, missing values, and outliers. The dataset has been further reduced in size using feature selection, which makes sure that only the most pertinent features are used for modeling. In the end, our preprocessing work supports our objective of lessening the environmental impact of automobiles by providing a solid basis for developing reliable and accurate

models to anticipate CO2 emissions.

5 modeling

After preparing the data through preprocessing and feature engineering, the next step in our project is modeling. This involves applying statistical models or machine learning algorithms to identify patterns, make predictions, and derive insights. Key activities include selecting appropriate models, training them on data to recognize patterns, evaluating performance, optimizing parameters for better results, and interpreting predictions to derive actionable insights for decision-making. This phase is crucial for transforming raw data into usable insights and decision support.

In this chapter, 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.

5.1 Metrics

Metrics in data science are standards used to evaluate the performance and quality of models or algorithms. They are crucial for objectively assessing how well a model makes predictions and for comparing different approaches.

5.1.1 Mean Squared Error (MSE)

MSE measures the average of the squared differences betweenactual and predicted values.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where y_i are the actual values and $\hat{y}_i$ are the predicted values.
MSE is for evaluating the overall quality of a model. Lower MSE values indicate more accurate predictions. A lower MSE indicates a better model, with values close to 0 being desirable. High MSE values indicate poor model performance, as larger errors are being made.

5.1.2 R^2 (R-Squared)

R^2 measures the proportion of variance in the dependent variable that is explained by the model.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of the actual values. A higher R^2 value indicates that the model explains more of the variance. Values closer to 1 are good, indicating that the model explains a high proportion of the variance, whereas values close to 0 indicate that the model explains very little of the variance.

5.1.3 Mean Absolute Error (MAE)

MAE measures the average magnitude of the errors between actual and predicted values.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MAE provides an intuitive sense of the error magnitude. Lower MAE values indicate better model performance, while higher MAE values suggest larger errors in predictions.

5.1.4 Root Mean Squared Error (RMSE)

RMSE is the square root of the MSE.

$$\text{RMSE} = \sqrt{\text{MSE}}$$

RMSE gives errors in the same units as the original data and is sensitive to large errors. Lower RMSE values are better, indicating more accurate predictions, whereas higher RMSE values indicate worse performance, especially sensitive to large errors.

5.1.5 Mean Squared Logarithmic Error (MSLE)

MSLE measures the squared error between the logarithms of actual and predicted values.

$$\text{MSLE} = \frac{1}{n}\sum_{i=1}^{n}(\log(1 + y_i) - \log(1 + \hat{y}_i))^2$$

MSLE is useful for measuring percentage differences between predictions and actual values. Lower MSLE values are better, while higher MSLE values indicate poor model performance.

5.1.6 Median Absolute Error (MedAE)

MedAE measures the median of the absolute errors.

$$\text{MedAE} = \text{Median}(|y_i - \hat{y}_i|)$$

MedAE is robust to outliers. Lower MedAE values indicate more accurate predictions, while higher MedAE values indicate worse model performance.

5.1.7 Max Error (Max Err)

Max Err measures the largest absolute error between actual and predicted values.

$$\text{Max Err} = \max(|y_i - \hat{y}_i|)$$

Max Err indicates the worst-case error. Lower Max Err values are better, while higher Max Err values suggest worse worst-case performance

5.1.8 Explained Variance Score (EVS)

EVS measures how well the variance of the actual data is explained by the model.

$$\text{EVS} = 1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)}$$

EVS close to 1 indicates good model performance, while values close to 0 indicate poor performance.

5.1.9 Mean Absolute Percentage Error (MAPE)

MAPE measures the average percentage error between actual and predicted values.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

MAPE is useful for interpreting errors as a percentage of actual values. Lower MAPE values are better, while higher MAPE values indicate larger percentage errors.

5.1.10 Adjusted R² (R² Adjusted)

Adjusted R^2 accounts for the number of predictors in the model and penalizes adding irrelevant predictors.

$$\text{Adjusted } R^2 = 1 - \left(1 - R^2\right) \frac{n-1}{n-p-1}$$

where n is the number of observations and p is the number of predictors. Adjusted R^2 is useful for comparing models with different numbers of predictors. Higher values close to 1 are good, while values significantly lower than the unadjusted R^2 indicate poor performance.

5.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a method for reducing the dimensionality of data. It reduces the number of variables by creating new, relevant variables that explain the greatest variation in the data.

Advantages of PCA:

- Reduces the number of variables while retaining important information
- Removes noise and irrelevant variations, leading to more robust models
- Simplifies the visualization of high-dimensional data
- Addresses multicollinearity, as the principal components are independent
- Can improve model accuracy and reduce overfitting

5.3 Applied Machine Learning Methods

In this chapter, we introduce and analyze the various machine learning models and algorithms employed in this study, emphasizing their implementation and effectiveness in solving specific issues. We aim to train baseline models using the preprocessed data to predict the CO2 emissions of different types of cars. The target feature for estimation is CO2 (g/km). In the following sections, we present the outcomes of these baseline models.

5.3.1 Linear Regression

The dataset is divided into training and testing sets, where 80% of the data is allocated for training (X_train, y_train) and 20% for testing (X_test, y_test), ensuring reproducibility with a random state of 42. During training, the linear model adjusts its coefficients to achieve the best fit to the data. Because of its simplicity and efficient implementation, linear regression is frequently chosen as a foundational method for subsequent modeling approaches.

Table 8 shows the metrics for PCA compared to direct modeling without PCA.

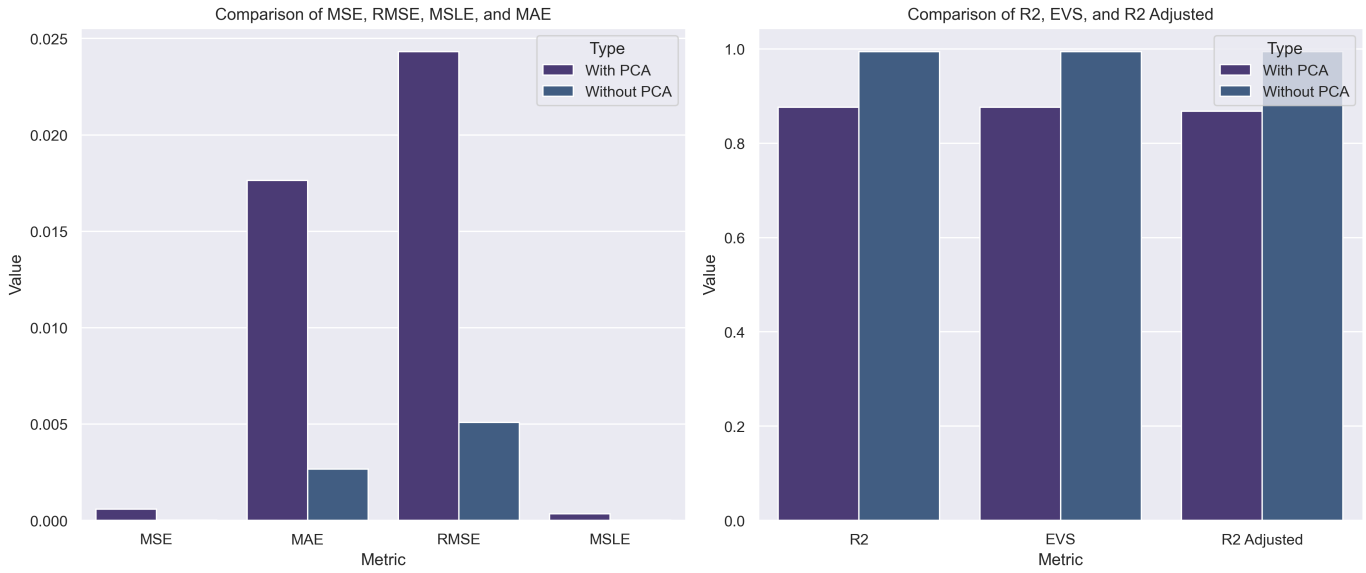

Figure 10: metrics for PCA compared to direct modeling without PCA (Linear Regression).

Without dimensionality reduction, the linear regression model predicts CO2 emissions well, with low MSE, MAE, RMSE, and a high R-squared value. After applying PCA to retain 95% of the variance, the model's application was faster but resulted in higher prediction errors and a decreased R-squared value. Overall, PCA slightly reduced the model's performance compared to the model without dimensionality reduction.

5.3.2 Ridge Regression

Ridge Regression, a form of linear regression with regularization, addresses high correlations between predictor variables and sensitivity to noise. Results from both Ridge Regression with PCA and without PCA demonstrate strong

performance, with an optimized alpha (0.1) improving metrics such as MSE, MAE, RMSE, and R-squared (see Figure 11).Alpha is the regularization parameter optimized through cross-validation.

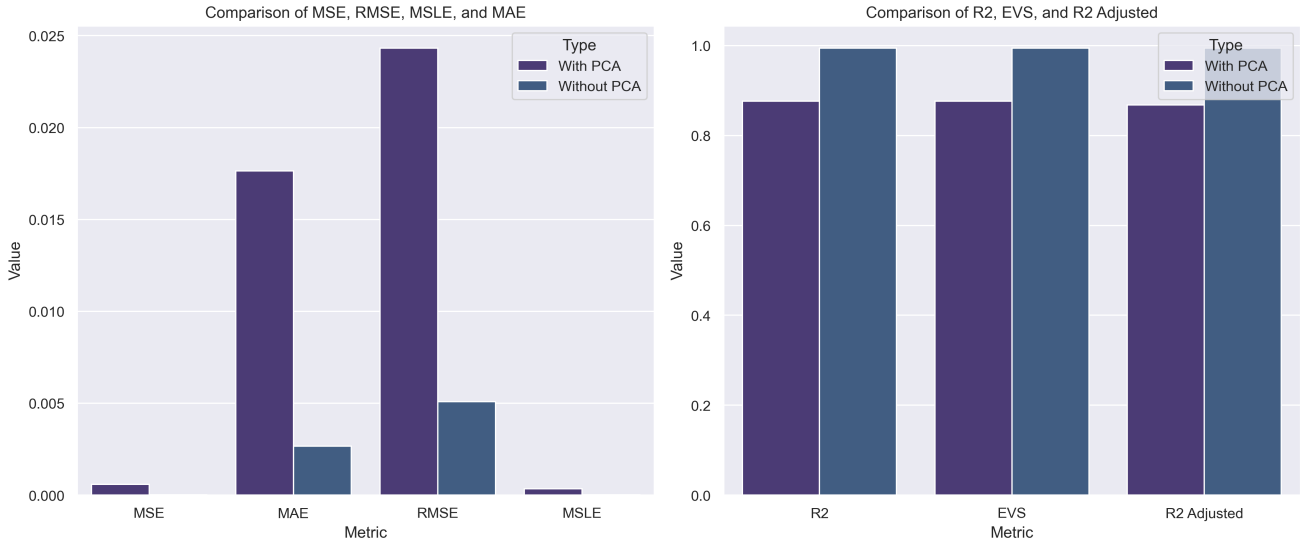

Figure 11: metrics for PCA compared to direct modeling without PCA (Ridge Regression).

Before applying PCA, the model achieved slightly better results, suggesting higher predictive power in estimating CO2 emissions from the original features. While PCA reduces model complexity and computation time, it underscores the trade-offs between performance and efficiency in data analysis.

5.3.3 Lasso Regression

Lasso Regression, also known as L1 regularization, adds a penalty term to the ordinary least squares objective function. This penalty, controlled by the regularization parameter alpha, aims to minimize both the sum of squared residuals and the sum of absolute coefficients, promoting sparsity in the coefficient matrix and performing feature selection by shrinking less important coefficients towards zero.

To visually depict the poor quality of the model, Figure 12 compares the metrics for Lasso Regression to Ridge Regression without PCA.

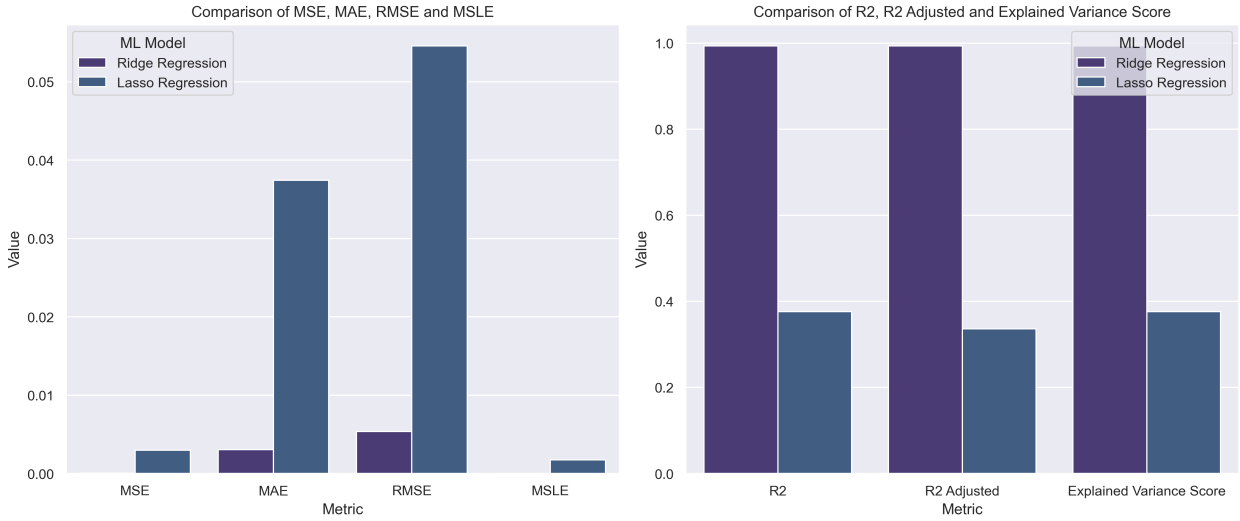

Figure 12: metrics for Lasso Regression compared to Ridge Regression without PCA

This comparison highlights the suboptimal performance of Lasso Regression in our project, despite tuning alpha to an optimal value of 0.01. The low R-squared value ($R^2 = 0.3729$) and high error metrics underscore the challenges in effectively capturing the relationship between features and the target variable using Lasso Regression in this context.

5.3.4 ElasticNet Regression

ElasticNet Regression combines Ridge and Lasso Regression to leverage their respective penalty terms. The initial results showed low error values (MSE, MAE, RMSE) but a low R^2 value, indicating poor model performance. After hyperparameter tuning with GridSearchCV, which determined the best values for alpha (0.1) and l1_ratio (0.1), the model's performance improved. This optimization resulted in an MSE of 0.0033 and an R^2 value of 0.3003. Although the model performs better than before, the relatively low R^2 value still indicates room for improvement. Figure 13 compares the poor quality of the ElasticNet Regression metrics with the Ridge Regression without PCA. However, this model is not suitable for our project.

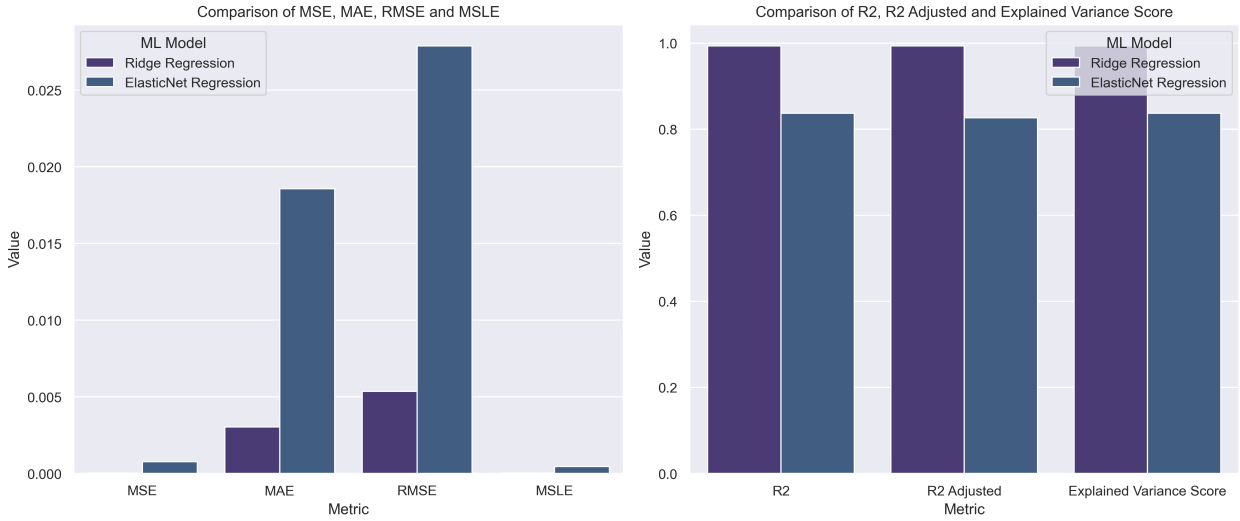

Figure 13: metrics for ElasticNet Regression compared to Ridge Regression without PCA

5.3.5 Decision Trees

Decision Trees are a widely favored algorithm for regression tasks due to their ability to effectively handle non-linear relationships and feature interactions. In our analysis, we chose to apply this model to our dataset because of its versatile nature and consistently strong performance across various scenarios. Decision Trees work by recursively partitioning the data into subsets based on features, aiming to minimize prediction error or maximize information gain at each split.
Figure 14 presents the results of the performance comparison between Decision Trees, Decision Trees with PCA, and Ridge Regression.

Decision Trees exhibit the best performance across all metrics, demonstrating very low error values (MSE, MAE, RMSE, MSLE) and very high fit metrics (R2, R2 Adjusted, Explained Variance Score).

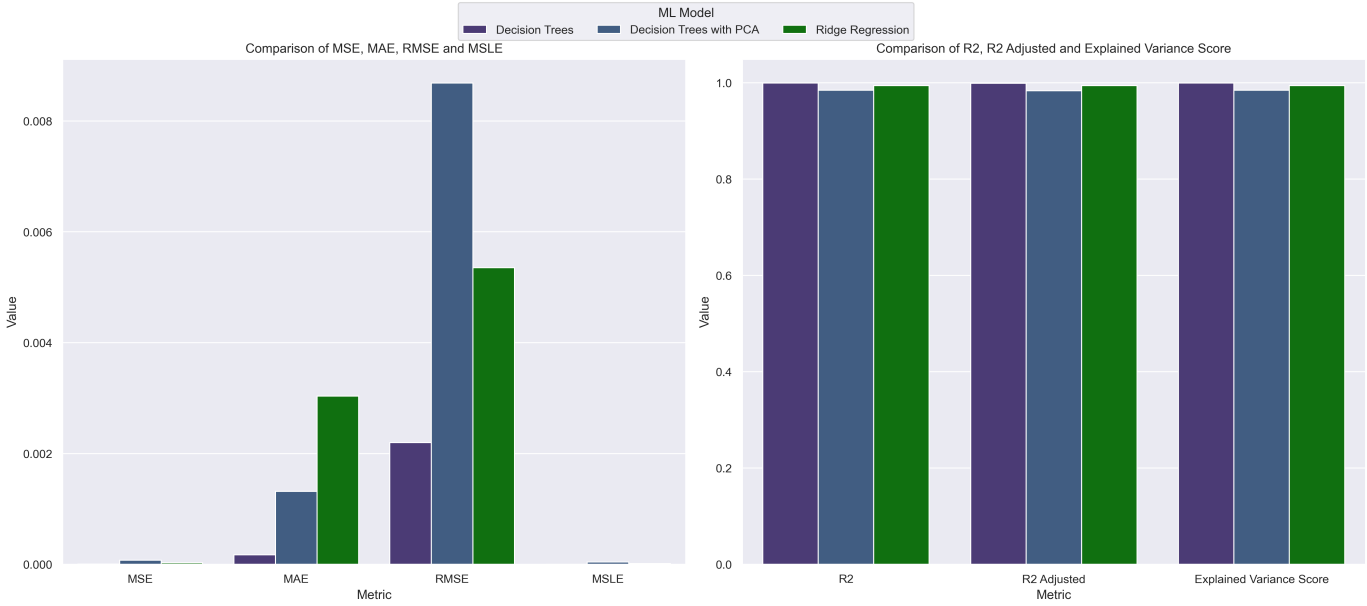

Figure 14: Comparison of Metrics between Decision Trees, Decision Trees with PCA, and Ridge Regression

Decision Trees with PCA also achieve good results, albeit slightly trailing behind regular Decision Trees, particularly in error metrics. Ridge Regression, currently the preferred model, also shows solid performance but falls slightly behind Decision Trees, especially in terms of error metrics. Overall, all models demonstrate strong performance, with Decision Trees achieving the best results, followed by Ridge Regression, which was included in the comparison as it currently stands as our best model. In third place are the Decision Trees with PCA.

5.4 Bagging Algorithms

Bagging algorithms such as Random Forest, Bagged Decision Trees, and Bagged SVM are ensemble techniques aimed at improving model prediction accuracy and robustness.

- **Random Forest**: Utilizes a group of decision trees trained on random subsets of the training data to reduce overfitting and improve prediction accuracy, especially in complex datasets. Combining predictions from multiple trees also enhances robustness against outliers.
- **Bagged Decision Trees**: Similar to Random Forest, trains multiple decision trees on random subsets of data to reduce variability and over-

fitting, leading to better generalization, particularly beneficial for deep and complex decision trees.

- **Bagged SVM**: Applies bagging to Support Vector Machines to enhance robustness and prediction accuracy. By training multiple SVM models on different random subsets of data, it increases prediction stability and reduces sensitivity to noise and outliers.

Figure 15 compares performance metrics for various machine learning models using different Bagging algorithms. Random Forest shows the lowest values for MSE, MAE, RMSE, and MSLE, indicating lower susceptibility to errors and higher prediction accuracy. Additionally, Random Forest achieves the highest values for R2, R2 Adjusted, and Explained Variance Score, suggesting it best explains the data and makes the most accurate predictions.

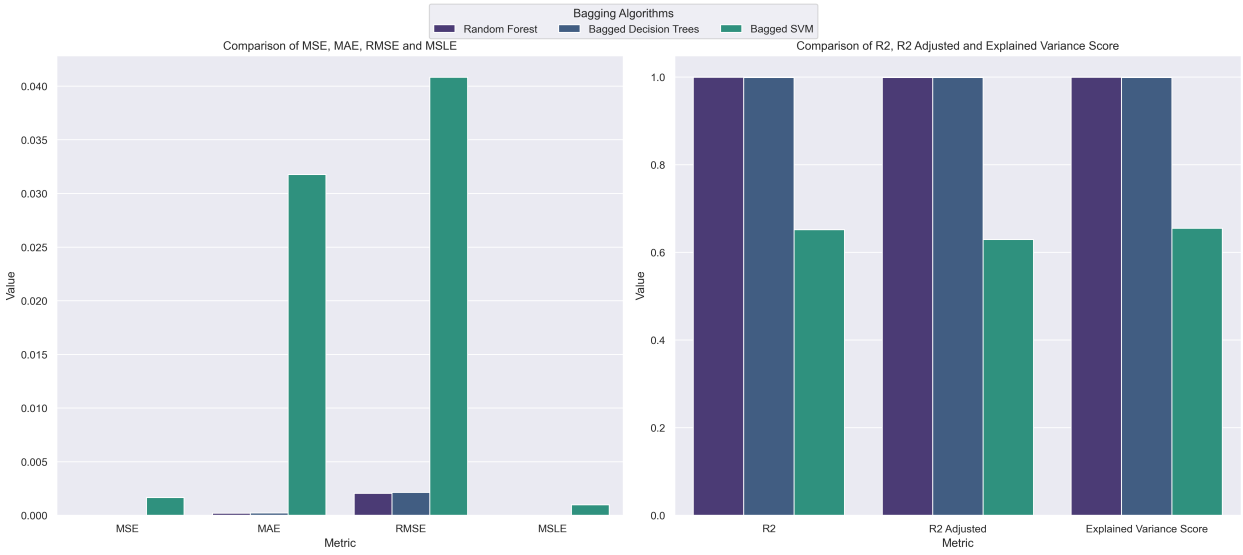

Figure 15: comparison of performance metrics for various bagging algorithms

Bagged Decision Trees demonstrates similar performance to Random Forest, although metric values are slightly lower. Nonetheless, it delivers robust results and reduces overfitting.

In contrast, Bagged SVM exhibits comparatively higher error metrics and lower determination measures, indicating it may be less suitable for this specific task compared to the other two models.

In summary, up to this point, the Random Forest model stands out as the best performer among the models considered, showcasing overall lower error metrics and higher determination scores for the task of prediction accuracy and model explanation.

5.5 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.

- **AdaBoost**: Is for Adaptive Boosting, combines multiple weak learners, typically decision trees, to create a strong learner. The key feature of AdaBoost is that it adjusts the weights of misclassified instances, putting more focus on difficult examples in subsequent iterations.
- **Gradient Boosting**: Sequentially builds a series of decision trees, where each new tree is trained to correct the residual errors of the previous trees. It fits a model to the negative gradient of the loss function to minimize the overall error. Gradient Boosting is flexible and can be used with different types of loss functions.
- **XGBoost**: For Extreme Gradient Boosting, is an optimized and scalable version of Gradient Boosting. It includes advanced features like regularization to prevent overfitting, parallel processing, and efficient handling of missing values. XGBoost is known for its high performance and speed, making it a popular choice in machine learning competitions.

Based on the results shown in Figure 16, XGBoost is the best model for the given task. It demonstrates the lowest values in error metrics (MSE, MAE, RMSE, MSLE) and the highest values in determination measures (R2, R2 Adjusted, Explained Variance Score), indicating the highest accuracy and predictive power among the boosting algorithms considered.

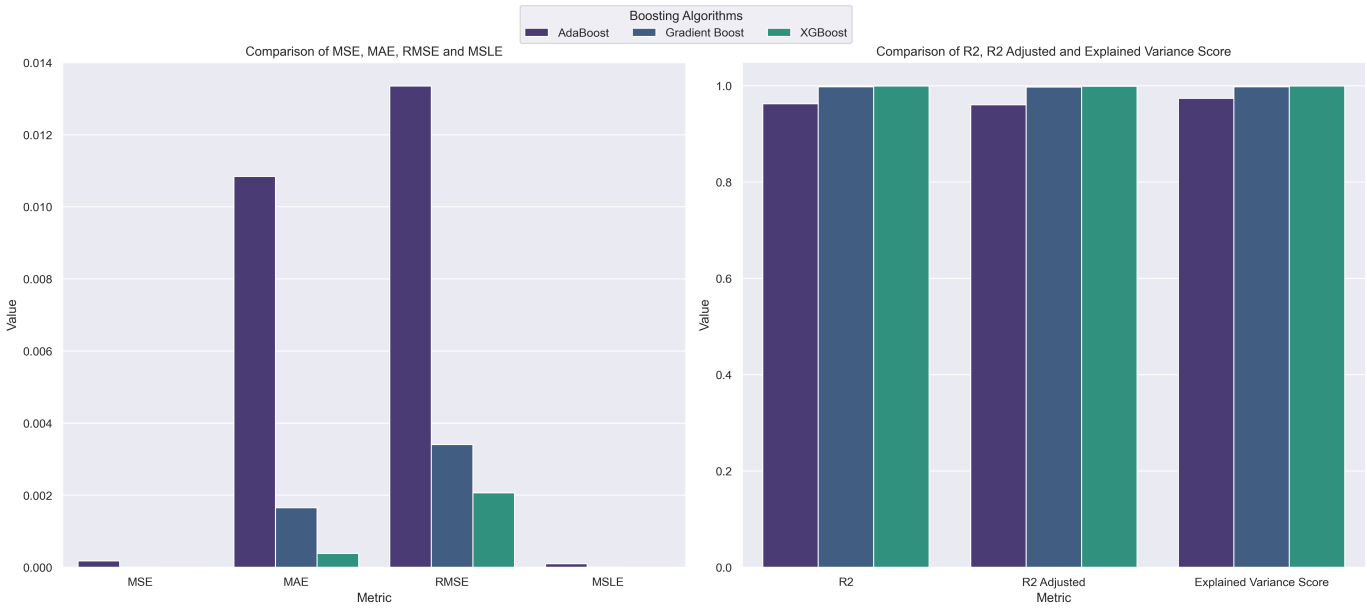

Figure 16: Comparison of performance metrics for various boosting algorithms

5.6 Deep Learning

Deep learning models, like neural networks, are a type of artificial neural network characterized by multiple layers of neurons and non-linear activation functions. These models can capture complex patterns in data and are suitable for tasks such as regression and classification. They are trained with large datasets to extract features, learn patterns, and make predictions. Key terms include epochs (iterations over the dataset during training), batch size (number of data samples propagated through the network before updating weights), learning rate (step size in gradient descent), loss function (evaluates model performance by measuring the difference between predicted and actual values), activation function (transforms neuron inputs to introduce non-linearity), and optimizer (algorithm adjusting network weights based on computed gradients). These parameters are crucial for training neural networks effectively.

In unserem Projekt haben wir mehrere Modelle angewendet:

- Multi-layer Perceptron (MLP):A basic neural network with multiple layers of neurons known for its ability to capture complex patterns in data.
- Convolutional Neural Network (CNN): Specialized for processing sequential data such as time series or images, characterized by its architecture of convolutional and pooling layers.

- Recurrent Neural Network (RNN): Designed for sequential data processing by maintaining a state across multiple time steps, making it ideal for tasks involving time series and textual data.

We also attempted these models with an Early Stopping function. All models were trained with a batch size of 32 and used ReLU activation function uniformly across layers. Below is a table summarizing the specific parameters these models:

Parameter	Value
Epochs	100
Batch Size	32
Learning Rate	Default ('adam')
Loss Function	'mse' (Mean Squared Error)
Activation Function	'relu' (Rectified Linear Unit)
Optimizer	'adam'
Early Stopping	Monitor='val_loss', Patience=10, Restore_best_weights=True

Table 8: Model Configuration Parameters

The next figure illustrates a comparison of performance metrics for various deep learning techniques.

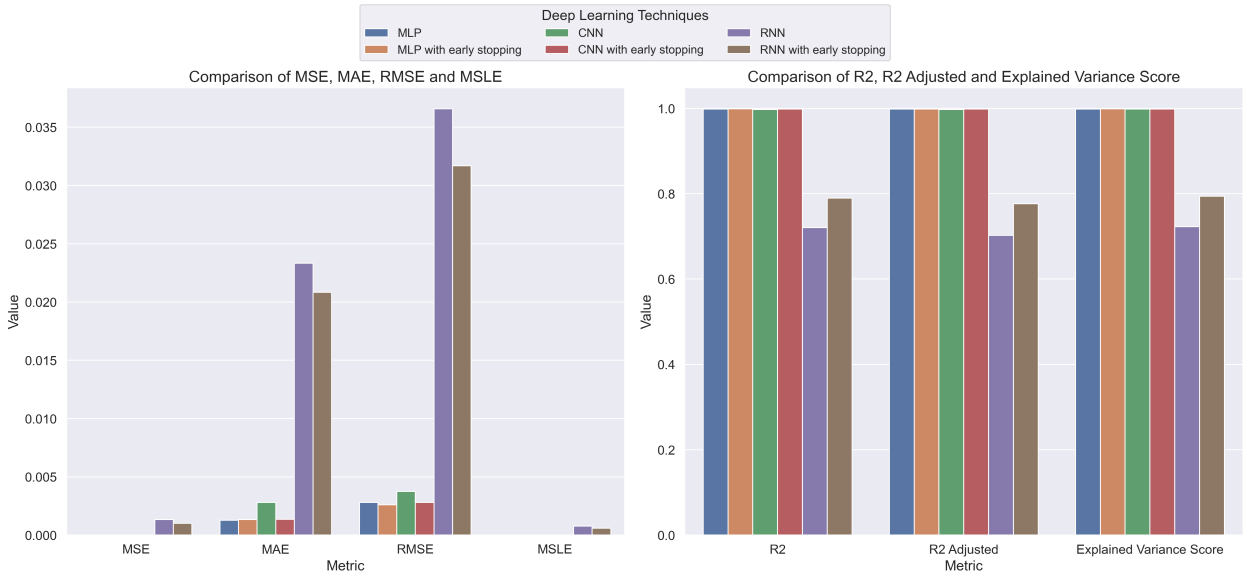

Figure 17: Comparison of performance metrics for various Deep Learning Techniques

Among the models, the MLP model emerges as the top performer, achieving an MSE of 8.00E-06 and an R2 of 0.9984. Following closely behind are models like MLP with Early Stopping, CNN, and CNN with Early Stopping, all demonstrating strong performance across most metrics, notably MSE and R2.

On the other end of the spectrum, the RNN with Early Stopping shows relatively lower performance with an MSE of 0.0010 and an R2 of 0.7763 compared to the other models evaluated.

5.7 Summary of the modeling

In summary of the modeling comparison across multiple metrics, the analysis reveals the top-performing models distinguished by their consistently superior performance in MSE, MAE, RMSE, and R2 metrics. These models are listed as follows (see the next table):

Rank	Model	MSE	MAE	RMSE	R2
1	Random Forest	4.00E-06	0.000201	0.002062	0.99911
2	XGBoost	4.00E-06	0.000387	0.002066	0.999107
3	Bagged Decision Trees	5.00E-06	0.000219	0.00213	0.999051

Table 9: Performance metrics of different models with rankings

In contrast, the model showing the least performance in this comparison is ElasticNet Regression, followed by deep learning models such as MLP, MLP with Early Stopping, CNN, CNN with Early Stopping, RNN, and RNN with Early Stopping. These models exhibit relatively lower scores across different metrics compared to other model types, indicating less accurate predictions. For detailed information on metrics for all models, please refer to the 'Performance Metrics Table' in the Appendix.

6 Interpretation of results

In this section, we present the results of applying interpretability tools on the top and good performing models, including Random Forest, XGBoost and Bagged Decision Trees. It is generally most effective to apply interpretability tools primarily on the best-performing models since applying these tools to every model can be time-consuming and resource-intensive. Interpreting the results using interpretability tools is a crucial step to understand the underlying relationships and ensure that the models make reasonable predictions.

In the following sections of this chapter, we will present the tools we used in this project and their results.

6.1 Feature Importance

Feature Importance measures the contribution of each feature to the model's prediction. It helps identify which features have the greatest impact on the target variable. This metric is used to rank features and identify the most important ones. Higher importance scores indicate more influential features. Figure 18 Shows a Comparison of Feature Importance Among Our Top 3 Models.

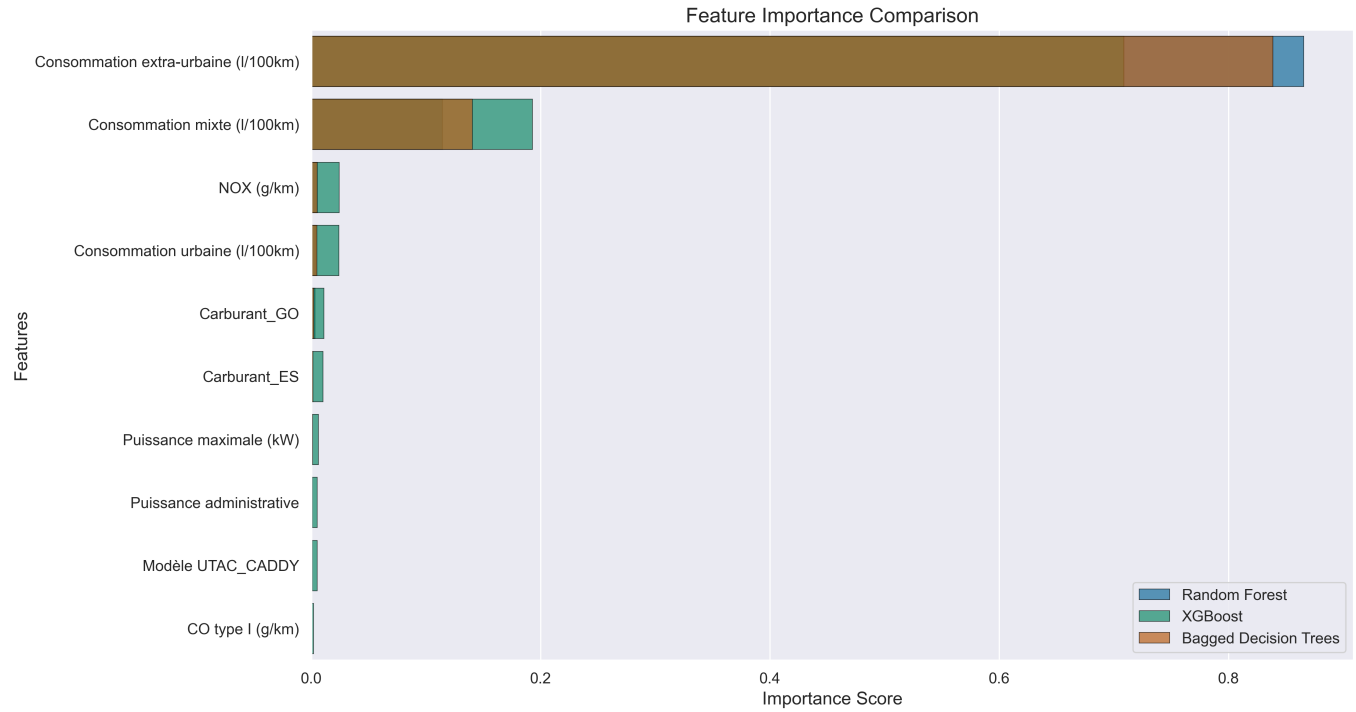

Figure 18: Comparison of feature importance

Based on this diagram, Consommation extra-urbaine (extra-urban fuel consumption) emerges as the most critical feature across all models, indicating its significant impact on CO2 emissions. Cars with higher extra-urban fuel consumption tend to emit more CO2. Consommation mixte (mixed fuel consumption) follows closely, highlighting the importance of overall fuel efficiency in CO2 emissions. The variations in other feature importance values across models underscore how each model interprets and applies these relationships. Features with minimal importance suggest they contribute little to

predicting CO2 emissions in cars. In summary, these findings underscore the pivotal role of fuel consumption, especially in extra-urban and mixed driving conditions, in determining CO2 emissions.

6.2 SHAP (SHapley Additive exPlanations)

SHAP values provide a unified measure of feature importance and impact on model predictions by assigning an importance value to each feature for a particular prediction. Based on game theory, SHAP values explain the contribution of each feature by computing the change in the prediction when the feature is added to a set of other features. This tool provides both local (individual predictions) and global (overall model behavior) explanations. Positive SHAP values indicate features that increase the prediction, while negative values indicate features that decrease it.

The results of applying this tool to the models are depicted in Figure 19.

Due to the computational intensity involved in processing our extensive dataset, we opted to sample the data and apply the method accordingly. Specifically, we randomly selected a subset of 1000 rows from our dataset for analysis. Figure 19 shows the results of the SHAP application in the example of the Random Forest model.

Based on these results, we can draw the following conclusions:

Fuel Consumption Dominance: The high SHAP values for features related to fuel consumption indicate that predictions from the models (Random Forest, XGBoost, and Bagged Decision Trees) are highly sensitive to changes in fuel efficiency. This highlights the critical role of fuel consumption in influencing model outputs, whether in terms of vehicle performance or environmental impact.

Pollutant Emissions: While features like NOX (g/km) and HC+NOX (g/km) show lesser influence compared to fuel consumption-related features, their presence in the top SHAP values underscores their significant contribution to predicting emissions within these models.

Consistency Across Models: The shared importance of these features across Random Forest, XGBoost, and Bagged Decision Trees indicates their reliability and relevance in predicting outcomes across different predictive models.

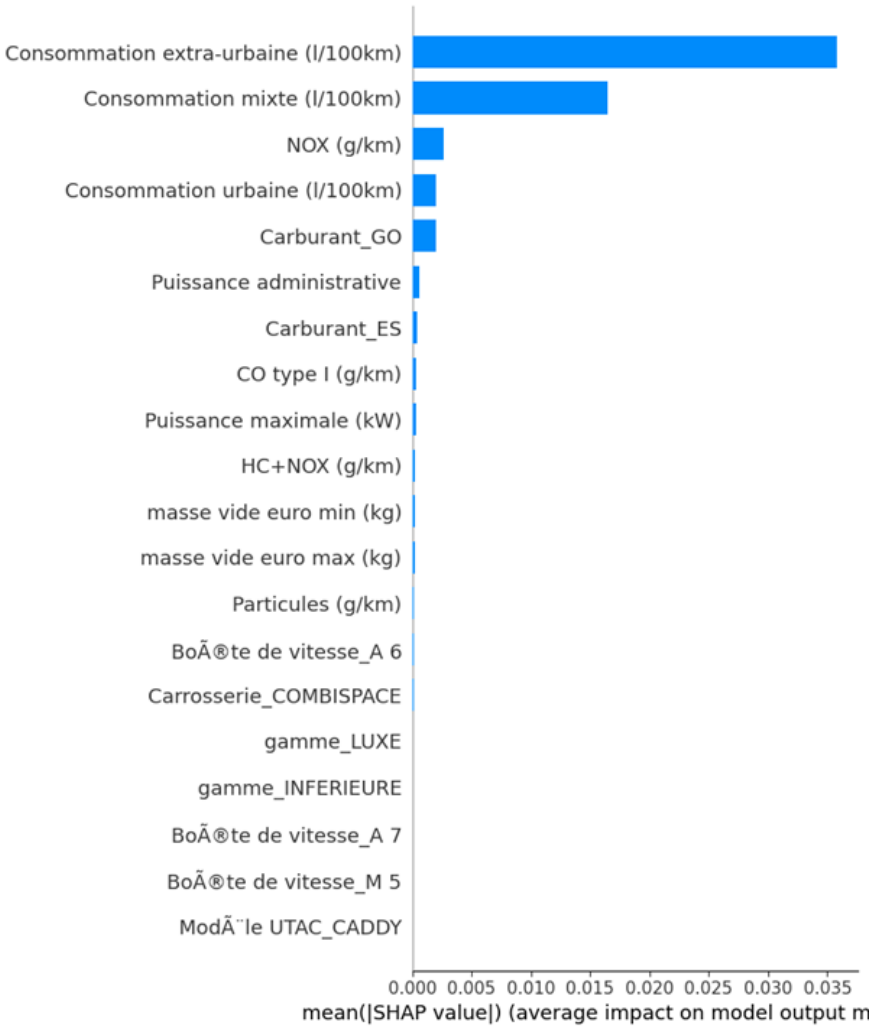

Figure 19: Results of SHAP on Random Forest

6.3 LIME (Local Interpretable Model-agnostic Explanations)

LIME (Local Interpretable Model-agnostic Explanations) is a method used to explain individual predictions of a model without requiring specific knowledge about the model itself. LIME approximates the model locally with an interpretable model to explain individual predictions. It perturbs the data around the instance to be explained and fits an interpretable model such as linear regression to locally approximate the black-box model. LIME provides weights for each feature for a single prediction, showing how much each feature contributes to that prediction. It is useful for understanding specific predictions rather than interpreting the entire model.

Figure 20 shows the results of the LIME application in the example of the Random Forest model.

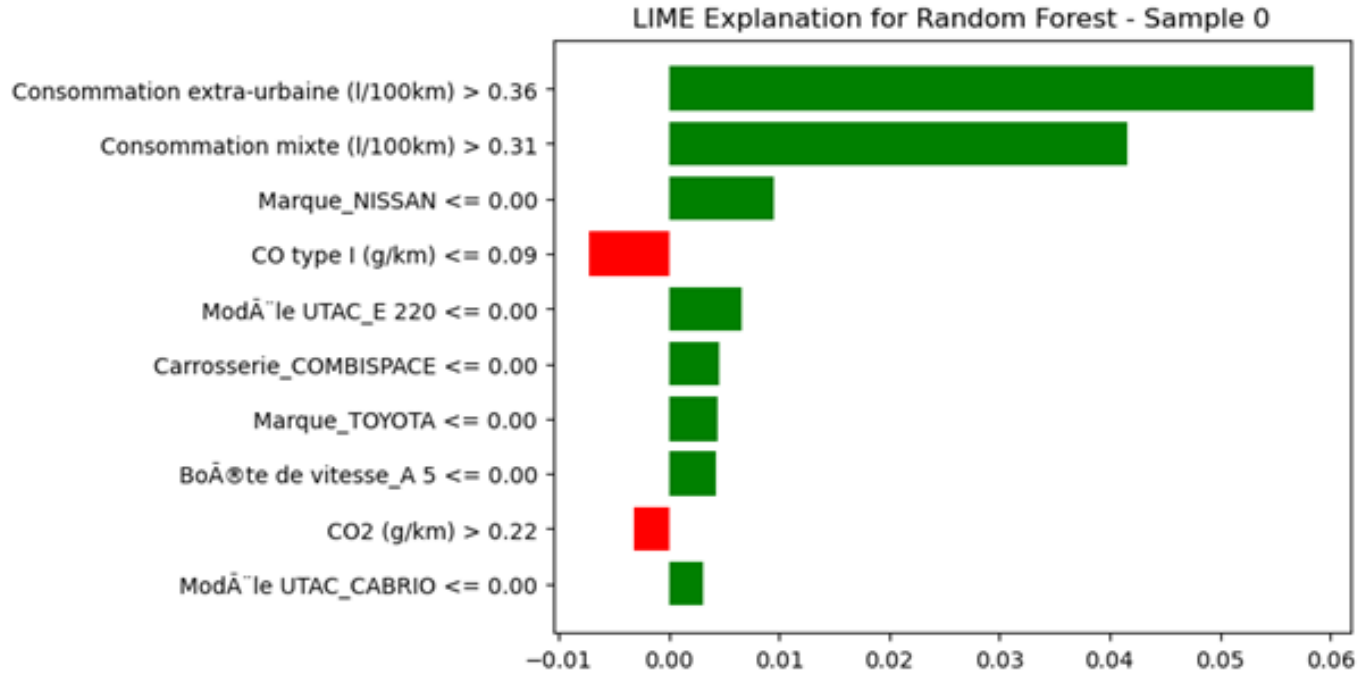

Figure 20: Results of LIME on Random Forest

Across all models, the most significant features influencing the predictions are:

Consommation extra-urbaine (l/100km): This feature consistently shows a strong positive impact on the predictions.
Consommation mixte (l/100km): Similarly, this feature also consistently shows a strong positive impact. CO type I (g/km): Generally has a negative impact.
Marque NISSAN and Marque TOYOTA: Show positive impacts in certain models.
Carrosserie CABRIOLET and Modèle UTAC E 220: Often have negative impacts.

These consistent patterns across models highlight the significant influence of fuel consumption metrics and specific car brands and models on the target variable. The consistent importance of these features suggests that optimizing fuel consumption and understanding the impact of car types and brands can be crucial for improving model performance and making accurate predictions.

6.4 Partial Dependence Plots (PDP)

PDPs show the relationship between a feature and the predicted outcome, averaging out the effects of all other features. For each value of the feature, the model's predictions are averaged over all instances to show how the feature affects the predictions. PDP shows how the predicted outcome changes as a feature changes, holding other features constant.

Figure 21 shows the results of the PDP application in the example of the Random Forest model

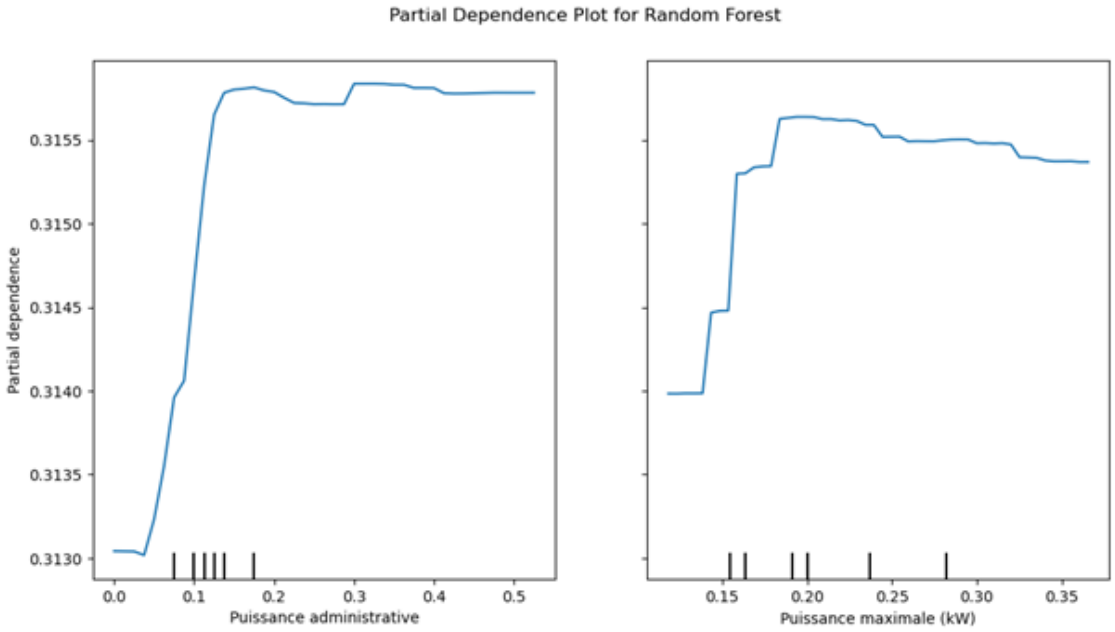

Figure 21: Results of PDP on Random Forest

As there are many features in our dataset, we selected two first features, Puissance administrative and Puissance maximale (kW)

Based on our analysis of the PDP and LIME diagrams for two features impacting CO emissions, here are the key findings:

Fuel Consumption: Higher extra-urban and mixed fuel consumption strongly predict higher CO emissions. This suggests that vehicles with greater fuel consumption tend to emit more CO.

Vehicle Characteristics: Certain vehicle models (like UTAC_E 200 and UTAC_E 220) and body styles (such as Cabriolet and Break) consistently influence CO emissions. This indicates that design choices in vehicles play a significant role in emissions levels.

Transmission Types: Specific transmission types, such as Boîte de vitesse_D

7, impact CO emissions, highlighting the importance of transmission technology in vehicle emissions.

Car Brands: LIME explanations reveal that different car brands, such as Nissan and Toyota, are associated with varying emission levels. This variability may stem from technological and design differences between brands.

6.5 Correlation Analysis

Given the large number of attributes, the heatmap becomes cluttered and difficult to interpret. To improve readability, we tried three approaches: clustering the heatmap by grouping similar features, selecting a subset of features, and filtering by correlation threshold. The results of selecting a subset of features were more readable than the other two approaches. Figure 22 shows the heatmap of the top 15 features.

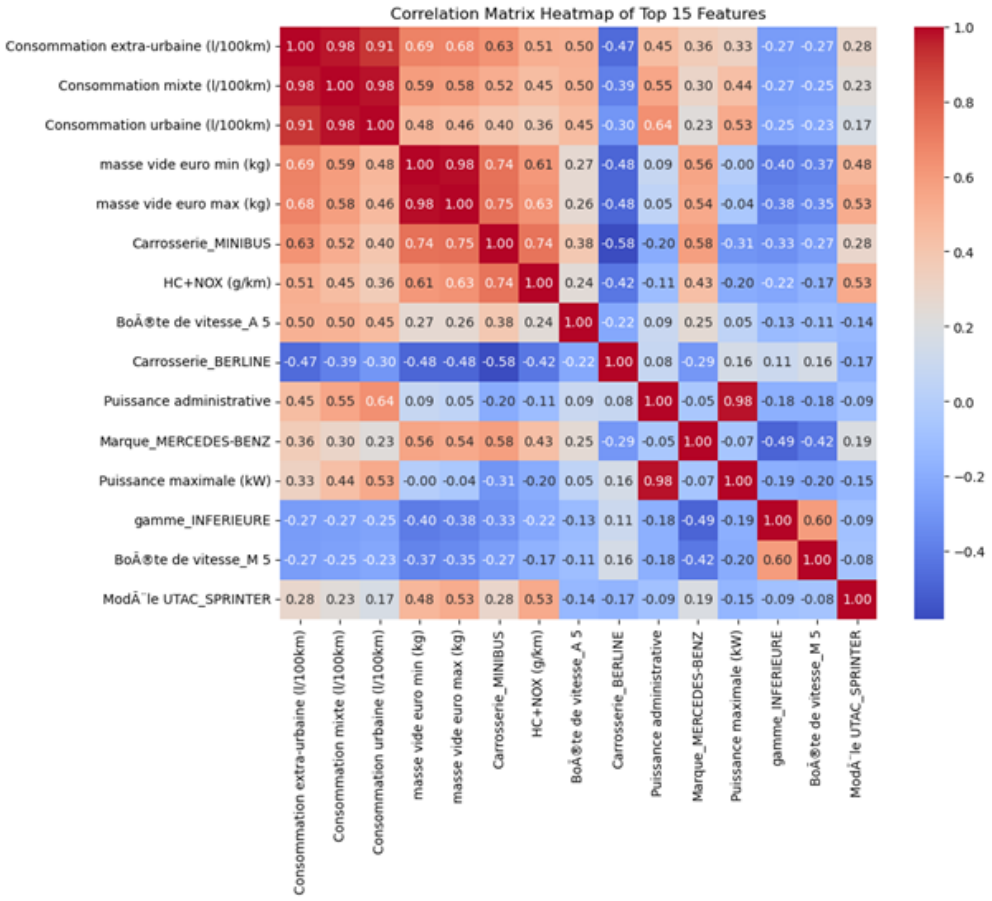

Figure 22: Correlation matrix heatmap of top 15 features

The following table presents key correlations between various features re-

lated to vehicle data. These correlations were identified through the analysis of a heatmap, providing insights into the relationships among fuel consumption, vehicle weight, body types, brands, and performance characteristics. Table 10 displays the strength of these correlations, along with additional comments for each relationship based on the analyzed data.

Feature	Correlation	Comment
Consommation extra-urbaine (l/100km)	0.98-1.00	Very strong positive correlation with Consommation mixte and Consommation urbaine.
Consommation mixte (l/100km)	0.98-1.00	Very strong positive correlation with Consommation extra-urbaine and Consommation urbaine.
Consommation urbaine (l/100km)	0.98-1.00	Very strong positive correlation with Consommation extra-urbaine and Consommation mixte.
Masse vide euro min (kg)	0.91	High positive correlation with Masse vide euro max.
Masse vide euro max (kg)	0.91	High positive correlation with Masse vide euro min.
Carrosserie_MINIBUS	0.75	Positive correlation with HC+NOX (g/km), indicating higher emissions for minibuses.
Marque_MERCEDES-BENZ	0.55	Moderate positive correlation with Puissance maximale (kW), suggesting Mercedes-Benz models have higher power.
Carrosserie_BERLINE	Negative	Negative correlation with various consumption metrics, indicating lower fuel consumption for sedans.
gamme_INFERIEURE	-0.45	Moderate negative correlation with Puissance maximale (kW), indicating lower-end models have less power.
Puissance administrative	0.54	Positive correlation with Puissance maximale (kW), administrative power as an indicator of maximum power.
Boîte de vitesse_A 5	Moderate	Moderate correlation with various variables, common gearbox type across different vehicle types.

Table 10: Correlations between Vehicle Features

7 Conclusion

Analyzing the evaluation results indicated in Table 12, it is clear that five models emerged as top performers: Random Forest, Bagged Decision Trees, XGBoost, Gradient Boosting, and Decision Trees. The ensemble methods (Random Forest, Bagging, Gradient Boosting, and XGBoost) provided better performance and robustness compared to the single Decision Tree model. While single models like Decision Trees offer more interpretability, ensemble methods deliver superior predictive performance at the cost of interpretability. Therefore, the choice of model depends on the specific needs of the application: Decision Trees for quick interpretability in small to medium-sized datasets, Random Forest for balanced performance, Gradient Boosting and XGBoost for high accuracy in resource-rich environments, and Bagging methods for reducing variance and improving robustness.

By applying interpretability tools on these five models, several key insights were derived. Firstly, the models consistently identified extra-urban fuel consumption (Consommation extra-urbaine) and mixed fuel consumption (Consommation mixte) as the most significant predictors of CO2 emissions. This highlights the critical role of fuel consumption in determining emissions, suggesting that automotive companies should optimize fuel efficiency in both extra-urban and mixed driving conditions. Additionally, emission-related features such as NOX (g/km) and HC+NOX (g/km) were also significant, though less influential than fuel consumption, indicating that continued investment in emission reduction technologies is crucial for regulatory compliance and market competitiveness. The consistency in the importance of these features across different models underscores their robustness and reliability in predicting CO2 emissions.

However, there are limitations and areas for improvement. Many features exhibited low importance, suggesting minimal impact on model predictions and indicating a need to re-evaluate the data collection process to focus on high-impact features. The high importance of a few features also poses a risk of overfitting, potentially neglecting other relevant factors. Implementing regularization techniques and thorough cross-validation can mitigate these risks. Strategic recommendations include prioritizing research and development in

fuel efficiency technologies, continuing efforts to reduce vehicle emissions, leveraging data analytics for continuous improvement, and highlighting advancements in fuel efficiency and low emissions in marketing campaigns to attract environmentally conscious consumers. By following these recommendations, automotive companies can drive sustainability, meet regulatory requirements, and maintain a competitive edge in the market, contributing to overall environmental health.

References

1. Introduction to Statistical Learning, James et al.
2. `https://datatab.net/tutorial/dispersion-parameter`
3. `https://learn.datascientest.com`

Performance Metrics Table

ML Model	MSE	MAE	RMSE	R2	MAPE	MedAE	Max Error	MSLE	R2 Adjusted	Explained Variance Score
Linear Regression	2.6e-05	0.00266	0.005086	0.99459	0.00973	0.001631	0.091257	1.6e-05	0.994244	0.99459
Linear Regression with PCA	0.000592	0.017638	0.024329	0.876186	0.062014	0.013498	0.170917	0.000351	0.868263	0.876187
Ridge Regression	2.9e-05	0.003036	0.005351	0.994011	0.010805	0.002148	0.09206	1.7e-05	0.993628	0.994011
Ridge Regression with PCA	2.7e-05	0.002803	0.005201	0.994342	0.010141	0.001923	0.092525	1.6e-05	0.99398	0.994342
Lasso Regression	0.002982	0.037452	0.054605	0.376282	0.140266	0.023212	0.237502	0.001758	0.336371	0.376283
ElasticNet Regression	0.000778	0.018575	0.027892	0.83726	0.06638	0.011893	0.186258	0.000448	0.826846	0.837264
Decision Trees	5e-06	0.000171	0.002197	0.99899	0.000686	0.0	0.170642	3e-06	0.998925	0.99899
Decision Trees with PCA	7.5e-05	0.001316	0.008683	0.98423	0.005516	0.0	0.255046	4.5e-05	0.983221	0.98423
Bagging Algorithms										
Random Forest	4e-06	0.000201	0.002062	0.99911	0.000851	0.0	0.152991	3e-06	0.999053	0.99911
Bagged Decision Trees	5e-06	0.000219	0.00213	0.999051	0.000938	0.0	0.150826	3e-06	0.998991	0.999051
Bagged SVM	0.001666	0.031755	0.040816	0.65152	0.115697	0.021914	0.103569	0.000986	0.629221	0.654758
Boosting Algorithms										
AdaBoost	0.000178	0.010845	0.013347	0.962735	0.035307	0.008316	0.131194	0.000102	0.960351	0.973949
Gradient Boost	1.2e-05	0.001656	0.003405	0.997574	0.006196	0.00088	0.099021	7e-06	0.997419	0.997575
XGBoost	4e-06	0.000387	0.002066	0.999107	0.001576	7.8e-05	0.146001	3e-06	0.99905	0.999107
Deep Learning Techniques										
MLP	8e-06	0.001266	0.002804	0.998355	0.00474	0.000894	0.123528	5e-06	0.99825	0.998398
MLP with early stopping	7e-06	0.001342	0.002601	0.998584	0.004898	0.000915	0.11706	4e-06	0.998494	0.99863
CNN	1.4e-05	0.002796	0.003738	0.997078	0.009604	0.002198	0.113691	8e-06	0.996891	0.998471
CNN with early stopping	8e-06	0.001356	0.00279	0.998371	0.005014	0.000813	0.084608	5e-06	0.998267	0.998376
RNN	0.001337	0.023328	0.036571	0.720235	0.081905	0.015557	0.275887	0.000775	0.702333	0.722463
RNN with early stopping	0.001005	0.020826	0.031701	0.789776	0.070289	0.01532	0.258336	0.000581	0.776325	0.794183

Table 12: Performance Metrics for Various Machine Learning Models