Modélisation du projet

Rappel de l'objectif du projet :
Analyser les variables qui impactent les émissions de CO2 des véhicules en France pour l'année 2022

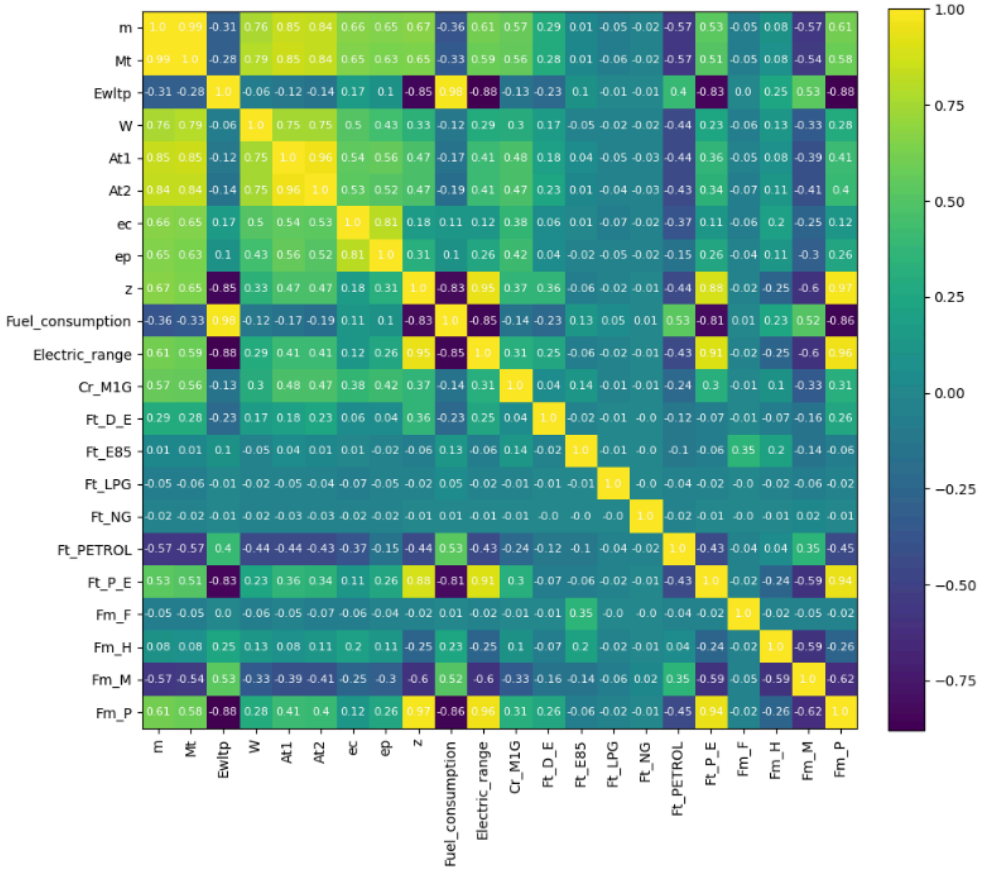
Nous avons donc affaire ici à une problématique d'estimation des émissions de CO2, pour les véhicules vendus en France pour l'année 2022.

Nous avons choisi dans un premier temps de tester plusieurs modèles de régression linéaire en utilisant les techniques de Machine Learning apprises.

Dans un second temps, nous avons testé un algorithme de régression linéaire utilisant les réseaux de neurones Dense du Deep Learning.

Complément de pré-processing

Suite au pré-processing initial, nous nous sommes aperçus qu'il subsistait des variables qui étaient d'une part fortement corrélées à la cible et d'autre part corrélées entre elles.



A partir de la heatmap ci-dessus, nous constatons que :

- La variable **Fuel_consumption**[1] présente une colinéarité avec notre variable cible. Ce qui est logique puisque les émissions de CO2 découlent directement de la consommation de carburant des véhicules.

- La variable **At2**[2], est extrêmement corrélée à la variable **At1** car ces variables correspondent aux essieux avant et arrière de la voiture, qui fonctionnent exactement de la même manière et font la même taille ce qui induit qu'ils ont exactement le même impact sur les émissions de Co2.

- Les variables **Electric_range**[3] et **z**[4]**,** qui sont liées à l'autonomie électrique des véhicules, ce qui n'entre pas en considération dans notre projet puisque, comme expliqué dans le rapport d'explorations, les véhicules électriques n'émettent pas de Co2.

- La variable **ec**[5] (engine capacity) est fortement corrélée à la variable **ep**[6] (engine power), l'une étant directement dépendante de l'autre.

Création des différents jeux de données

Pour nous aider à déterminer le meilleur modèle à partir des différentes variables, nous avons testé différents jeux de données en conservant/supprimant certaines variables afin d'analyser leur impact sur les données que l'on obtient :

- Un premier où nous avons uniquement retiré les variables At2, Electric_range et z (*no_at2_z_er*)

- Un second où nous avons retiré en plus la variable ec (*no_at2_z_er_ec*)

- Un troisième où nous avons retiré les variables Ft_P_E et Fm_P qui sont fortement corrélées à notre variable cible (*no_at2_z_er_ec_ftpe_fmp*).

[1] Document entre l'émission de CO2 et la consommation : https://www.econologie.com/emissions-co2-litre-carburant-essence-diesel-ou-gpl/
[2] At1, At2: Largeur de l'essieu directeur, arrière
[3] Electric range (km) : Autonomie électrique
[4] z (Wh/km) : Consommation électrique
[5] ec (cm3): Cylindrée
[6] ep (Kw): Puissance du moteur

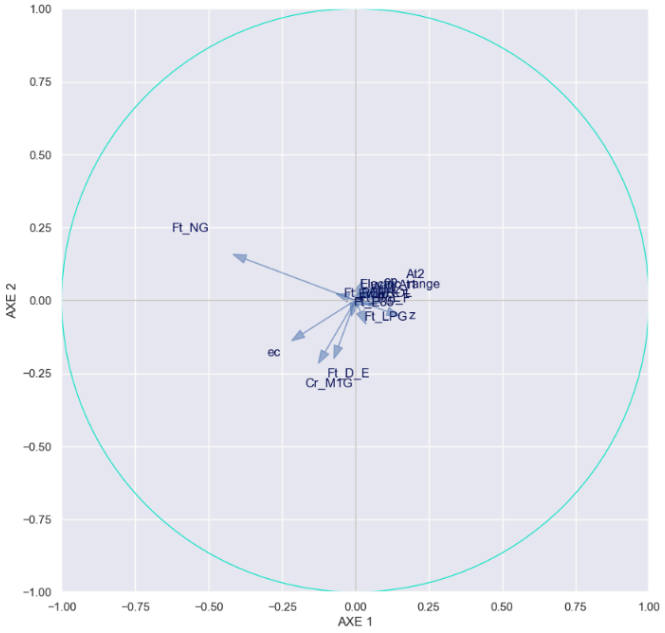
Métrique de référence

La métrique de performance utilisée est le Score R2 (version normalisée de la Mean Squared Error) car celle-ci facilite la comparaison entre les différents modèles.

Choix du modèle et optimisation

Abandon de la PCA

Nous avons essayé de réduire le nombre de dimensions en utilisant une PCA, mais cela n'a pas été concluant (énorme perte d'informations car peu de corrélation) puisqu'il n'était pas possible d'interpréter les corrélations obtenues entre les variables :



Recherche du meilleur modèle avec GridSearch

Nous avons entraîné un GridSearch avec les différents modèles de régression linéaire ci-dessous :

- LinearSVR
- LinearRegression
- Ridge
- Lasso
- ElasticNet

en utilisant les trois jeux de données suivants :

- *no_at2_z_er_ec*
- *no_at2_z_er*
- *no_at2_z_er_ec_ftpe_fmp*

Initialement, nous souhaitions également utiliser des modèles comme le RandomForest ou le SVR, mais nous avons rencontré des problèmes lors de l'exécution de ceux-ci (temps de calcul trop long et/ou mémoire insuffisante).

Résultats obtenus

Modèle	**Jeu de données**	**Score**	**Paramètres**
LinearRegression	no_at2_z_er	0.912640	fit_intercept=False
LinearSVR	no_at2_z_er	0.912638	C=10.0, fit_intercept=False, loss=squared_epsi...
RidgeCV	no_at2_z_er	0.912637	alpha=0.1, fit_intercept=True
RidgeCV	no_at2_z_er_ec_ftpe_fmp	0.911958	alpha=0.1, fit_intercept=True
RidgeCV	no_at2_z_er_ec	**0.911957**	alpha=0.1, fit_intercept=True
LinearSVR	no_at2_z_er_ec_ftpe_fmp	0.911956	C=10.0, fit_intercept=True, loss=squared_epsil...
LinearSVR	no_at2_z_er_ec	0.911956	C=10.0, fit_intercept=False, loss=squared_epsi...
LinearRegression	no_at2_z_er_ec	**0.911954**	fit_intercept=False
LinearRegression	no_at2_z_er_ec_ftpe_fmp	0.911954	fit_intercept=True
Lasso	no_at2_z_er_ec	0.908417	alpha=0.1, fit_intercept=True, max_iter=2000
Elastic_Net	no_at2_z_er_ec	0.908417	alpha=0.1, fit_intercept=True, l1_ratio=1, max...
Lasso	no_at2_z_er	0.906490	alpha=0.1, fit_intercept=True, max_iter=2000
Elastic_Net	no_at2_z_er	0.906490	alpha=0.1, fit_intercept=True, l1_ratio=1, max...
Lasso	no_at2_z_er_ec_ftpe_fmp	0.903778	alpha=0.1, fit_intercept=True, max_iter=2000
Elastic_Net	no_at2_z_er_ec_ftpe_fmp	0.903778	alpha=0.1, fit_intercept=True, l1_ratio=1, max..

Suite à ces résultats, nous observons que les résultats sont relativement proches les uns des autres.
Le LinearRegression étant le modèle le plus rapide et celui que nous avons le plus souvent été amené à utiliser durant la formation, nous le choisissons avec le jeu de données épuré de toute corrélation *no_at2_z_er_ec.*

Bagging, Boosting

Par la suite, dans l'optique d'améliorer davantage la performance de notre modèle, nous avons testé des modèles de Boosting et de Bagging, mobilisables dans une problématique de régression telles que AdaBoostRegressor et BaggingRegressor.
Toutefois, nous avons constaté que les résultats restent sensiblement identiques avec le Bagging et dégradés avec le Boosting.

model	df	score_train	score_test	RMSE_train	RMSE_test
BaggingRegressor	no_at2_z_er_ec	0.911334	0.911956	14.187333	14.145094
AdaBoostRegressor	no_at2_z_er_ec	0.887232	0.885628	15.999829	16.121921

Régression linéaire avec un réseau de neurones Denses

Par ailleurs, nous avons également entraîné un modèle de régression en Deep Learning afin de tester la performance des réseaux de neurones sur notre problématique d'émissions de CO2. Ce dernier affiche de très bons résultats, puisque nous obtenons un score de 0,98.
Par manque de temps et de maîtrise des modèles de Deep Learning, nous n'en avons testé qu'un seul.

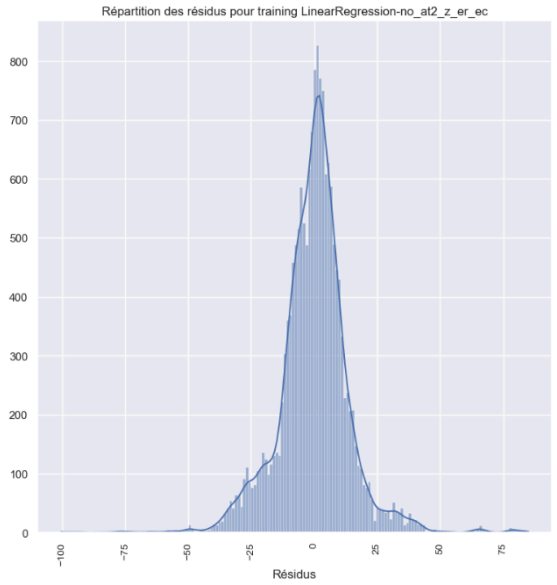
Layer (type)	Output Shape	Param #
dense_16 (Dense)	(None, 16)	272
dense_17 (Dense)	(None, 128)	2,176
dense_18 (Dense)	(None, 2048)	264,192
dropout_4 (Dropout)	(None, 2048)	0
dense_19 (Dense)	(None, 256)	524,544
dense_20 (Dense)	(None, 256)	65,792
dropout_5 (Dropout)	(None, 256)	0
dense_21 (Dense)	(None, 256)	65,792
dense_22 (Dense)	(None, 128)	32,896
dense_23 (Dense)	(None, 1)	129

Total params: 2,867,381 (10.94 MB)
Trainable params: 955,793 (3.65 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 1,911,588 (7.29 MB)

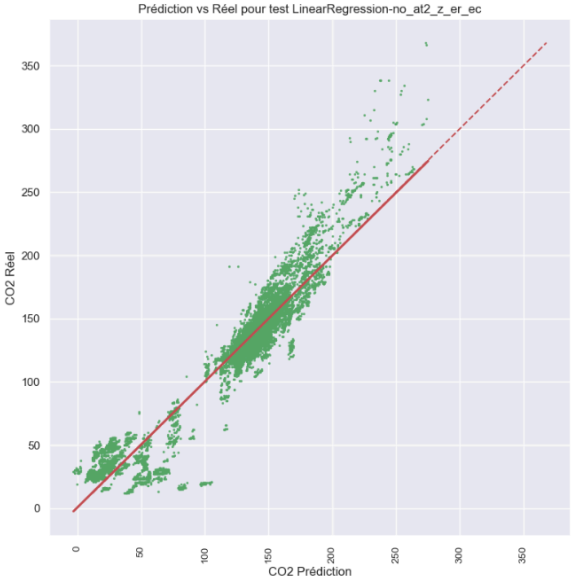
Scores DNN :
R2 Train: 0.9876576285698524
R2 Test: 0.9884528242490008

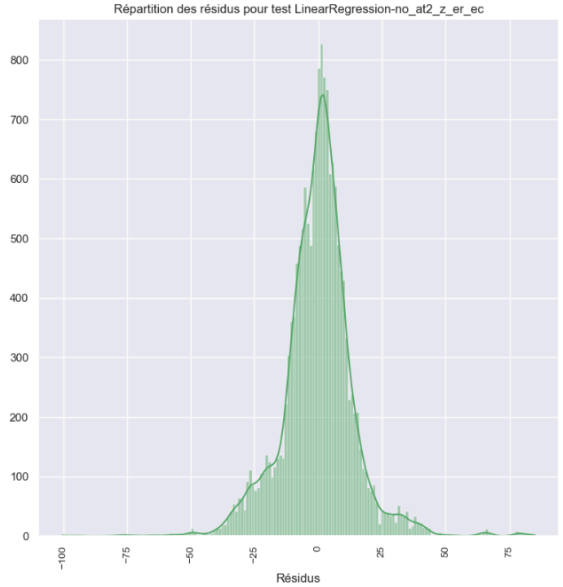
Visualisation des résultats

Régression Linéaire pour le jeu de train:

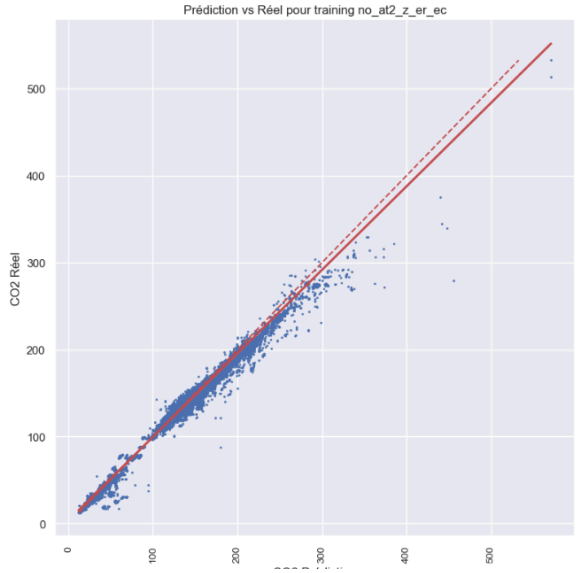


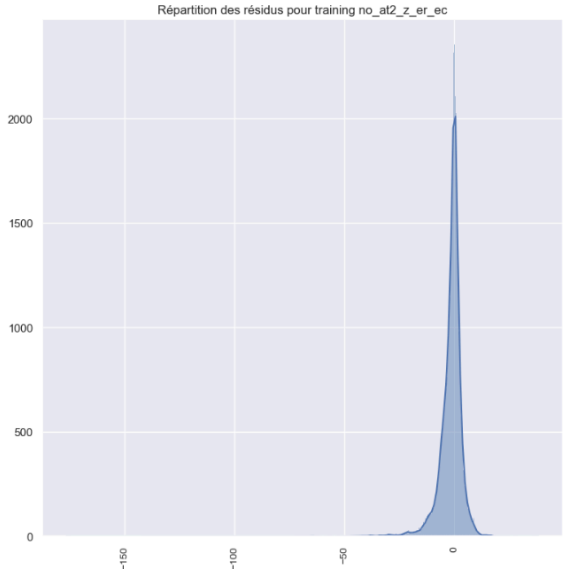
Régression Linéaire pour le jeu de test:



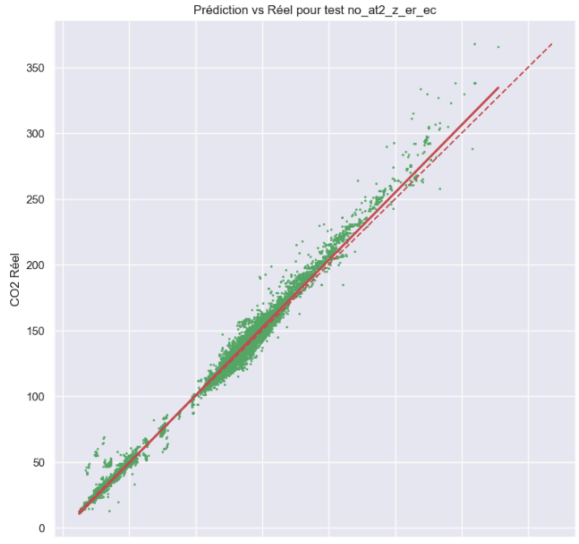


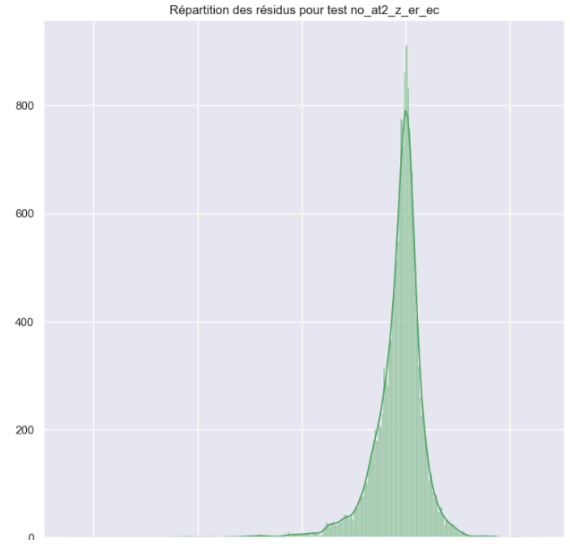
Réseau de neurone Dense Train :





Réseau de neurone Dense Test:

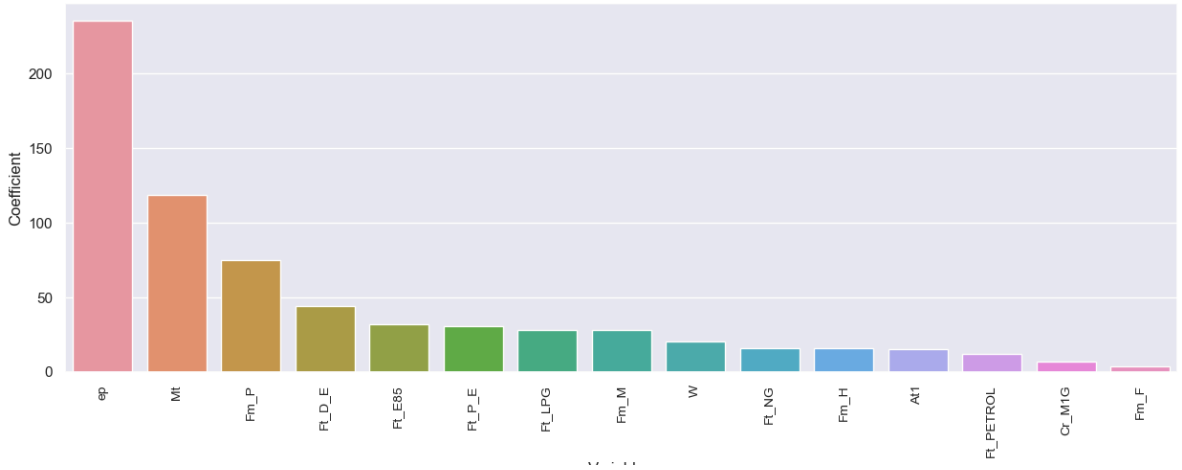




Nous observons une forte linéarité ce qui confirme les résultats obtenus avec les modèles de Machine Learning. Toutefois, nous observons une répartition des résidus centrée et très étendue (données autant sur-évaluées que sous-évaluées).

Interprétation des résultats

Nous avons analysé l'impact respectif des variables catégorielles (en valeur absolue) sur la variable cible via le graphique suivant :



On voit que la puissance du moteur, la masse et le carburant de type hybride sont les variables qui impactent le plus fortement les émissions de CO_2 des véhicules.

Quelques limites de notre projet : Nous avons délibérément fait le choix de restreindre notre étude aux données concernant la France en 2022 en raison de la taille importante des jeux de données et de la puissance limitée de nos ordinateurs. Ainsi, cela limite la possibilité d'extrapoler nos résultats au-delà de ce contexte.