1. 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.

1.1. Linear Regression

As the first model, we trained the Linear Regression model. The results of evaluating this model are presented in Table 1 and Table 2. Table 1 shows the results based on the data cleaned through the first preprocessing step. Table 2 shows the evaluations results of all the models based on the data cleaned through the second preprocessing step.

1.1.1. 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.

1.1.2. 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.

1.2. 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.

1.2.1. 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 the cross validation. These are the results of the application of the model with 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.

1.2.2. 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.

1.3. 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.

1.4. 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': 0.1}
- Mean Squared Error (MSE): 0.0033337131647687627
- R-squared (R2): 0.30028459159136967

With these hyperparameters, the ElasticNet model achieved an MSE of approximately 0.0033 and an R2 value of approximately 0.3003 on the test set. Overall, while the model's performance improved compared to the previous results, it still may not be satisfactory, especially considering the relatively low R-squared value.

Therefore, this model is not a good model for our project.

1.5. Decision Trees

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

1.5.1. 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.**

1.5.2. 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.

1.6. 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.

1.7. 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 on 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.

1.7.1. 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.

1.7.2. 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.

1.7.3. 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.

1.8. 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.

1.8.1. 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 perform 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.

1.8.2. 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 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.

1.8.3. 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 perform 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.

1.9. 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.

1.9.1. 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.

1.9.2. 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 to see which one yields the best results for our specific use case.

The results of applying this technique are shown in Table 2. These results show that this model performs also very well on the task. But, the problem is that training this model for 100 Epoc lasts more than 30 minutes. Therefore, to select between multiple models, we need to consider other factors, including

1. **Model Complexity:** CNNs are typically more complex than MLPs, as they are specifically designed for handling spatial data such as images.
2. **Interpretability:** MLPs often provide more straightforward interpretability compared to CNNs, as they consist of densely connected layers.
3. **Computational Efficiency:** MLPs are generally **faster to train** compared to CNNs, especially for tabular data. If computational resources are limited, an MLP might be a more practical choice.

We also applied the early stopping technique to improve efficiency of the model training. The model training stopped after 32 epochs (~ 14 min). The evaluation results are shown in Table 2.

1.9.3. Recurrent Neural Networks (RNNs)

RNNs are designed to handle sequential data by maintaining a state across time steps. They are commonly used in natural language processing (NLP) tasks, time series analysis, and sequence prediction. The results of applying this technique are shown in Table 2. This model shows a decent performance with an R^2 value of around 0.72, suggesting it explains 72% of the variance. However, when compared to the MLP and CNN models, the RNN performs less effectively on this dataset. MLP and CNN both achieve R^2 values above 0.99, indicating a much better fit to the data. Thus, MLP and CNN are likely more suitable for this particular regression task than RNN.

To improve the efficiency of the model training, we applied the early stopping technique. The model training stopped after 31 epochs (~ 34 min). The evaluation results are shown in Table 2.

2. An overview of the results of all models and their interpretation

This section shows an overview of the results for different models we tried on the dataset. In the first pre-processing step, we applied mean, median, mode, and KNN method to handle missing values. Table 1 shows the results of applying the first six ML model on the cleaned dataset by using these methods. In the next step, we decided to use KNN to handle missing values in all numerical columns. Table 2 shows the results of applying ML models to the dataset cleaned by using this method.

As can be seen in Table 1, the results of ElasticNet Regression are worse than Ridge Regression. As ElasticNet is a combination of Ridge Regression and Lasso Regression, we expect that the results be at least as good as one of the two models. To improve the results of ElasticNet, we use **GridSearchCV** to search for the best combination of **alpha** and **l1_ratio** hyperparameters for ElasticNet Regression using cross-validation. The best hyperparameters are: {'alpha': 0.01, 'l1_ratio': 0.1. The results of the ElasticNet model by using these hyperparameters are shown in Table 2.

We also added some further metrics in Table 2, including:

1. **Mean Absolute Percentage Error (MAPE)**: MAPE measures the average absolute percentage difference between the predicted and actual values. It's particularly useful when you want to understand the magnitude of errors relative to the actual values.
2. **Median Absolute Error**: Similar to MAE, but using the median instead of the mean. It's less sensitive to outliers compared to MAE.
3. **Explained Variance Score**: You already have this metric, but it's worth mentioning. Explained Variance Score measures the proportion of variance in the target variable that is explained by the model. It ranges from 0 to 1, where 1 indicates perfect prediction.

4. **Max Error**: Max Error calculates the maximum residual (absolute difference between predicted and true values) across all samples. It gives you an idea of the worst-case error of your model.
5. **Mean Squared Logarithmic Error (MSLE)**: MSLE measures the mean of the squared differences between the natural logarithm of the predicted and true values. It's useful when the target variable has exponential growth.
6. **R-squared Adjusted (R2 Adjusted)**: R2 Adjusted adjusts the R-squared value for the number of predictors in the model. It penalizes model complexity, providing a more reliable measure of goodness-of-fit when comparing models with different numbers of predictors.

Table 1. Results of applying six ML models after the first preprocessing step

Metric / ML Model	Mean Squared Error (MSE)	Mean Absolute Error (MAE)	Root Mean Squared Error (RMSE)	R-squared (R2)
Linear Regression	2.6485711266259274e-05	0.0025955619001083283	0.005146427038855139	0.9944408953741078
Linear Regression with PCA	0.0005988825887608515	0.017800248244260315	0.024472077736899488	0.8743001108757107
Ridge Regression	2.7492705385780724e-05	0.0026688016450271846	0.005243348680545737	0.9942295366678302
Ridge Regression with PCA	0.0005945705163835583	0.017751162611685575	0.024383816690246796	0.8752051747895626
Lasso Regression	0.0029877575268617627	0.0374890930641223	0.05466038352281991	0.3728974645366506
ElasticNet Regression	0.0033	0.05109835557138939	0.06902582379534494	0.3003
Decision Trees	5.0408947205889875e-06	0.00016894822536393775	0.002245193693334494 3	0.9989419630502595
Decision Trees with PCA	7.293080520500923e-05	0.0012739928590183088	0.008539534662086 56	0.9846925018358221
Random Forest	4.391058709938986e-06	0.0002066444634440724	0.002095485316087656 7	0.9990783575890566

Table 2. Results of applying ML models after further optimization of the dataset (rounded to the six number of figures)

Metric / 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

2.1. Best Performing Models

Based on Table 2, we can classify the models as top, good, moderate, and poor performers.

Top Performers:

- **Random Forest, Bagged Decision Trees, and XGBoost**: These models show the lowest errors (MSE, MAE, RMSE) and highest R2 scores, indicating excellent predictive accuracy. These models are robust and generalize well to unseen data.
    - **MSE**: 4e-06 (Random Forest, XGBoost)
    - **MAE**: 0.000201 (Random Forest)
    - **R2**: 0.99911 (Random Forest, XGBoost)

Good Performers:

- **Gradient Boost and Decision Trees**: These models also have low errors and high R2 scores, although not as high as the top performers.
    - **MSE**: 1.2e-05 (Gradient Boost)
    - **MAE**: 0.001656 (Gradient Boost)
    - **R2**: 0.997574 (Gradient Boost)

Moderate Performers:

- **Linear Regression, Ridge Regression, MLP, and CNN**: These models have good performance but are outperformed by ensemble methods.
    - **MSE**: 2.6e-05 (Linear Regression), 2.9e-05 (Ridge Regression)
    - **MAE**: 0.00266 (Linear Regression)
    - **R2**: 0.99459 (Linear Regression)

Lower Performers:

- **Linear Regression with PCA, ElasticNet, Lasso Regression, Bagged SVM, AdaBoost, and RNN**: These models show higher errors and lower R2 scores compared to others. PCA seems to reduce the performance for linear models.
    - **MSE**: 0.000592 (Linear Regression with PCA), 0.002982 (Lasso Regression)
    - **MAE**: 0.017638 (Linear Regression with PCA), 0.037452 (Lasso Regression)
    - **R2**: 0.876186 (Linear Regression with PCA), 0.376282 (Lasso Regression)

Therefore,

1. **Ensemble Methods** (Random Forest, Bagged Decision Trees, XGBoost) outperform other models significantly, with extremely low errors and high R2 values.
2. **Boosting Methods** (Gradient Boost, AdaBoost) also perform well but slightly less effectively than ensemble bagging methods.
3. **Simple Linear Models** (Linear Regression, Ridge Regression) are generally good but are less accurate than ensemble methods.
4. **Models using PCA** show reduced performance, likely due to information loss during dimensionality reduction.

5. **Lasso and ElasticNet** have higher errors, indicating they may not capture the complexity of the data as well as other models.

3. Interpreting results using interpretability tools

Interpreting the results using interpretability tools is a critical step to understand the underlying relationships and ensure the models are making reasonable predictions. The interpretability tools and techniques used in machine learning are:

1. **Feature Importance**
   Feature importance measures the contribution of each feature to the model's prediction. It helps to identify which features have the most significant impact on the target variable. It is used to rank features and identify the most important ones. Higher importance scores indicate more influential features.

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.

3. **LIME (Local Interpretable Model-agnostic Explanations)**
   LIME approximates the model locally with an interpretable model to explain individual predictions. Perturbs the data around the instance to be explained and fits an interpretable model (like linear regression) to approximate the black-box model locally. 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 the entire model).

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.

5. **Correlation Analysis**
   Correlation analysis examines the statistical relationship between features and the target variable. Correlation coefficients (like Pearson or Spearman) are calculated to measure the strength and direction of the relationship. Features with high correlation to the target are likely good predictors.

3.1. Applying the interpretability tools to the models

In this section, we present the results of applying interpretability tools on the top and good performing tools, including **Random Forest, Bagged Decision Trees, XGBoost, Gradient Boost and 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.

3.1.1. Feature Importance

Figures 1 to 5 show the importance of feature for the five models.

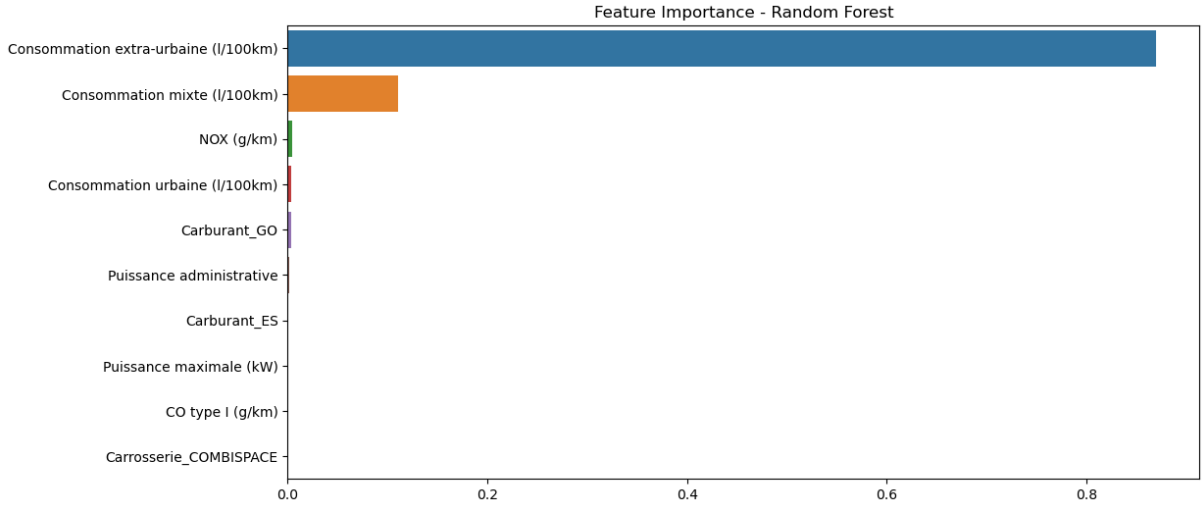

Figure 1. Feature importance in Random Forest

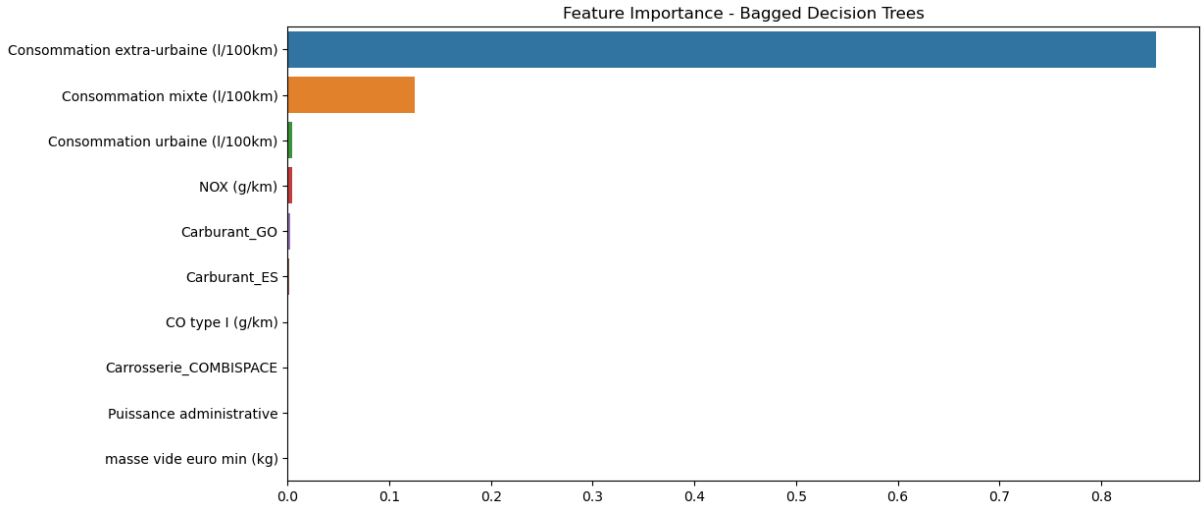

Figure 2. Feature importance in Bagged Decision Trees

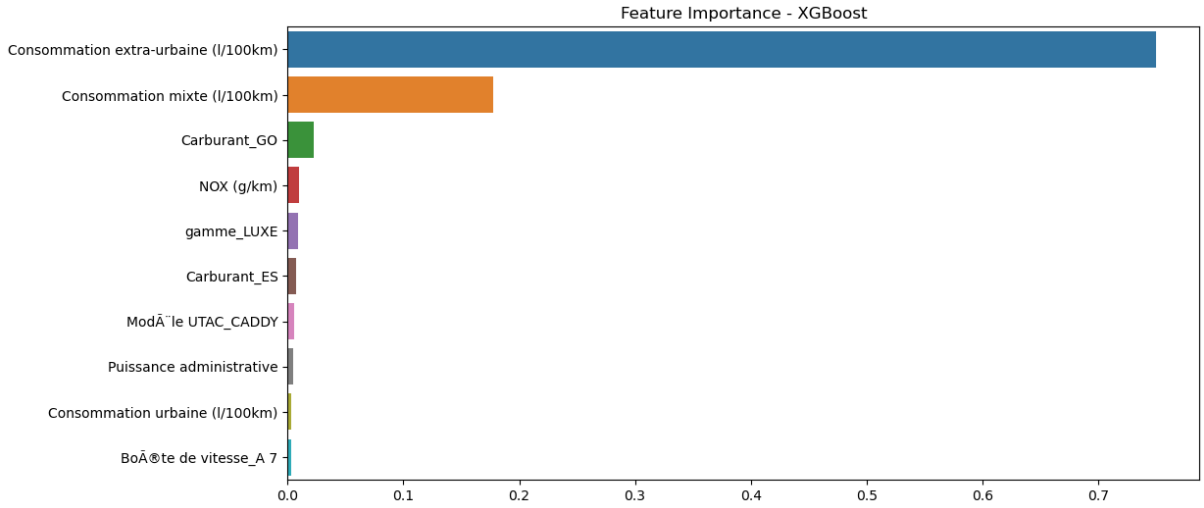

Figure 3. Feature importance in XGBoost

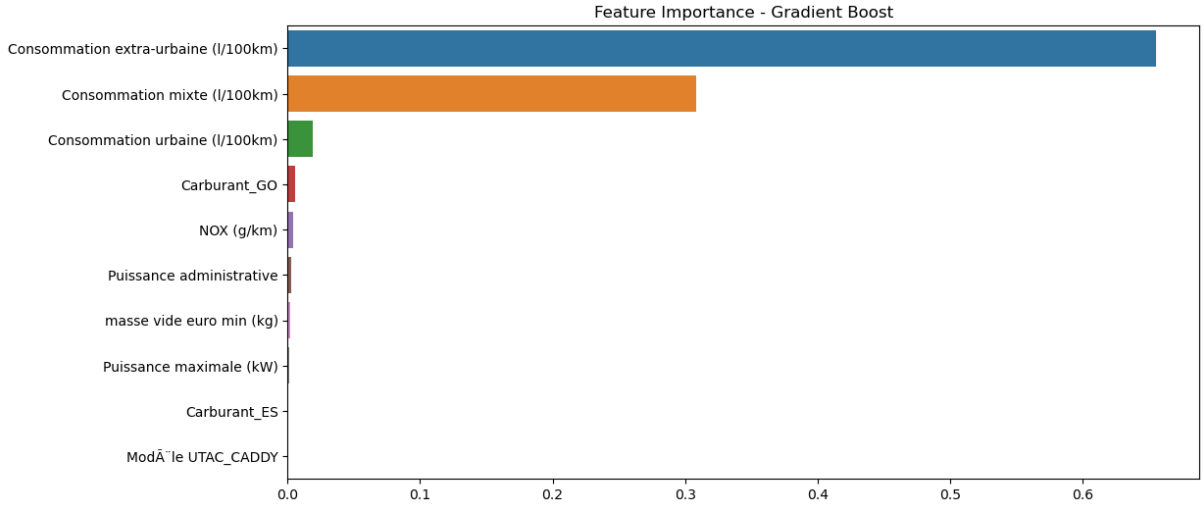

Figure 4. Feature importance in Gradient Boost

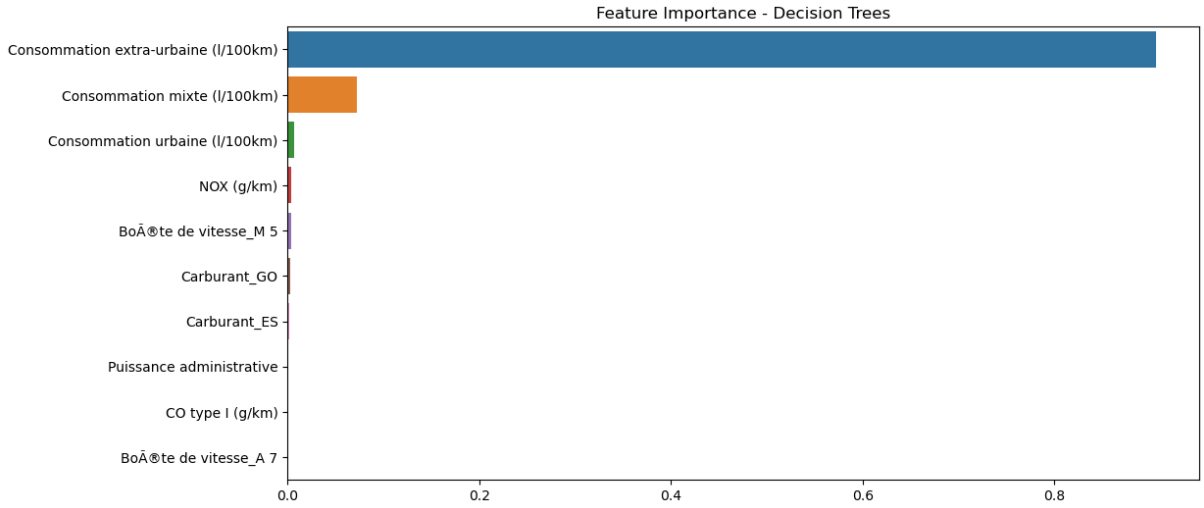

Figure 5. Feature importance in Decision Trees

Based on these figures:

1. **Consommation extra-urbaine (Extra-urban fuel consumption)** being the most important feature across all models indicates that the amount of fuel consumed during extra-urban driving conditions has a significant impact on CO2 emissions. This suggests that cars with higher extra-urban fuel consumption tend to emit more CO2.

2. **Consommation mixte (Mixed fuel consumption)** being the second most important feature reinforces the importance of overall fuel efficiency in determining CO2 emissions. Cars with higher mixed fuel consumption are likely to emit more CO2.

3. The variations in other important features across different models indicate the nuances in how each model learns and generalizes the relationships between features and CO2 emissions.

4. Features with very low importance values indicate that they have minimal impact on predicting CO2 emissions in cars according to these models.

Overall, this interpretation highlights the critical role of fuel consumption, especially in extra-urban and mixed driving conditions, in determining CO2 emissions.

3.1.2. SHAP

The results of applying this tool to the models are presented in Figures 6 to 10. As applying this tool to a large dataset like ours takes too much time, we decided to sample the data and then apply this method.

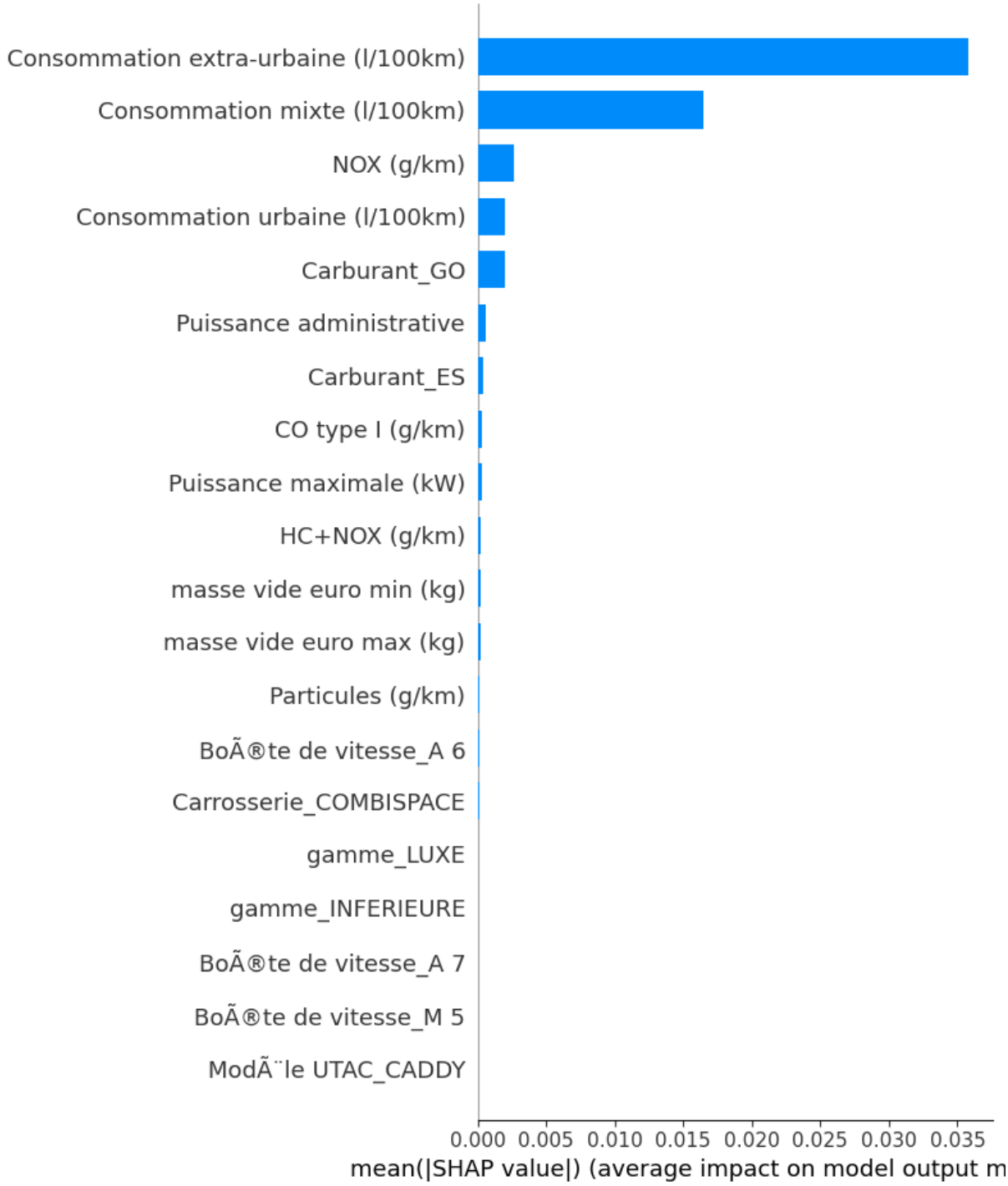

Figure 6. Results of SHAP on Random Forest

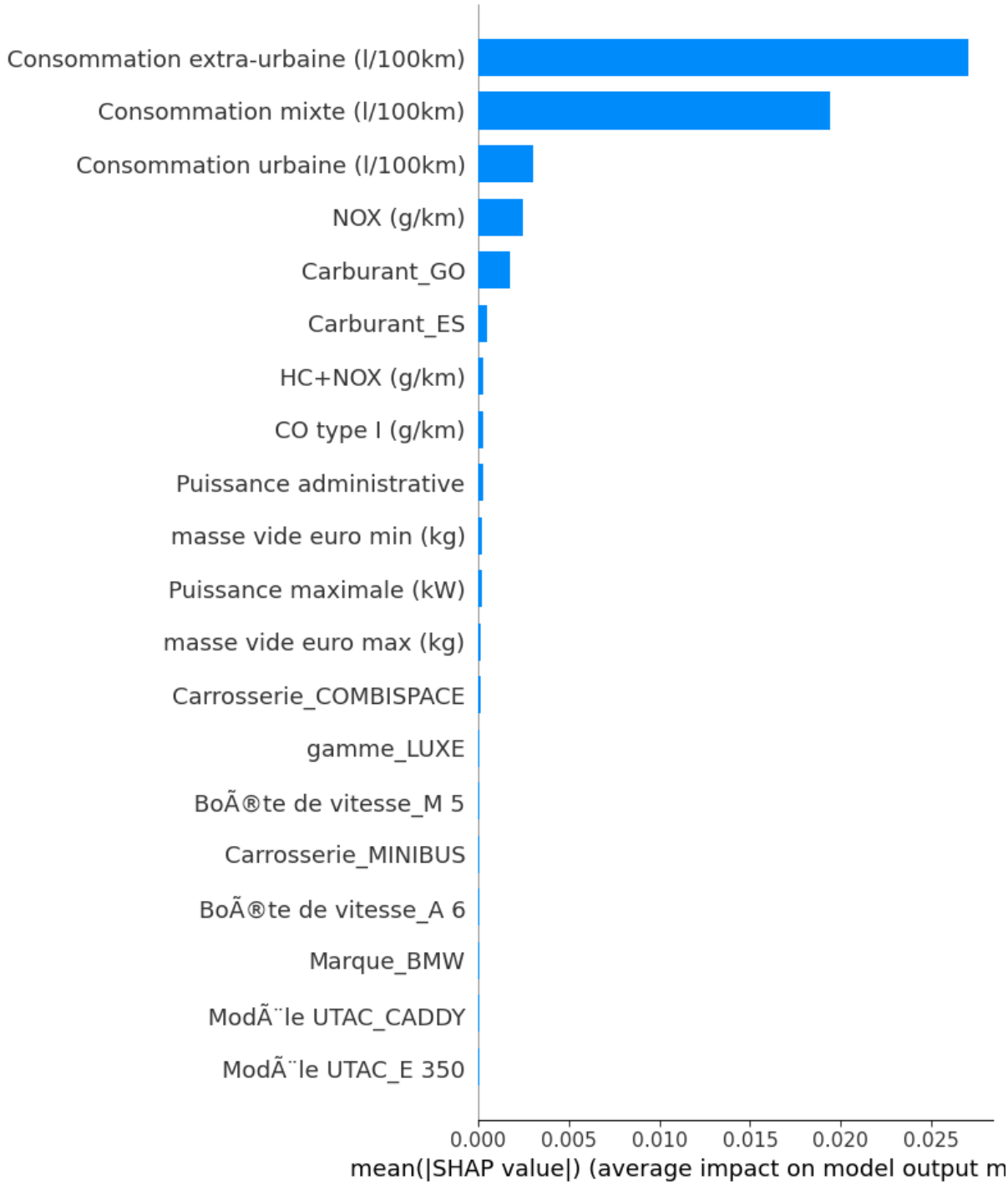

Figure 7. Results of SHAP on Bagged Decision Trees

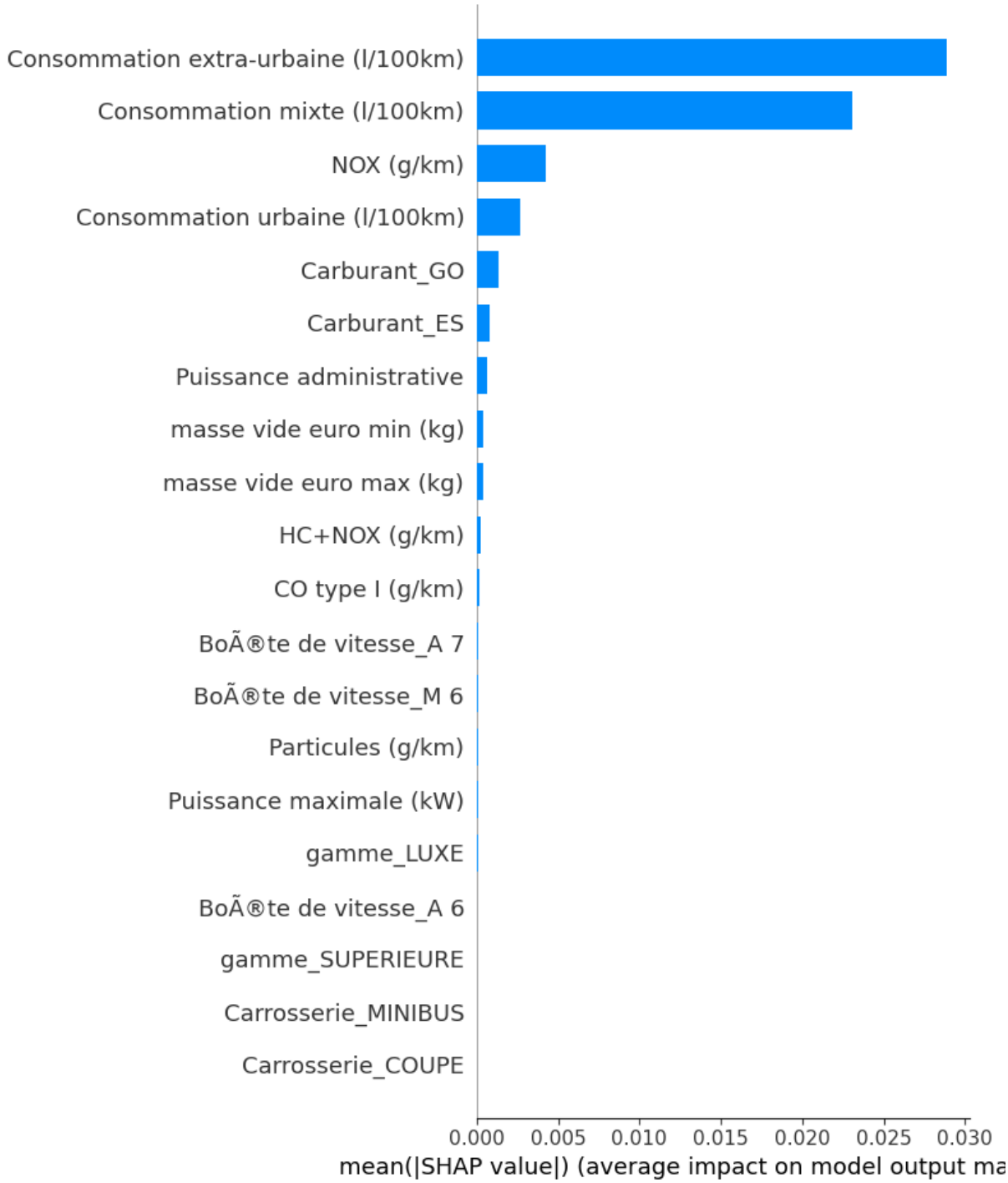

Figure 8. Results of SHAP on XGBoost

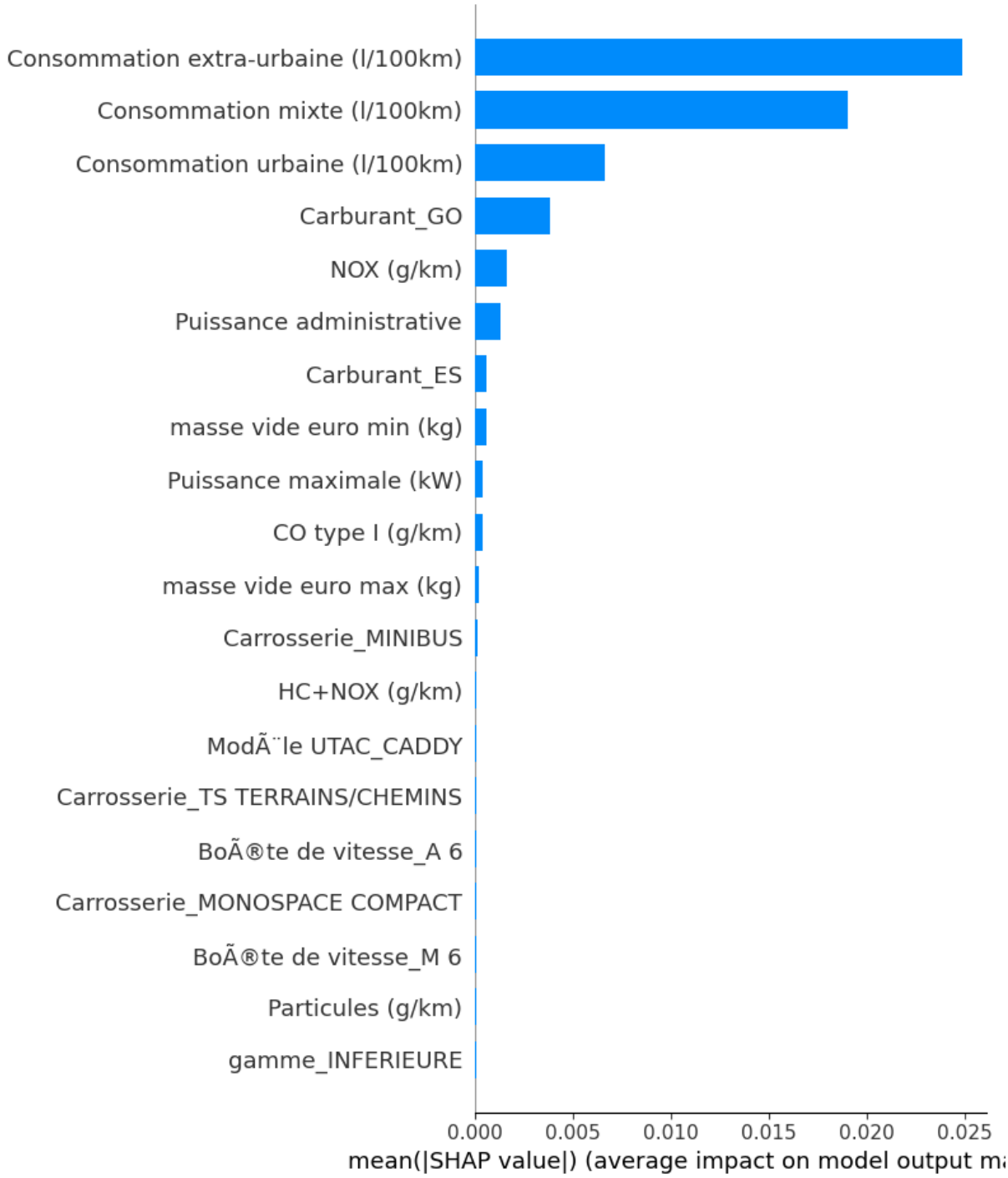

Figure 9. Results of SHAP on Gradient Boost

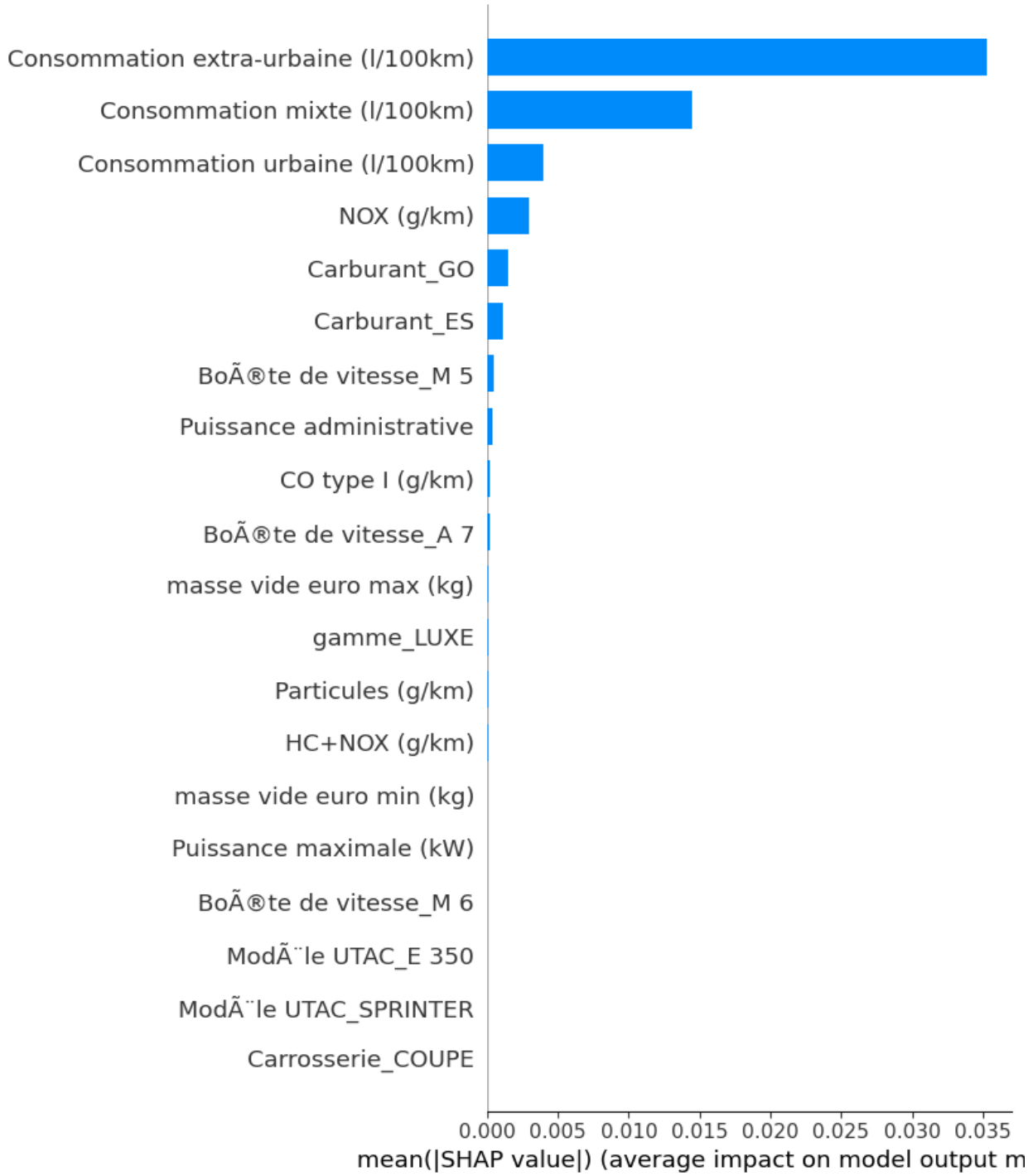

Figure 10. Results of SHAP on Decision Trees

Based on the SHAP value visualizations provided, we can derive the following insights:

1. **Consommation extra-urbaine (l/100km)** is consistently the most important feature across all models. This means that extra-urban fuel consumption significantly influences the model's predictions.
2. **Consommation mixte (l/100km)** follows as the second most important feature in all models, showing its strong impact on the output.
3. **Consommation urbaine (l/100km)**, **NOX (g/km)**, **Carburant_GO**, and **Carburant_ES** also appear frequently in the top features, although their importance is notably less than the top two features.

4. The importance values for other features like **Puissance administrative**, **CO type I (g/km)**, and **Puissance maximale (kW)** are relatively low, indicating lesser influence on the model's predictions.
5. The consistent importance of **Consommation extra-urbaine** and **Consommation mixte** suggests that fuel consumption, both in extra-urban and mixed driving conditions, is critical for predicting the target variable in this dataset.

Based on these results, we can conclude that

- **Fuel Consumption Dominance**: The high SHAP values for fuel consumption-related features indicate that the model predictions are highly sensitive to changes in fuel consumption. This could be due to fuel efficiency being a primary concern in the dataset's context (e.g., vehicle performance or environmental impact).
- **Pollutant Emissions**: Features like **NOX (g/km)** and **HC+NOX (g/km)**, although less influential than fuel consumption, are still significant, highlighting the role of emissions in the models' predictions.
- **Consistency Across Models**: The consistency in the importance of these features across various models (Random Forest, Bagged Decision Trees, XGBoost, Gradient Boosting, Decision Trees) reinforces their critical role and reliability in the predictive modeling context.

3.1.3. LIME

The results of applying this tool to the models are presented in Figures 11 to 15.

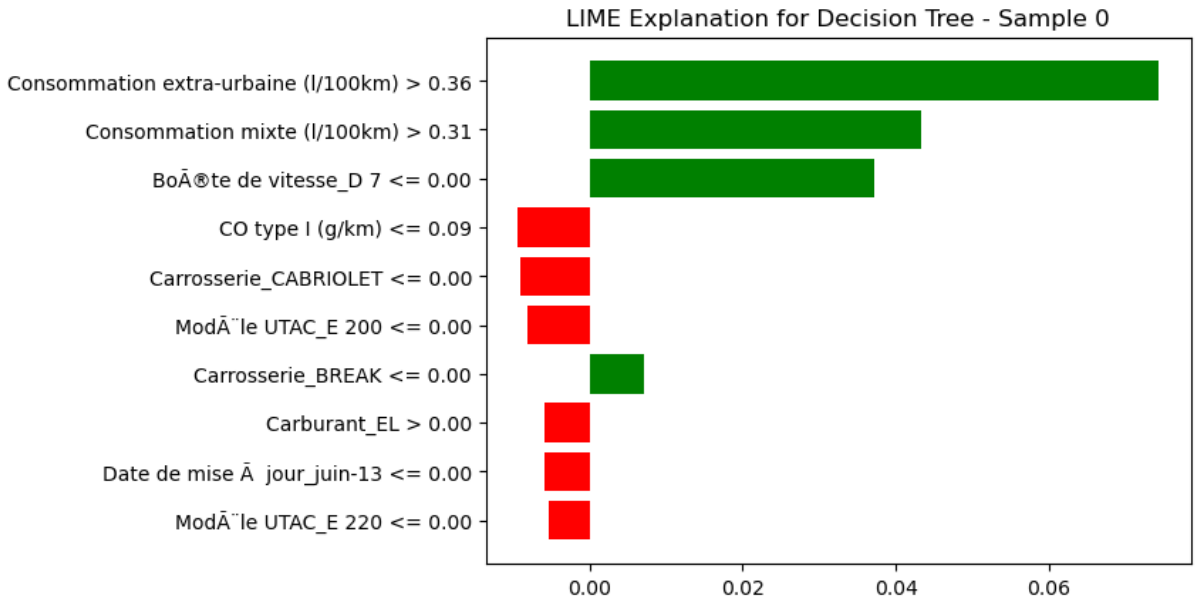

Figure 11. Results of LIME on Decision Trees

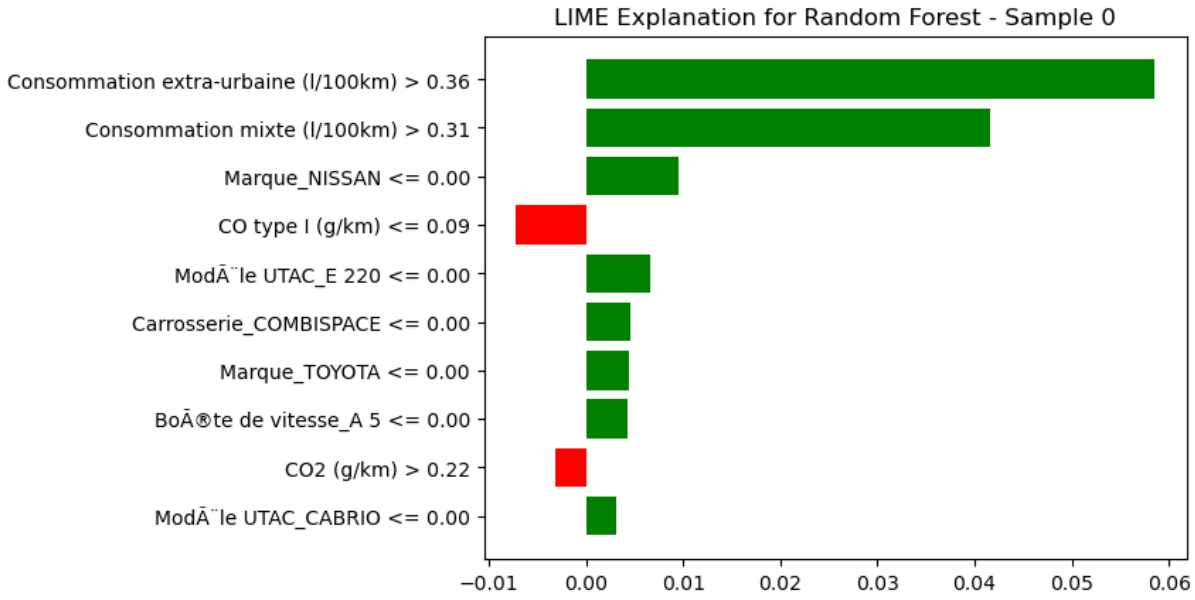

Figure 12. Results of LIME on Random Forest

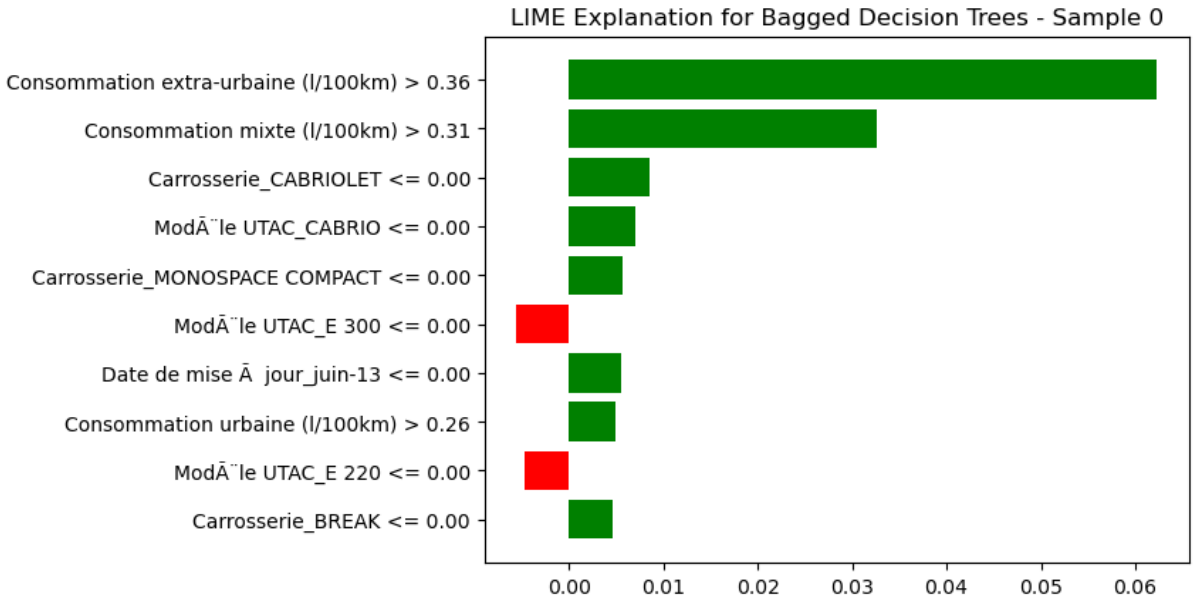

Figure 13. Results of LIME on Bagged Decision Trees

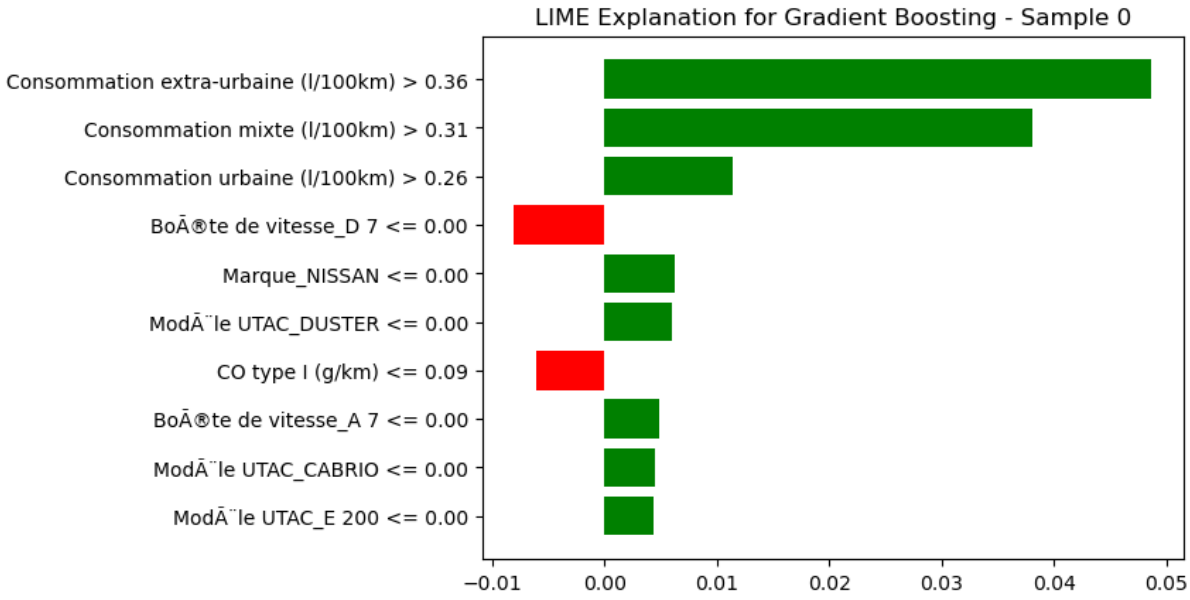

Figure 14. Results of LIME on Gradient Boosting

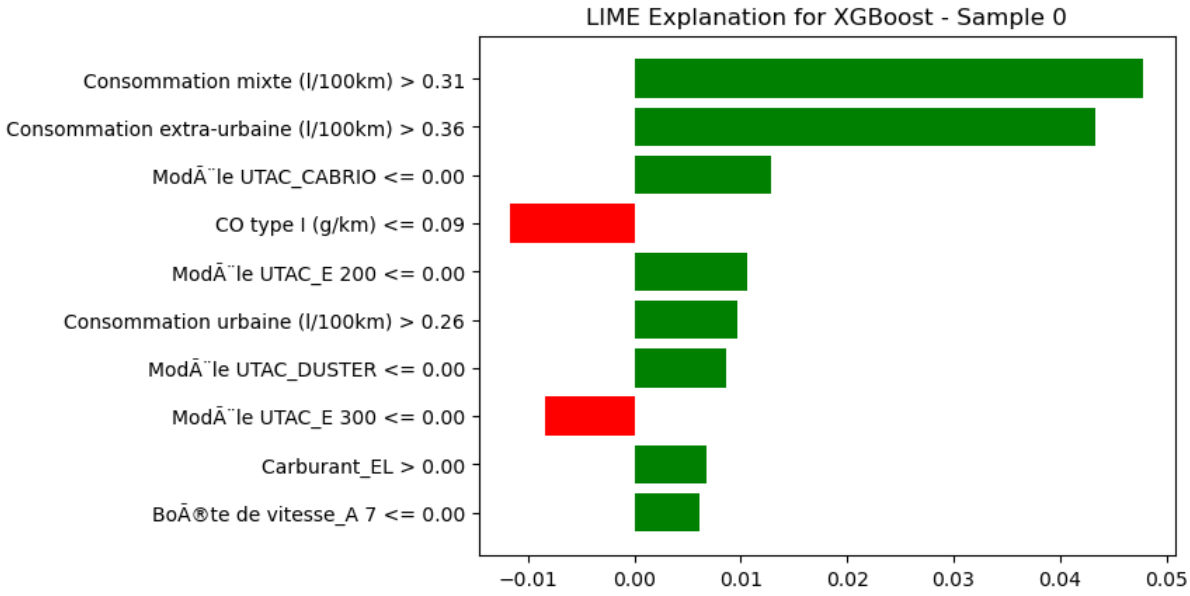

Figure 15. Results of LIME on XGBoost

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

1. **Consommation extra-urbaine (l/100km)**: This feature consistently shows a strong positive impact on the predictions.
2. **Consommation mixte (l/100km)**: Similarly, this feature also shows a consistently strong positive impact.
3. **CO type I (g/km)**: Generally has a negative impact.
4. **Marque_NISSAN** and **Marque_TOYOTA**: Show positive impacts in certain models.
5. **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 focusing on optimizing fuel consumption and understanding the impact of car types and brands can be crucial for improving model performance and making accurate predictions.

3.1.4. PDP

The results of applying this tool to the models are presented in Figures 16 to 20. As there are many features in our dataset, we selected two first features, Puissance administrative and Puissance maximale (kW)

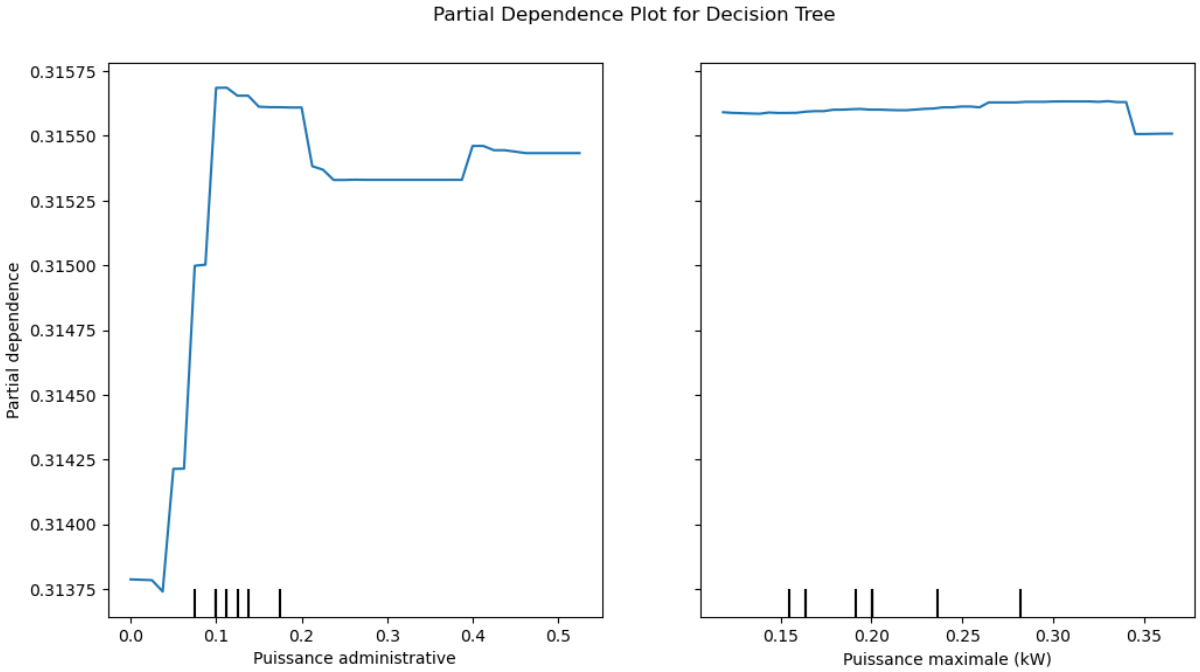

Figure 16. Results of PDP on Decision Trees

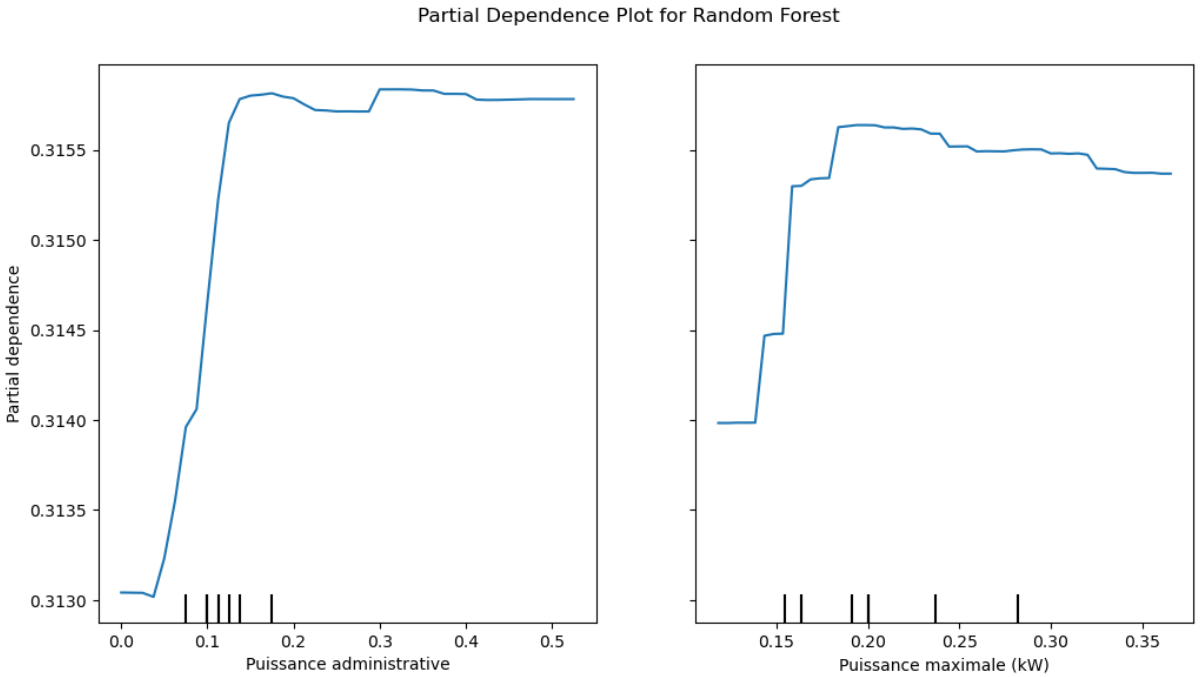

Figure 17. Results of PDP on Random Forest

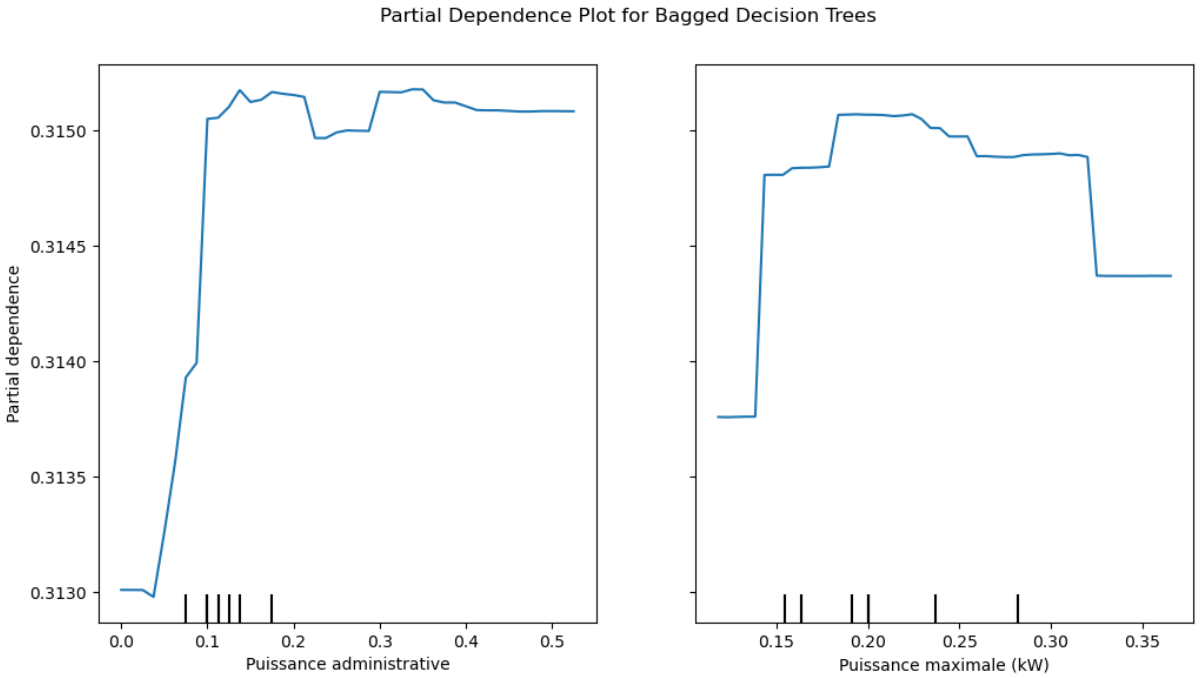

Figure 18. Results of PDP on Bagged Decision Trees

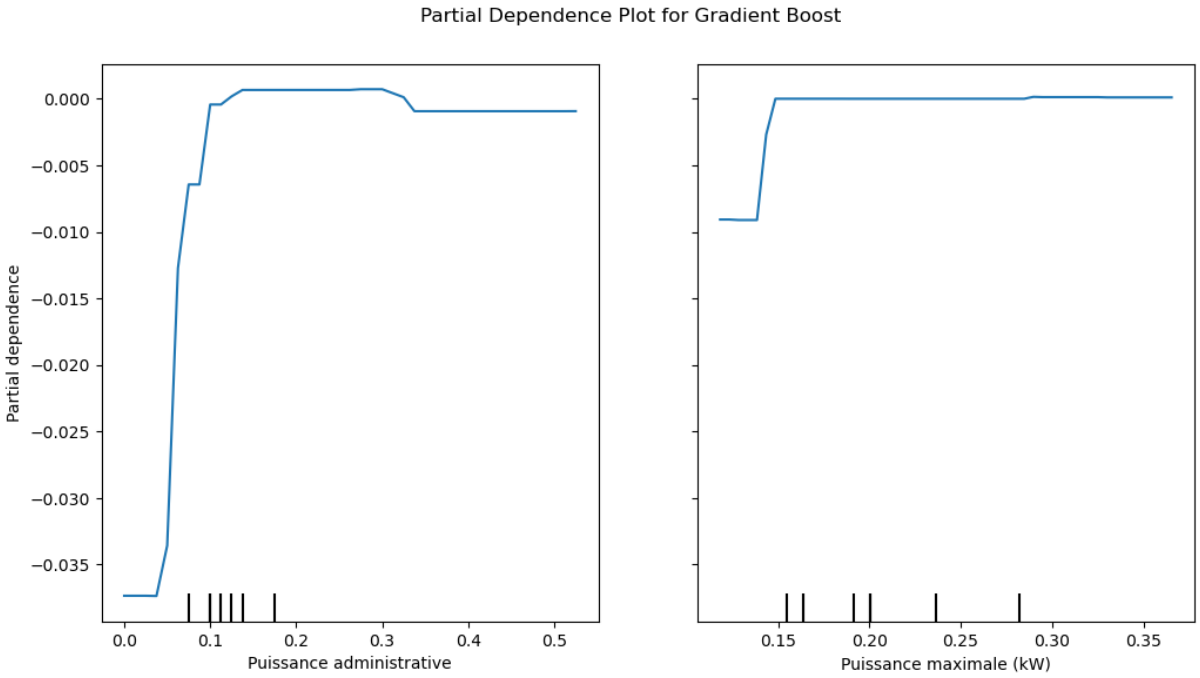

Figure 19. Results of PDP on Gradient Boost

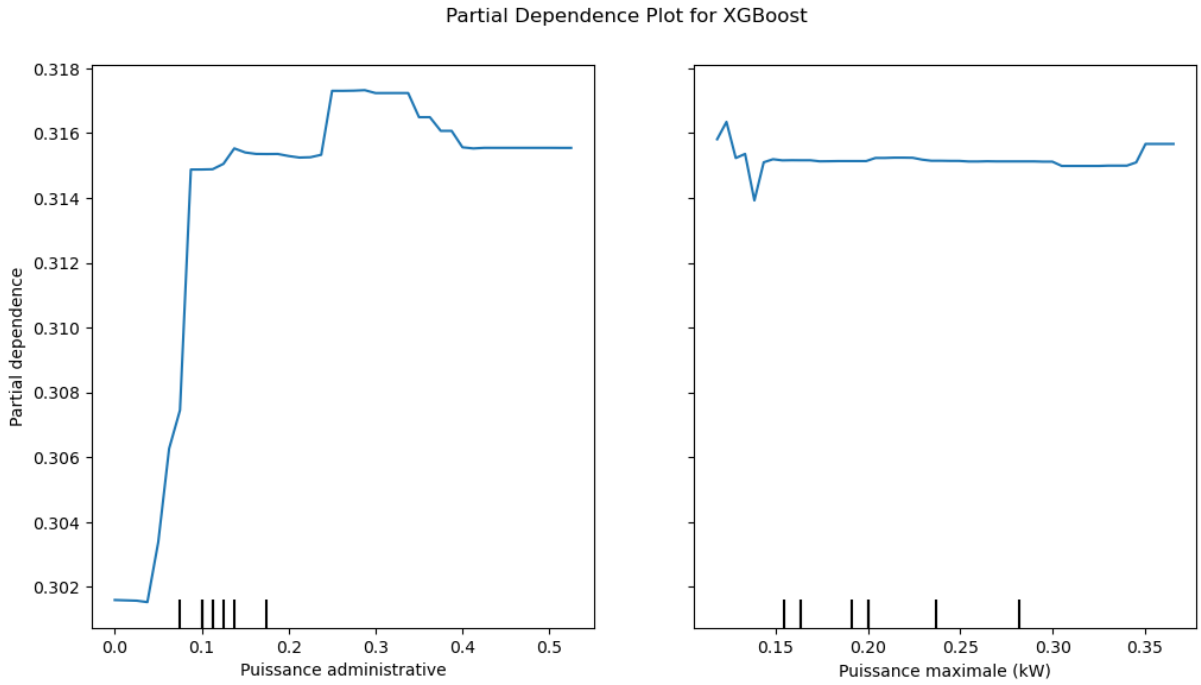

Figure 20. Results of PDP on XGBoost

As we have only investigated the impact of two features on the target feature, here are our interpretations based on the PDP diagrams and also the LIME diagrams:

- **Fuel Consumption**: Extra-urban and mixed fuel consumption are strong predictors of higher CO emissions, indicating that vehicles with higher fuel consumption tend to emit more CO.
- **Vehicle Characteristics**: Certain vehicle models and body styles consistently influence CO emissions. For example, models like UTAC_E 200 and UTAC_E 220, and body styles like Cabriolet and Break, are associated with different emission levels, suggesting that design choices impact emissions.
- **Transmission Types**: Specific transmission types, such as Boîte de vitesse_D 7, influence CO emissions, indicating the importance of transmission in vehicle emissions.
- **Car Brands**: The LIME explanations show that certain car brands, such as Nissan and Toyota, are associated with different emission levels, which could be due to differences in technology and design among brands.

3.1.5. Correlation Analysis

As we are dealing with a large number of attributes, the heatmap becomes cluttered and difficult to interpret. To improve the readability of the heatmap, we tried three approaches: clustering the heatmap by grouping similar features together, selecting a subset of features, and filtering by correlation threshold. The result of selecting a subset of features were more readable than other two approaches. Figure 21 shows the heatmap of top 15 features.

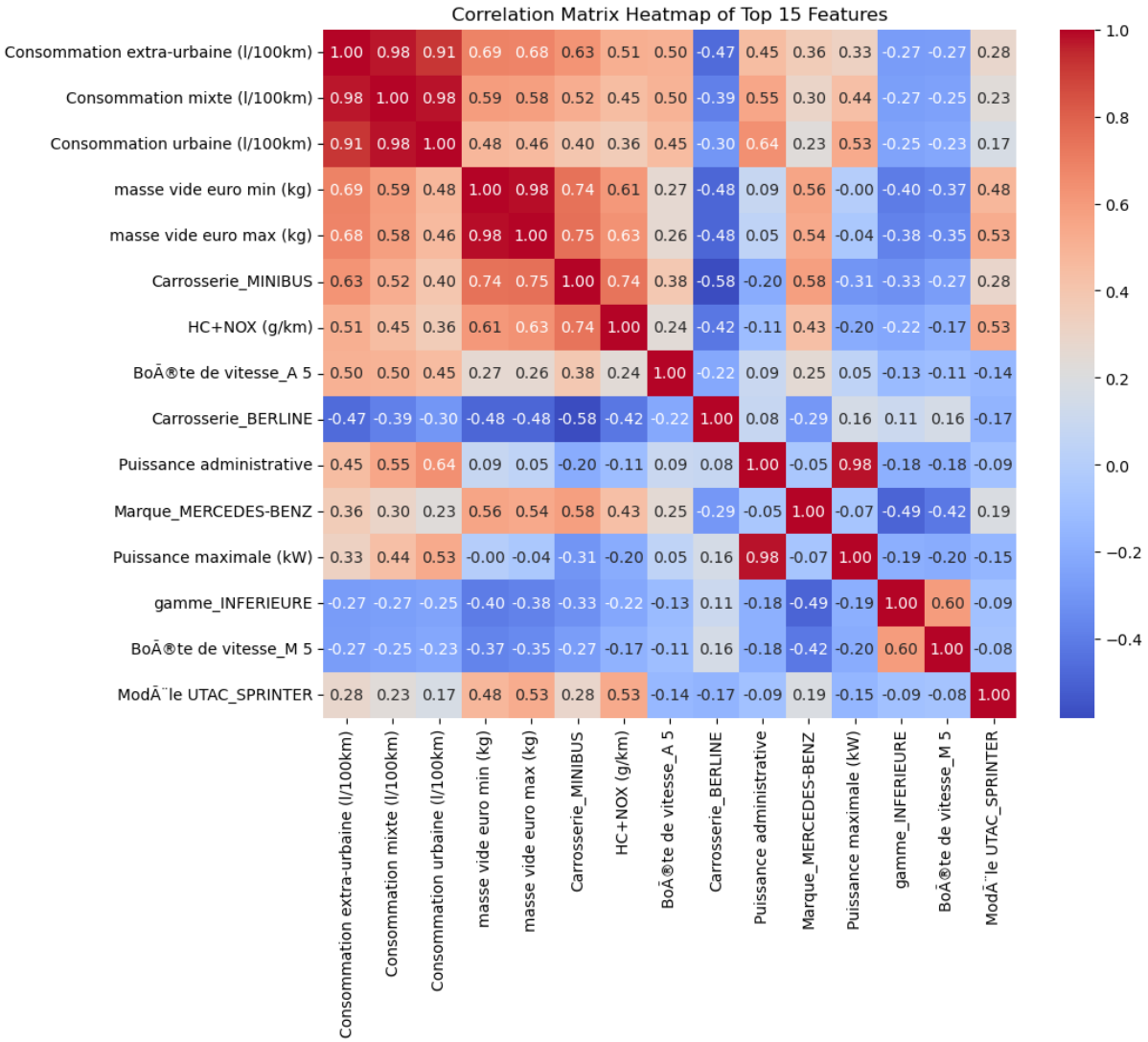

Figure 21. Correlation matrix heatmap of top 15 features

By analyzing the heatmap, we can notice that:

Strong Positive Correlations:

1. **Consommation extra-urbaine (l/100km), Consommation mixte (l/100km), and Consommation urbaine (l/100km)**:
    - These three fuel consumption metrics are highly positively correlated with each other, with correlation coefficients around 0.98-1.00. This indicates that as one type of fuel consumption increases, the others tend to increase as well, which makes intuitive sense.
2. **Masse vide euro min (kg) and Masse vide euro max (kg)**:
    - These two weight measurements are also highly positively correlated (0.91), suggesting that vehicles with a higher minimum empty weight also tend to have a higher maximum empty weight.

Moderate Positive Correlations:

3. **Consommation extra-urbaine (l/100km) and Masse vide euro min (kg)**:
    - There is a moderate positive correlation (0.69) indicating that heavier vehicles tend to have higher extra-urban fuel consumption.

4. **Consommation mixte (l/100km) and Masse vide euro min (kg)**:
    - Similar to extra-urban fuel consumption, mixed fuel consumption also shows a moderate positive correlation (0.68) with vehicle weight.

Notable Correlations with Categorical Variables:

5. **Carrosserie_MINIBUS and HC+NOX (g/km)**:
    - There is a positive correlation (0.75) between the minibus body type and HC+NOX emissions, indicating that minibuses might have higher emissions.
6. **Marque_MERCEDES-BENZ and Puissance maximale (kW)**:
    - Mercedes-Benz vehicles have a moderate positive correlation (0.55) with maximum power, suggesting that Mercedes-Benz models tend to have higher engine power.

Negative Correlations:

7. **Carrosserie_BERLINE and various consumption metrics**:
    - The Berline (sedan) body type shows negative correlations with fuel consumption metrics, indicating that sedans tend to have lower fuel consumption compared to other body types.
8. **gamme_INFERIEURE and Puissance maximale (kW)**:
    - There is a moderate negative correlation (-0.45) between the inferior range (likely representing lower-end models) and maximum power, indicating that lower-end models tend to have less powerful engines.

Insights on Specific Variables:

9. **Puissance administrative and Puissance maximale (kW)**:
    - These two power metrics have a positive correlation (0.54), showing that administrative power is a good indicator of maximum power.
10. **Boîte de vitesse_A 5 and other variables**:
    - The presence of a 5-speed automatic gearbox (Boîte de vitesse_A 5) has moderate correlations with several variables, indicating its widespread use across various vehicle types.

4. Interpreting results using interpretability tools

Analyzing the evaluations results shown in Table 2 indicate that five models are doing as top and good models, including **Random Forest, Bagged Decision Trees, XGBoost, Gradient Boost and Decision Trees.** The ensemble methods (Random Forest, Bagging, Gradient Boosting, XGBoost) provided better performance and robustness compared to the single Decision Tree model. While single models like Decision Tree are more interpretable, ensemble methods offer superior predictive performance at the cost of interpretability. Therefore,

- For Quick Interpretability: It is recommended to use Decision Trees for scenarios where model interpretability is crucial and the dataset is small to medium-sized.
- For Balanced Performance: Random Forest is a good choice for balancing interpretability, accuracy, and computational efficiency.

- For High Accuracy: Gradient Boosting and XGBoost should be preferred for applications requiring high accuracy and where computational resources are available. However, as these models require more computational resources, businesses should ensure they have the necessary infrastructure to support these models for large-scale applications.
- For Reducing Variance: Bagging methods, including Bagging Regressor and Random Forest, are recommended to reduce model variance and improve robustness.

By applying interpretability tools on these five models, we found the following insights:

1. Insights and implications based on the modeling success

- **High Importance of Fuel Consumption**:
    - **Scientific Conclusion**: The models consistently identified **Consommation extra-urbaine (l/100km)** and **Consommation mixte (l/100km)** as the most significant predictors. This implies that fuel consumption, both in extra-urban and mixed conditions, is a critical factor influencing the target variable.
    - **Business Implication**: For businesses in the automotive industry, optimizing fuel efficiency in both extra-urban and mixed driving conditions can lead to significant improvements in performance metrics. Focusing on fuel-efficient technologies and promoting them can enhance market competitiveness and meet regulatory standards.
- **Moderate Influence of Emission Factors**:
    - **Scientific Conclusion**: Features related to emissions, such as **NOX (g/km)** and **HC+NOX (g/km)**, though less influential than fuel consumption, still play a significant role in the models.
    - **Business Implication**: Businesses should continue to invest in technologies that reduce emissions. This is not only important for regulatory compliance but also for enhancing the environmental profile of the vehicles, which can be a strong selling point in an increasingly eco-conscious market.
- **Consistent Model Performance**:
    - **Scientific Conclusion**: The consistency in the importance of the top features across different models (Decision Trees, Random Forest, Gradient Boosting, XGBoost) indicates the robustness and reliability of these features in predicting the target variable.
    - **Business Implication**: The reliability of these features across different modeling techniques suggests that investments in understanding and improving these aspects (fuel consumption and emissions) are likely to yield consistent and positive results, irrespective of the modeling approach used.

2. Modeling limitations and areas for improvement

- **Low Importance of Other Features**:
    - **Scientific Conclusion**: Many features exhibited low importance, indicating they have minimal impact on the model predictions.
    - **Business Implication**: It may be beneficial to re-evaluate the data collection process and the relevance of these features. Investing resources in features with low impact might not be cost-effective. Instead, we can focus on enhancing the quality and granularity of high-impact features.

- **Potential Overfitting Risks**:
    - **Scientific Conclusion**: Given the high importance of a few features, there is a risk that models might overfit these aspects, potentially neglecting other relevant factors.
    - **Business Implication**: Implementing regularization techniques and conducting thorough cross-validation can help mitigate overfitting risks. Additionally, exploring new data sources or feature engineering techniques might uncover other influential factors, leading to more balanced and generalizable models.

3. Strategic Recommendations

- **Enhanced Fuel Efficiency Programs**:
    - Given the dominant importance of fuel consumption, automotive companies should prioritize research and development in fuel efficiency technologies. This can include advanced engine designs, lightweight materials, and hybrid or electric vehicle options.
- **Emission Reduction Initiatives**:
    - Continued efforts in reducing vehicle emissions are crucial. Investing in cleaner technologies and ensuring compliance with stringent emission regulations can enhance the brand's reputation and market appeal.
- **Data-Driven Decision Making**:
    - The insights from the modeling emphasize the need for a data-driven approach in decision-making processes. Companies should leverage data analytics to continually monitor and improve key performance metrics related to fuel consumption and emissions.
- **Customer Education and Marketing**:
    - Highlighting the advancements in fuel efficiency and low emissions in marketing campaigns can attract environmentally conscious consumers. Educating customers about the benefits of these features can also help in differentiating the brand in a competitive market.