Rendu 2 - Rapport de modélisation

1. Classification du problème

1.1. À quel type de problème de machine learning votre projet s'apparente-t-il ?

Notre projet s'apparente à **deux types complémentaires de problèmes de machine learning supervisé** :

- **Un problème de régression supervisée**, dont l'objectif est de prédire la valeur continue des émissions de CO_2 WLTP (en g/km).
- **Un problème de classification supervisée multiclasse ordonnée**, dans lequel cette valeur continue est discrétisée en **8 classes de niveaux d'émissions (0 à 7)**, conformément à la grille réglementaire européenne de l'étiquette énergie (directive 1999/94/EC), enrichie d'une classe spécifique pour les véhicules zéro émission.

Cette double approche permet à la fois :

- D'évaluer la **capacité prédictive maximale** des modèles (régression), cette approche est pertinente d'un point de vue technique, mais elle suppose que l'utilisateur final soit capable d'interpréter ces valeurs numériques.
- Les constructeurs et utilisateurs finaux raisonnant par **classes d'émissions** (A, B, C, D, etc.), nous proposons une **sortie interprétable et directement exploitable métier** (classification).

1.2. À quelle tâche de machine learning votre projet s'apparente-t-il ? (détection de fraude, reconnaissance faciale, analyse de sentiment, etc) ?

Notre projet s'apparente à de l'apprentissage supervisé, il consiste à :

- Construire et entraîner un modèle afin de prédire **le niveau d'émissions de CO_2 d'un véhicule** à partir de ses caractéristiques techniques (motorisation, cylindrée, puissance, masse, dimensions, type de carburant, technologies embarquées),
- Puis à **classer automatiquement les véhicules selon leur impact carbone**, en catégories compréhensibles et normées.

Cette problématique correspond directement :

- Aux besoins d'**homologation WLTP**,
- À l'**évaluation réglementaire et fiscale** des véhicules,
- À l'**aide à la conception** pour les constructeurs, flottes, collectivités ou régulateurs,
- Ainsi qu'à des cas d'usage d'**automatisation de la labellisation environnementale**.

1.3. Quelle est la métrique de performance principale utilisée pour comparer vos modèles ? Pourquoi celle-ci ?

- Pour la **régression**, la métrique principale retenue est le **RMSE (Root Mean Squared Error)**, car :
  - Elle pénalise fortement les erreurs importantes,
  - Elle est directement interprétable en **g/km de CO_2**,
  - Elle est adaptée aux enjeux réglementaires où les erreurs élevées sont critiques.
- Pour la **classification**, la métrique principale est le **F1-score pondéré**, car :

- Le dataset présente un déséquilibre entre classes, le score F1 pondéré permet de refléter ce déséquilibre,
- Le F1-score pondéré permet de concilier précision et rappel

1.4. Avez-vous utilisé d'autres métriques de performances qualitative ou quantitative ? Si oui, détaillez-le.

- Pour la **régression**
  - **MAE** : pour mesurer l'erreur moyenne absolue.
  - **MSE** : pour mesurer l'erreur quadratique moyenne
  - **R^2** : pour évaluer les performances des modèles et mesurer à quel point ils s'ajustent aux données (éviter le surapprentissage)
- Pour la **classification**
  - **Accuracy** : pour évaluer la performance globale des modèles
  - **Matrice de confusion** : pour distinguer tous les cas de bonnes ou mauvaises classifications pour chaque classe.
  - **Classification report :** pour obtenir une évaluation complète de la performance

2. Choix du modèle et optimisation

2.1. Quels algorithmes avez-vous essayés ?

Nous avons testé une large gamme de modèles, allant des baselines aux modèles avancés

Régression

- Régression linéaire, Ridge, Lasso, ElasticNet
- RandomForestRegressor
- KNeighborsRegressor
- XGBoostRegressor

Modèle	R^2 Cross Validation	R^2 Train	R^2 Test	MAE	MSE	RMSE
Random Forest	0.9948849	0.998010	0.993819	2.436274	29.304231	5.413338
KNN	0.9898879	0.994041	0.989878	3.294370	47.989274	6.927429
RidgeCV	0.9344312	0.934468	0.935798	12.855016	304.389997	17.446776
Ridge	0.9344312	0.934468	0.935798	12.855034	304.390133	17.446780
Linear Regression	-2.001800e+21	0.934468	0.935797	12.855326	304.395216	17.446926
Lasso	0.9266005	0.926648	0.927446	13.687712	343.986807	18.546881
Elastic Net	0.8777564	0.877803	0.877820	17.637387	579.268673	24.068001

Classification

- Logistic Regression (baseline)
- KNeighborsClassifier
- Gradient Boosting
- DecisionTreeClassifier

- RandomForestClassifier
- Bagging de Decision Trees
- XGBoost Classifier

Modèle	Accuracy_train	Accuracy_test	F1 Score
XGB	0.931047	0.923282	0.930407
BaggingClassifier	0.948904	0.921747	0.928753
Random Forest Classifier	0.949954	0.921333	0.928805
Arbres de décision	0.949956	0.918734	0.925257
KNN	0.925815	0.896947	0.905331
Logistic Regression	0.762282	0.762897	0.769626
Ada Boost	0.648459	0.651021	0.658861

Résultats des modèles testés avec un LazyClassifier :

Model	Accuracy	Balanced Accuracy	F1 Score	Time Taken
XGBClassifier	0,92	0,93	0,92	8,72
BaggingClassifier	0,92	0,93	0,92	17,21
RandomForestClassifier	0,92	0,93	0,92	40,28
DecisionTreeClassifier	0,92	0,93	0,92	3,19
ExtraTreesClassifier	0,92	0,93	0,92	48,52
LGBMClassifier	0,92	0,93	0,92	6,34
ExtraTreeClassifier	0,91	0,92	0,91	1,11
KNeighborsClassifier	0,90	0,90	0,90	40,20
LogisticRegression	0,76	0,75	0,76	8,40
LinearDiscriminantAnalysis	0,70	0,68	0,69	1,92
LinearSVC	0,67	0,65	0,65	83,30
SGDClassifier	0,64	0,65	0,60	11,48
NearestCentroid	0,60	0,64	0,60	0,93
Perceptron	0,55	0,55	0,54	5,29
BernoulliNB	0,55	0,53	0,51	0,82
RidgeClassifierCV	0,59	0,52	0,55	2,09
RidgeClassifier	0,59	0,52	0,55	0,94
PassiveAggressiveClassifier	0,51	0,52	0,50	5,25
AdaBoostClassifier	0,52	0,43	0,47	15,28
QuadraticDiscriminantAnalysis	0,27	0,41	0,28	1,78
GaussianNB	0,28	0,40	0,28	1,12
DummyClassifier	0,24	0,12	0,09	0,62

2.2. Décrivez celui / ceux que vous avez retenu et pourquoi ?

- En **régression**, le **Random Forest Regressor** obtient les meilleures performances ($R^2 \approx 0.994$, RMSE ≈ 5.41 g/km), confirmant la forte capacité prédictive des modèles non linéaires sur ce problème.

- En **classification**, le **XGBoost Classifier** constitue le meilleur compromis :
  - F1-score pondéré test ≈ 0. 930407
  - Excellente généralisation (écart validation / test très faible)
  - Forte robustesse
  - Explicabilité avancée via SHAP

Le modèle de Bagging optimisé obtient des performances très proches, mais le XGBoost a été privilégié pour :

- Sa capacité de modélisation fine,
- Mieux adapté dans le cas des classes déséquilibrés
- Possibilité de tunning très fin des hyperparamètres
- Ses outils d'explicabilité plus riches,
- Sa meilleure adaptabilité en contexte industriel.

2.3. Avez-vous utilisé des techniques d'optimisation de paramètres de type Grid Search et Validation Croisée ?

- Utilisation de **validation croisée**
- Optimisation via **Optuna** pour :
  - Bagging (nombre d'arbres, sous-échantillonnage, profondeur implicite),
  - XGBoost (n_estimators, learning_rate, max_depth, subsample, colsample_bytree, gamma, min_child_weight, reg_alpha, reg_lambda).
- Tests avec **early stopping** via l'API native XGBoost.

3. Interprétation des résultats

3.1. Avez-vous analysé les erreurs de votre modèle ?

L'analyse des matrices de confusion montre que :

- Les erreurs se concentrent quasi exclusivement entre **classes adjacentes** (ex : 2 ↔ 3, 4 ↔ 5).
- Les classes extrêmes (véhicules zéro émission ou très fortement émetteurs) sont **quasi parfaitement prédites**.

Ces erreurs correspondent à des véhicules proches des **seuils réglementaires**, ce qui est cohérent d'un point de vue physique et métier. Des tests d'ajustement de seuils de décision ont été menés, mais n'ont apporté qu'un gain marginal, confirmant que le modèle est déjà proche de son optimum.

3.2. Cela a-t-il contribué à son amélioration ? Si oui, décrivez.

Oui, mais l'impact chiffré est minime. L'analyse des erreurs a permis de confirmer que les rares biais concernaient les classes adjacentes (2↔3, 4↔5). Des ajustements de seuils ciblés ont été testés : ils réduisent légèrement la sur-prédiction de la classe 3 (≈0,2–0,3 pp) mais ne changent quasiment pas le F1 (gain ~1e-4).

En pratique, cela valide surtout que le modèle est déjà proche de son optimum et qu'un tuning supplémentaire doit plutôt passer par des hyperparamètres que par un réglage de seuils.

3.3. Avez-vous utilisé des techniques d'interprétabilité de type SHAP, LIME, Skater… (Grad-CAM pour le Deep Learning..)

Nous avons utilisé principalement SHAP (SHapley Additive exPlanations), particulièrement adapté aux modèles de type boosting et aux problématiques de classification multiclasses :

- importances globales (gain, permutation, SHAP),
- SHAP summary plots,
- analyses SHAP par classe (approche one-vs-rest),
- Partial Dependence Plots,
- Explications locales (waterfall plots).

Dans un contexte de classification multi-classes ordonnée (8 classes CO_2), ces analyses ont permis d'identifier à la fois les déterminants globaux du modèle et les variables discriminantes spécifiques à chaque classe.

Les décisions du modèle sont principalement pilotées par la cylindrée, la puissance, la masse, les dimensions du véhicule et le type de carburant. Les effets observés sont monotones et cohérents avec les lois physiques.

Les analyses montrent que les erreurs se concentrent quasi exclusivement entre classes adjacentes, ce qui est cohérent avec la nature continue des émissions de CO_2 et la discrétisation réglementaire en classes.

L'explicabilité globale et locale confirme que le modèle apprend des relations structurelles robustes et fournit des décisions auditables et interprétables, compatibles avec un usage métier et réglementaire.

3.4. Qu'est ce qui a (ou non) engendré une amélioration significative de vos performances ?

Les gains de performance significatifs proviennent principalement du tuning des hyperparamètres :

- Optimisation XGBoost (Optuna) :
    - Réglage du learning rate, de la profondeur, du subsampling, de la régularisation et des paramètres de croissance des arbres, permettant d'atteindre un F1-score pondéré test d'environ 0,93047 contre ~0,919 en configuration par défaut.
- Optimisation du Bagging (Optuna) :
    - Gains plus modérés (~0,9276 vs ~0,9264), confirmant la robustesse mais la moindre flexibilité de cette approche.
- Les modèles baselines (Logistic Regression, KNN, Gradient Boosting) n'apportent pas d'amélioration significative face aux ensembles de type Random Forest, Bagging ou XGBoost.
- L'early stopping et le sous-échantillonnage pour SHAP ont amélioré le temps de calcul et la stabilité des analyses, sans impact sur les performances finales.

Conclusion :

Le modèle XGBoost « tuné » atteint ~0,93047 de F1 pondéré, avec des erreurs essentiellement sur des cas aux frontières entre classes (2↔3, 4↔5), ce qui est cohérent avec la réalité métier : les véhicules proches des seuils réglementaires sont les plus difficiles à arbitrer. Les classes extrêmes (zéro émission ou très émetteurs) sont quasiment parfaites, ce qui sécurise le pilotage des offres sur ces segments.

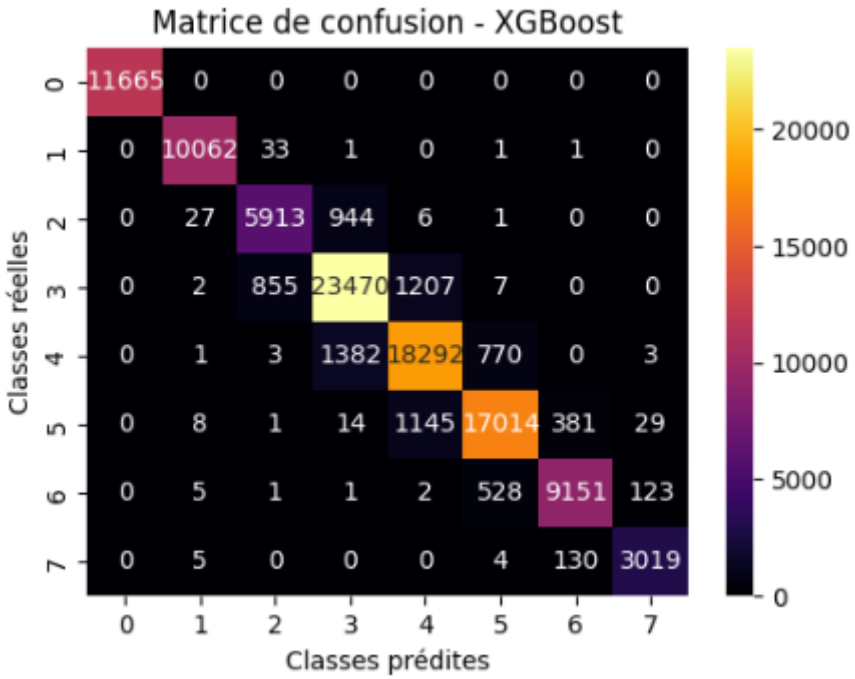
Les variables moteur/puissance/masse/dimensions et type de carburant pilotent les décisions, et l'explicabilité (SHAP) confirme des effets monotones attendus : plus de puissance/cylindrée/masse → classes supérieures. Aucun comportement aberrant détecté.

D'un point de vue industriel, le modèle est prêt pour un déploiement pilote : il offre une bonne granularité pour segmenter l'offre, anticiper les impacts réglementaires et guider les arbitrages marketing/produit.

Les points d'attention portent sur le suivi en production (dérive des distributions, glissements sur les classes adjacentes) et la capacité à rafraîchir le modèle avec de nouvelles motorisations et évolutions réglementaires.

Annexe – Schémas d'interprétabilité du XGBoost :

Classe	Precision	Recall	F1_score	Support
0	1.00	1.00	1.00	11665
1	1.00	1.00	1.00	10098
2	0.87	0.86	0.86	6891
3	0.91	0.92	0.91	25541
4	0.89	0.89	0.89	20451
5	0.93	0.92	0.92	18592
6	0.95	0.93	0.94	9811
7	0.95	0.96	0.95	3158
Accuracy			0.93	106207
Macro_avg	0.94	0.93	0.93	106207
Weighted_avg	0.93	0.93	0.93	106207

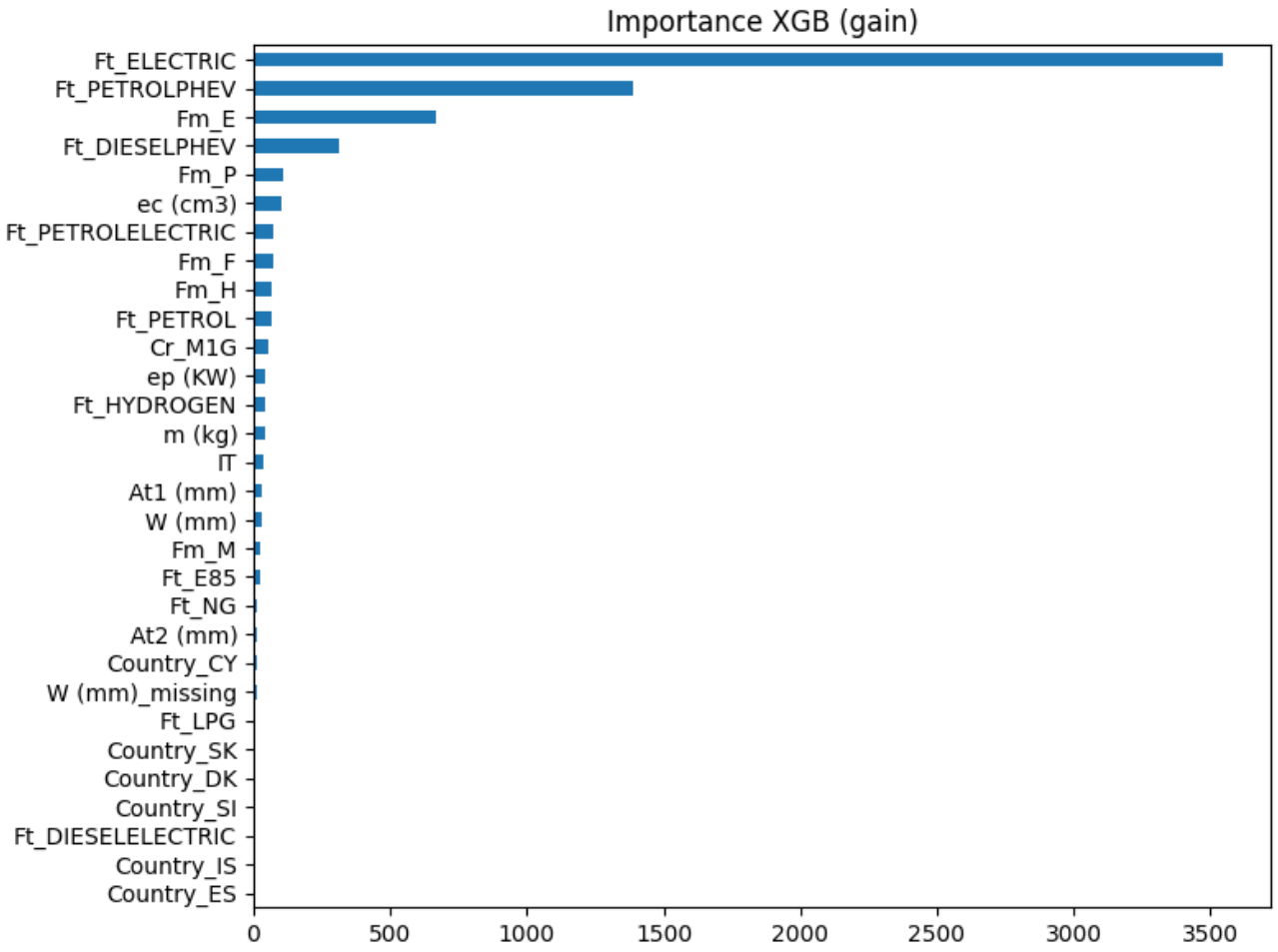


Importance des variables — XGBoost (Gain)

Objectif du graphique

Ce visuel présente les **20 variables les plus importantes selon le critère de gain** du modèle XGBoost.

Le *gain* mesure la réduction moyenne de la fonction de perte obtenue lors de l'utilisation d'une variable donnée pour diviser une branche de l'arbre. Il reflète donc l'impact réel d'une variable sur l'amélioration des décisions du modèle.



Rang	Variable	Importance_gain
1	Ft_ELECTRIC	3543.706787
2	Ft_PETROLPHEV	1385.981934
3	Fm_E	665.505371
4	Ft_DIESELPHEV	311.322906
5	Fm_P	108.086815
6	ec (cm3)	100.132469
7	Ft_PETROLELECTRIC	72.920021
8	Fm_F	72.832771
9	Fm_H	68.460449

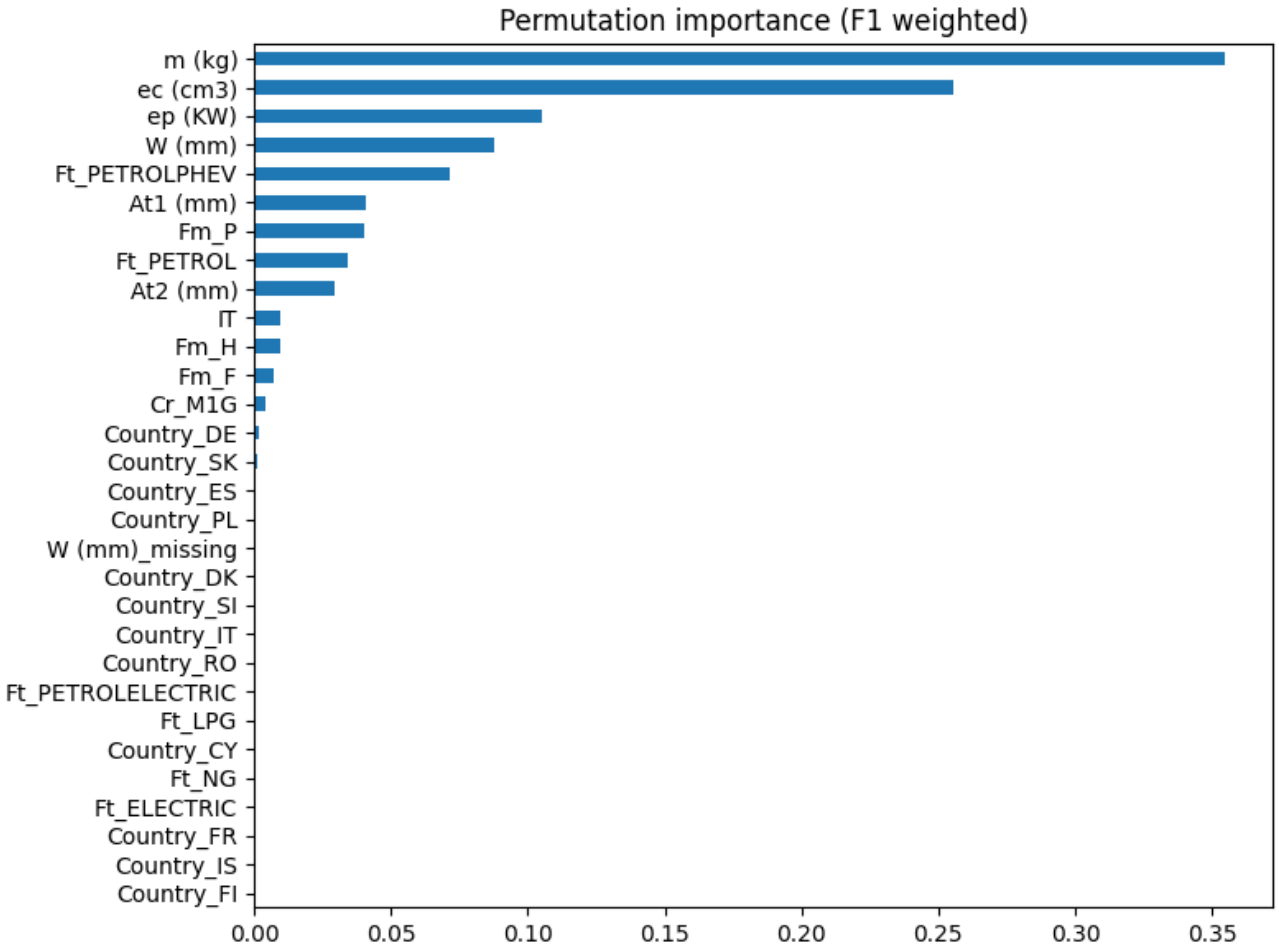
10	Ft_PETROL	65.201988
11	Cr_M1G	56.128819
12	ep (KW)	41.572922
13	Ft_HYDROGEN	39.602661
14	m (kg)	39.598667
15	IT	34.152302
16	At1 (mm)	31.214884
17	W (mm)	27.398220
18	Fm_M	23.392092
19	Ft_E85	21.488998
20	Ft_NG	14.628716

Importance par permutation — XGBoost (F1-score pondéré)

Objectif du graphique

Ce visuel présente les **20 variables les plus importantes selon l'importance par permutation**, mesurée ici par la **baisse du F1-score pondéré** lorsque les valeurs d'une variable sont aléatoirement permutées.

Contrairement à l'importance par gain, cette méthode évalue **L'utilité prédictive** des variables donc leurs impacts réels sur la performance finale du modèle.



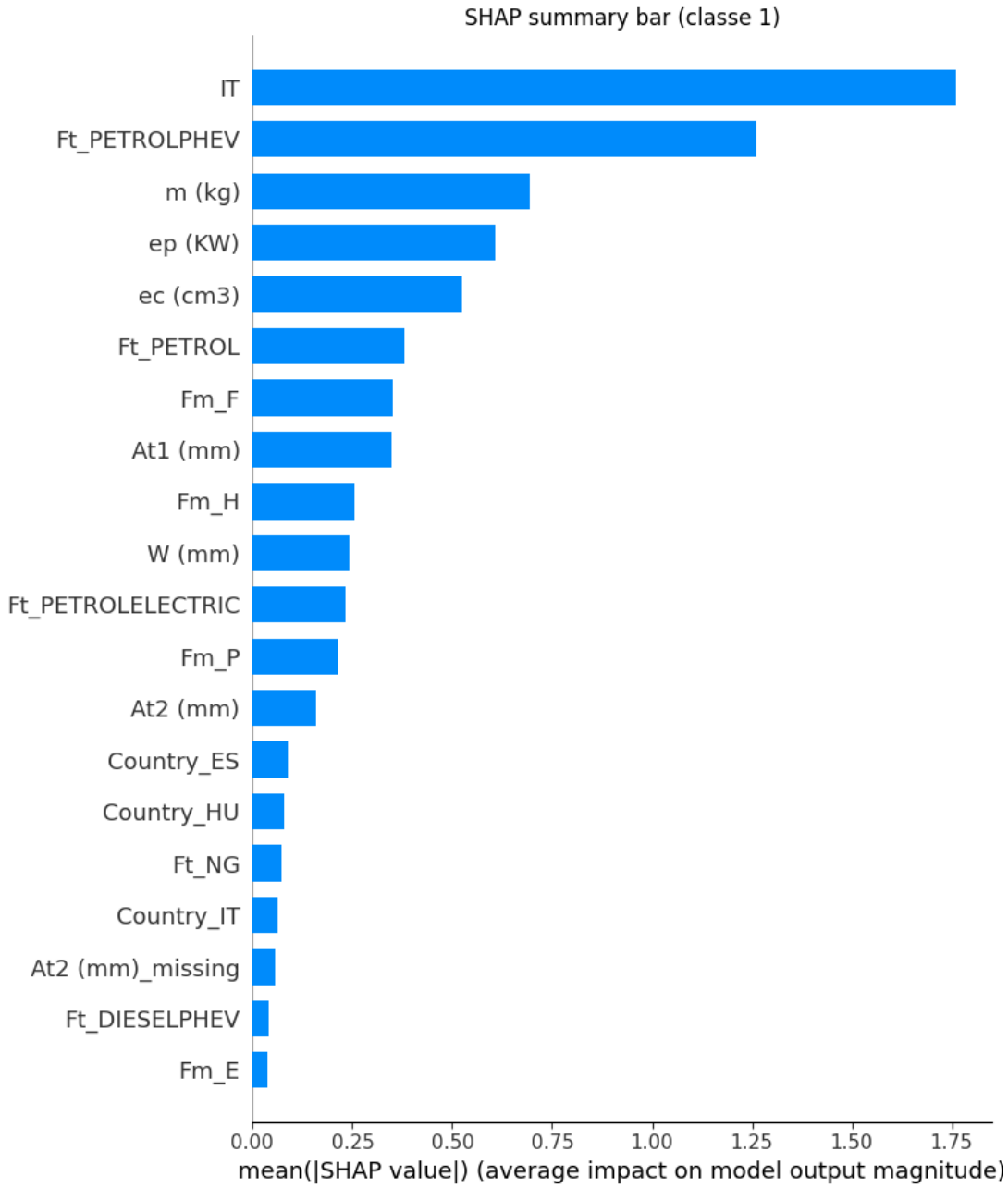
Rang	Variable	Permutation_importance_F1_weighted
1	m (kg)	0.354395
2	ec (cm3)	0.255245
3	ep (KW)	0.105083
4	W (mm)	0.087706
5	Ft_PETROLPHEV	0.071810
6	At1 (mm)	0.041129
7	Fm_P	0.040545
8	Ft_PETROL	0.034562
9	At2 (mm)	0.029733

10	IT	0.009644
11	Fm_H	0.009338
12	Fm_F	0.007130
13	Cr_M1G	0.004167
14	Country_DE	0.002048
15	Country_SK	0.001207
16	Country_ES	0.000876
17	Country_PL	0.000855
18	W (mm)_missing	0.000815
19	Country_DK	0.000795
20	Country_SI	0.000548

SHAP Summary Bar — Classe 1

Objectif du graphique

Ce graphique présente l'importance des variables **spécifiquement pour la classe 1**, mesurée par la moyenne des valeurs absolues SHAP. Il permet d'identifier ce qui **fait basculer un véhicule vers cette classe** plutôt qu'une autre.



Rang	Variable	SHAP_mean_abs_all_classes
1	m (kg)	1.640421
2	ec (cm3)	1.494292
3	ep (KW)	0.607350
4	At1 (mm)	0.433375
5	W (mm)	0.375113
6	IT	0.370918

7	Fm_P	0.292255
8	Ft_PETROL	0.287307
9	At2 (mm)	0.227392
10	Ft_PETROLPHEV	0.200790
11	Fm_H	0.190824
12	Ft_ELECTRIC	0.135141
13	Fm_F	0.126485
14	Cr_M1G	0.076248
15	Country_DE	0.065632
16	Fm_E	0.053824
17	Country_ES	0.032685
18	Ft_PETROLELECTRIC	0.031645
19	W (mm)_missing	0.030034
20	Country_PL	0.023584

SHAP Summary (nuage) — Classe 1

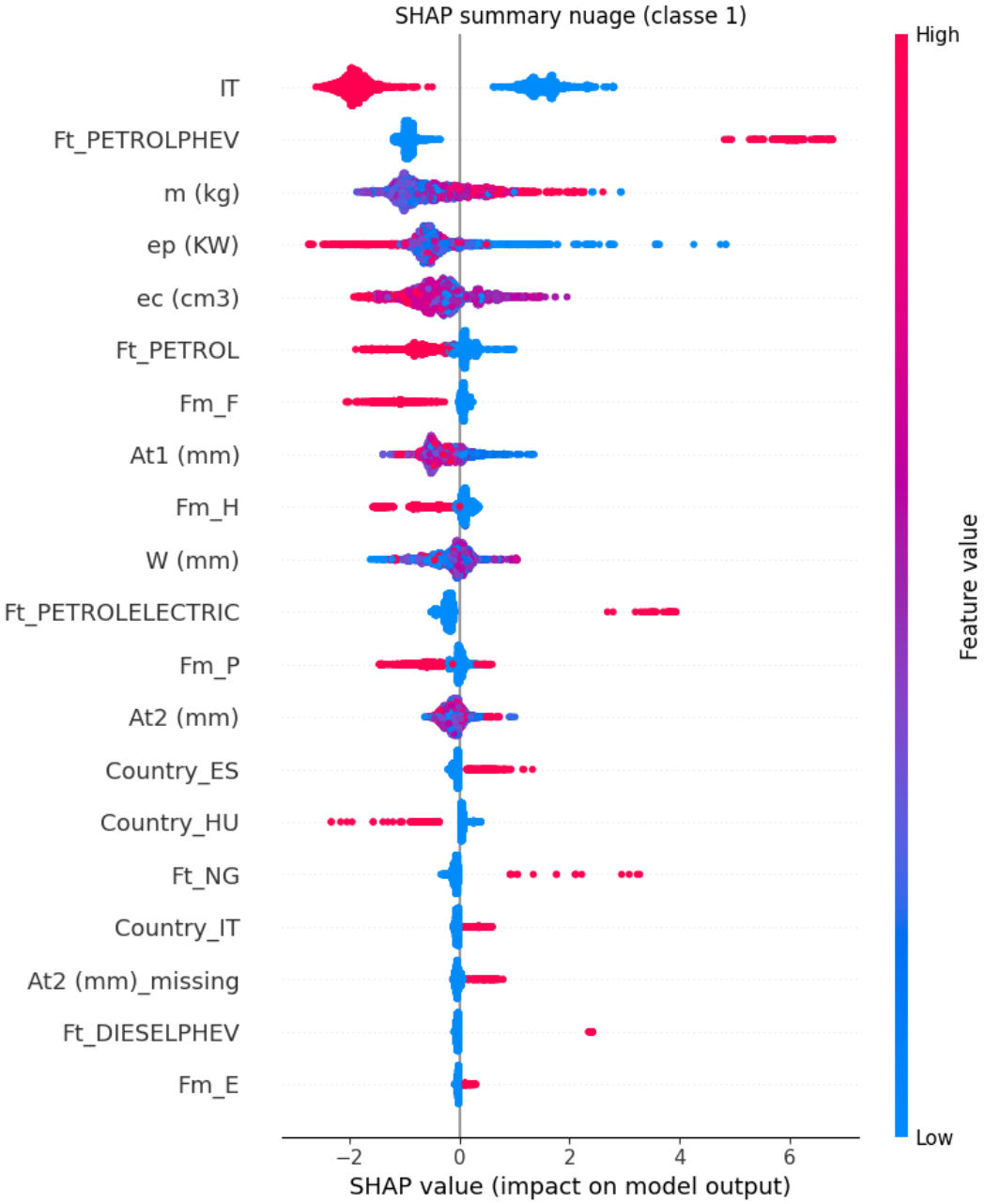
Objectif du graphique

Le nuage SHAP permet de visualiser à la fois :

- l'importance,

- le **sens de l'impact**,

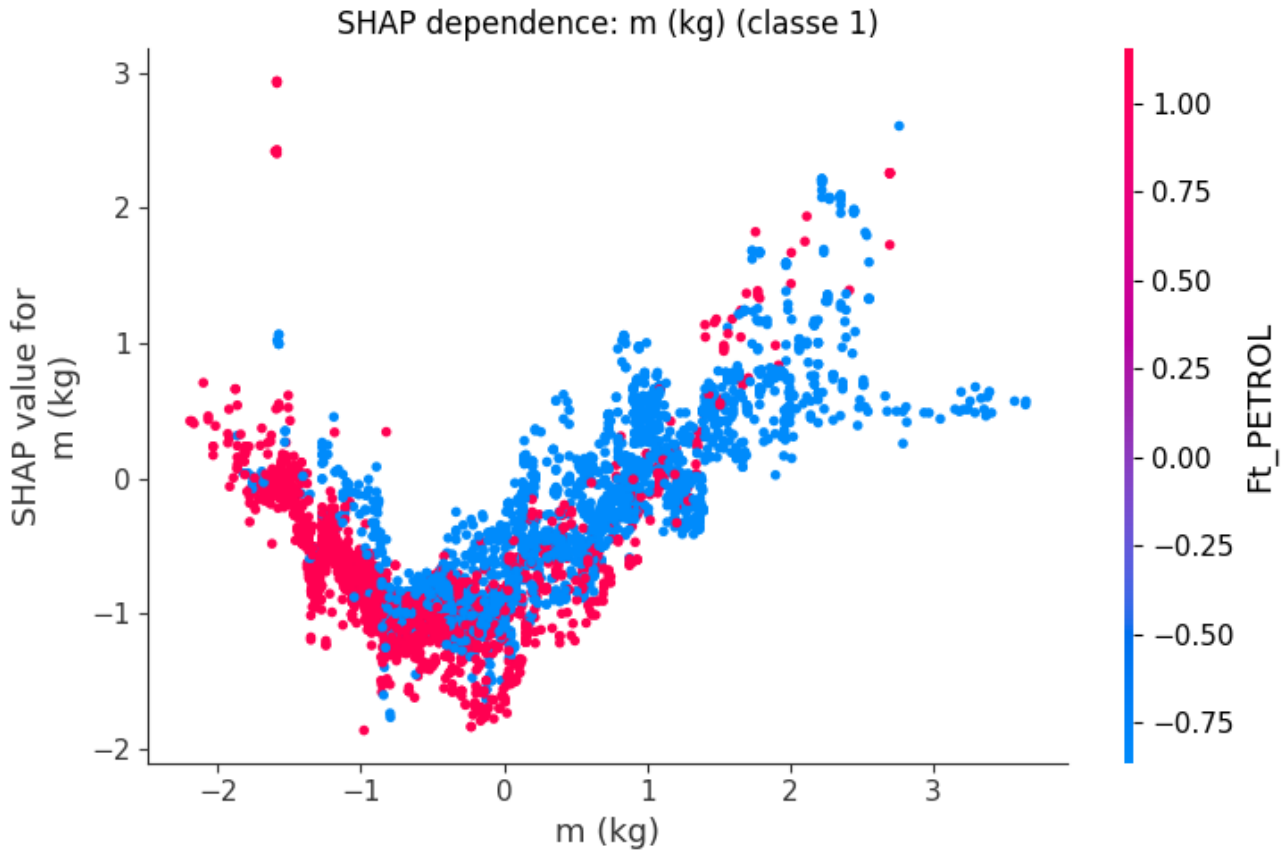
- et la **dispersion des effets** des variables pour la classe 1.



SHAP Dependence Plot — `m (kg)` (Classe 1)

Objectif du graphique

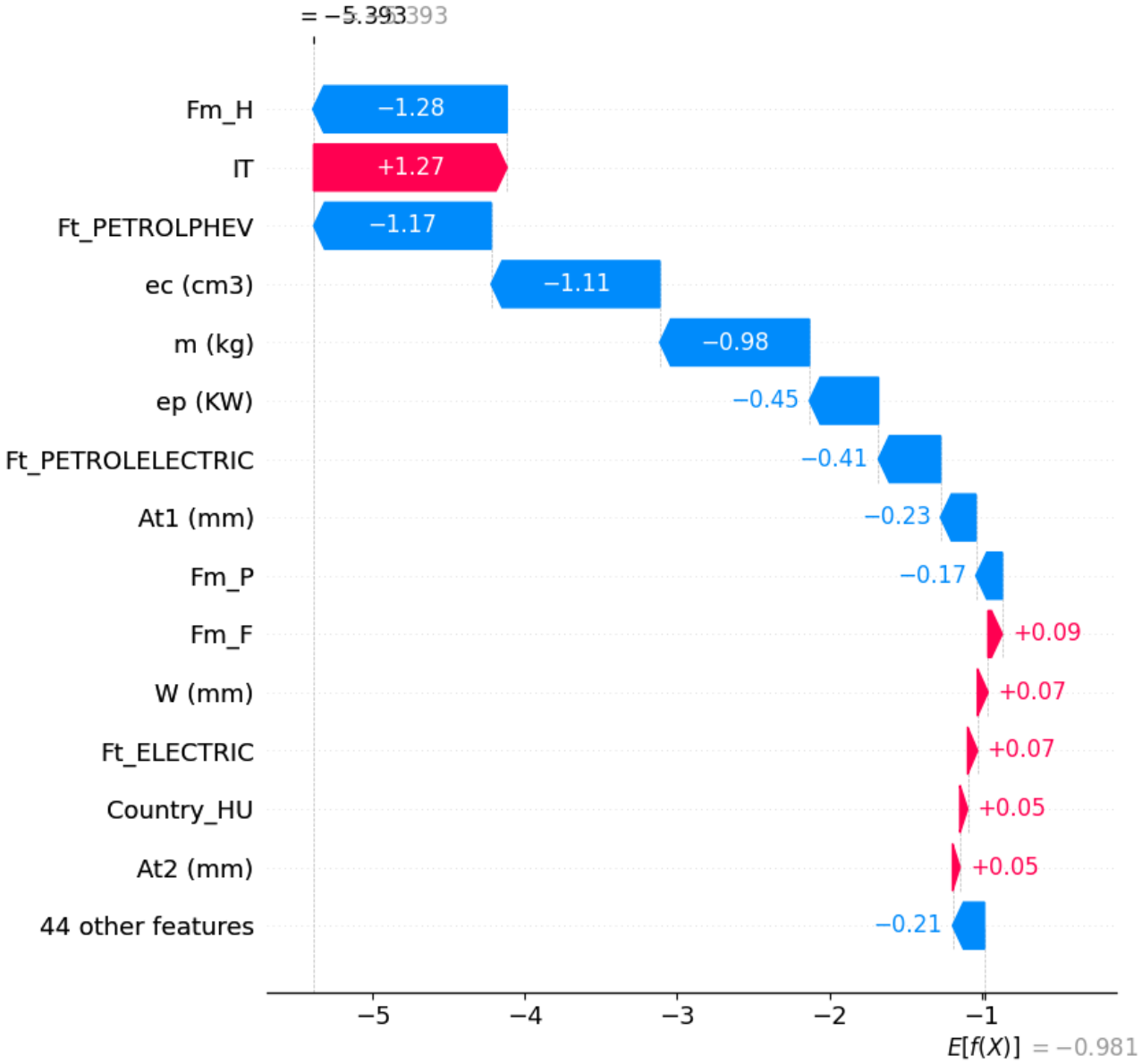
Ce graphique analyse l'effet marginal de la **masse** sur la prédiction de la classe 3, avec une coloration par une autre variable (interaction).



SHAP Waterfall — Explication locale (Classe prédite)

Objectif du graphique

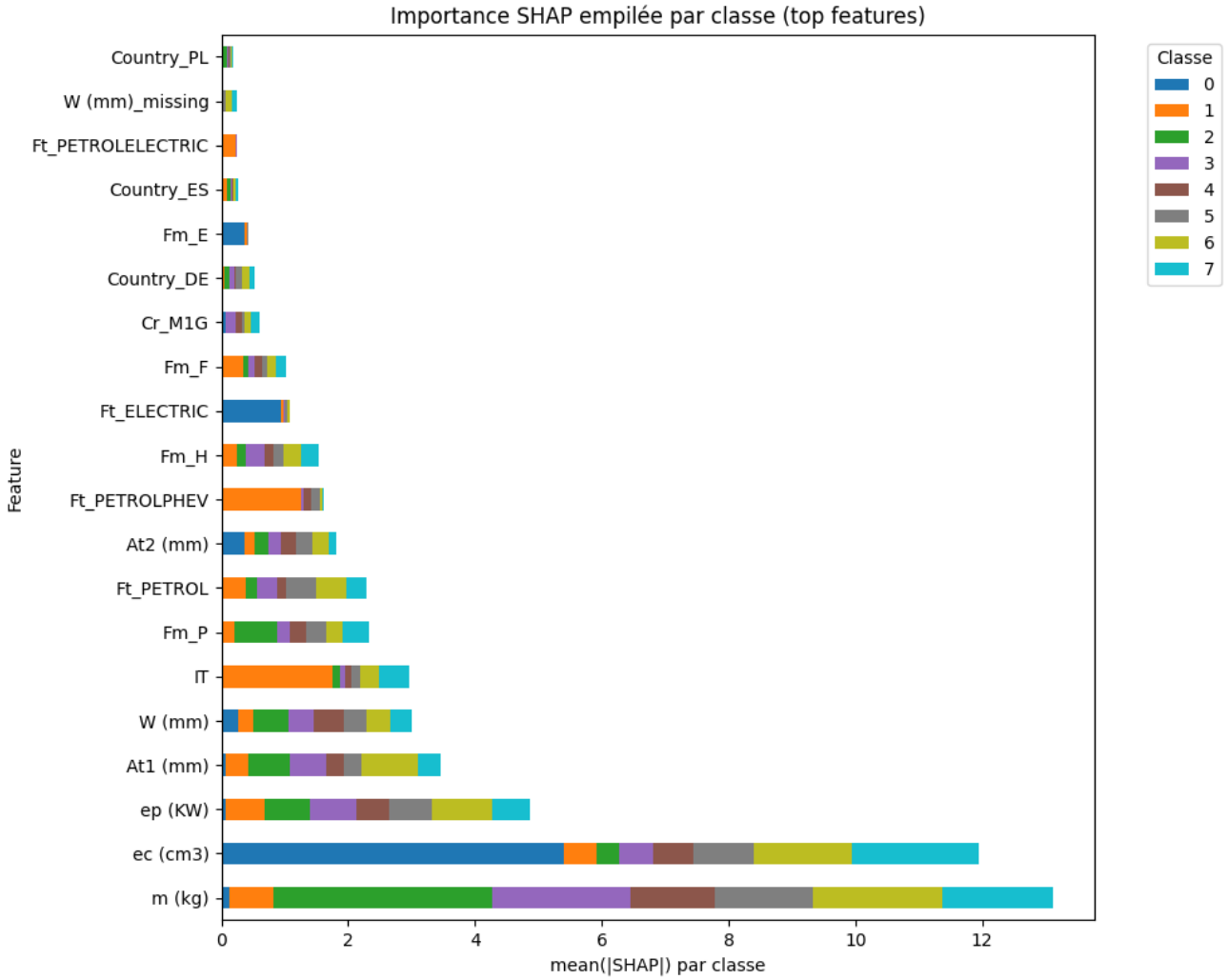
Le waterfall plot explique **une prédiction individuelle**, en décomposant la contribution de chaque variable depuis la valeur moyenne du modèle jusqu'au score final.



Importance SHAP empilée par classe (vue comparative)

Objectif du graphique

Ce graphique compare l'importance des variables **pour chaque classe**, via des barres empilées représentant la moyenne des valeurs absolues SHAP.



Synthèse globale explicabilité

- Les décisions du modèle sont pilotées par des **facteurs physiques et énergétiques plausibles**.

- Les effets sont monotones, continus et cohérents avec la réglementation.

- Les erreurs se concentrent logiquement aux **frontières de classes**.

- Le modèle est à la fois **performant, explicable et fiable**.

Cette vue confirme que le modèle apprend une **structure globale cohérente**, tout en adaptant ses décisions aux spécificités de chaque classe CO_2.