kaggle

Estimate CO2 emissions from cars

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Roll Johanna
Yakoub Mélanie

Sommaire

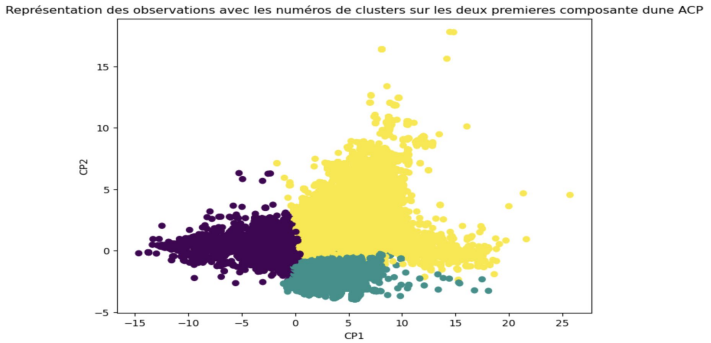
Sommaire

Data Visualisation

Analyse Exploratoire en Bref

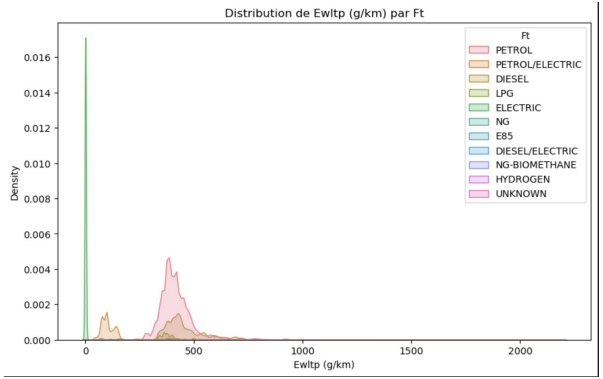
- K-Means :
  - Identification de clusters.
  - Insights sur la structure des données.
- ACP (Analyse en Composantes Principales) :
  - Réduction de dimension.
  - Visualisation de la variance explicative.
- AutoCarver :
  - Discrétisation des variables.
  - Sélection des features.
- Matrice de Corrélation :
  - Relations entre les variables.
  - Identification de corrélations significatives.

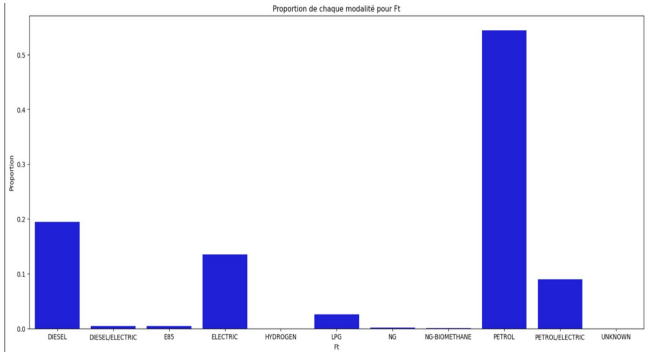
Objectif : Obtention d'informations essentielles pour guider les prochaines étapes de modélisation et d'analyse.

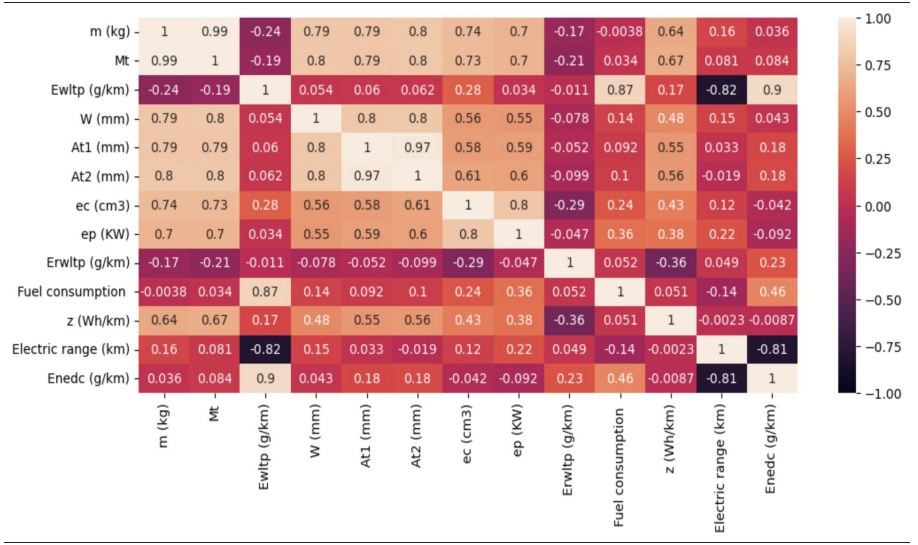


UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Data Visualisation



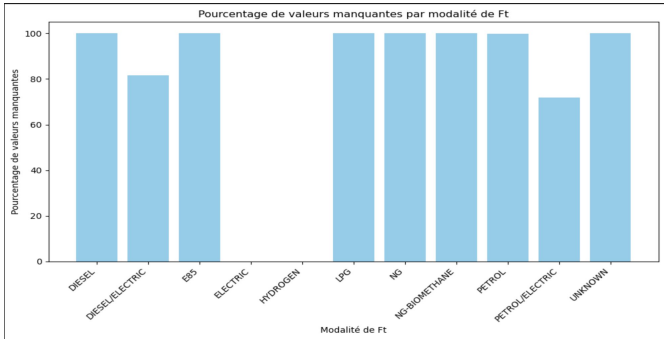




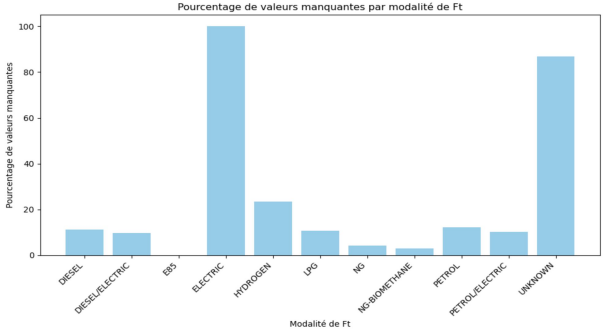
UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Imputation

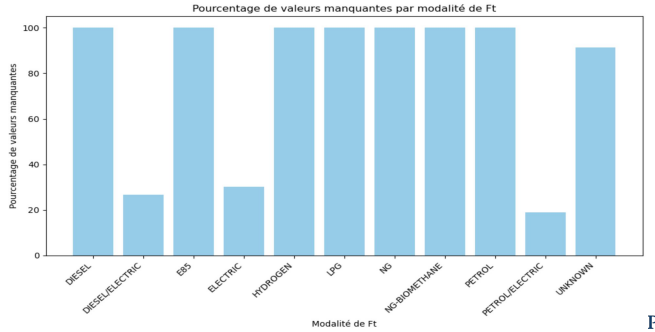
Enedc (g/km)



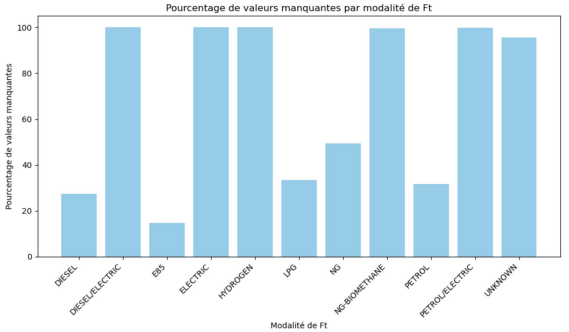
Fuel consumption



Electric range (km)



Erwltp (g/km)



UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Imputation

Imputation avec médiane pour Train et Test fusionnés car les distribution très similaires sur ces 2 ensembles

→ Augmenter les performances

Approche :

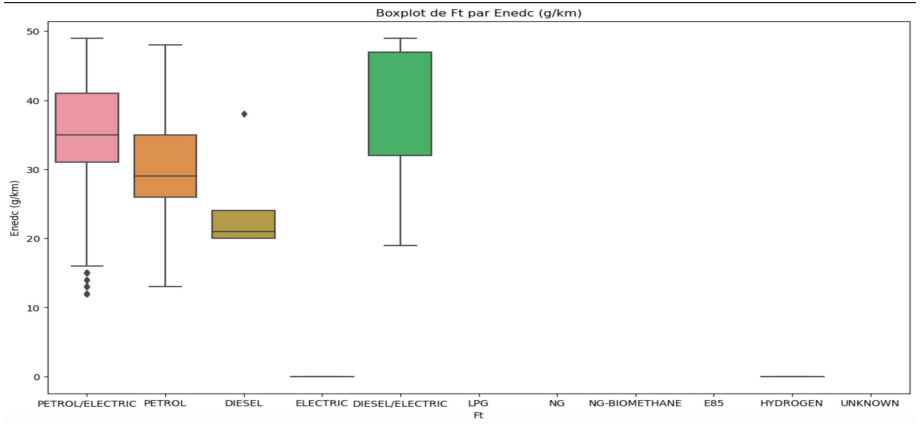
- Médiane utilisée pour imputer les valeurs manquantes.
- Fusion des ensembles de données train et test pour assurer une cohérence.

Critères de Groupement :

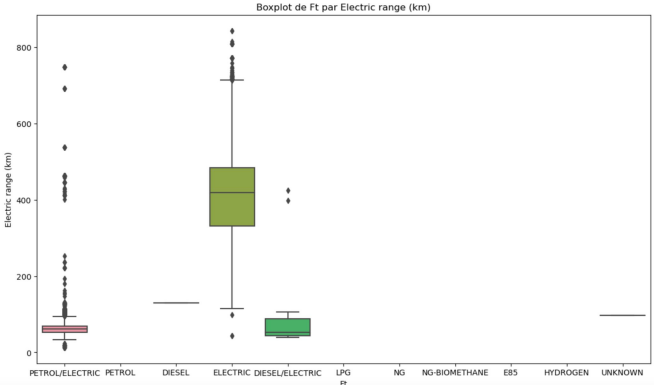
- Pourcentage faible de valeurs manquantes par modalité.
- Différences de médianes significatives entre les groupes.

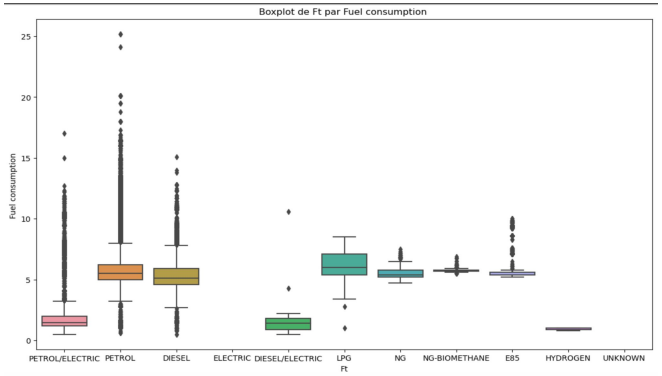
Variable Enedc (g/km) :

- Groupement par modalité de la variable Ft.
- Séparation hydro/électrique d'un côté et le reste de l'autre pour réduire le bruit.



Imputation



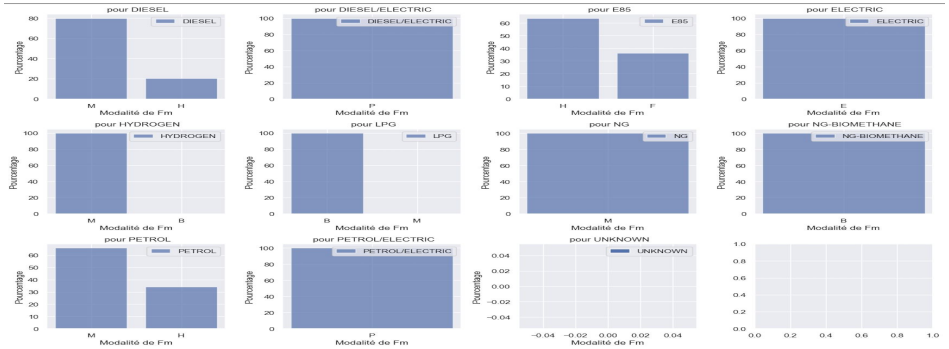


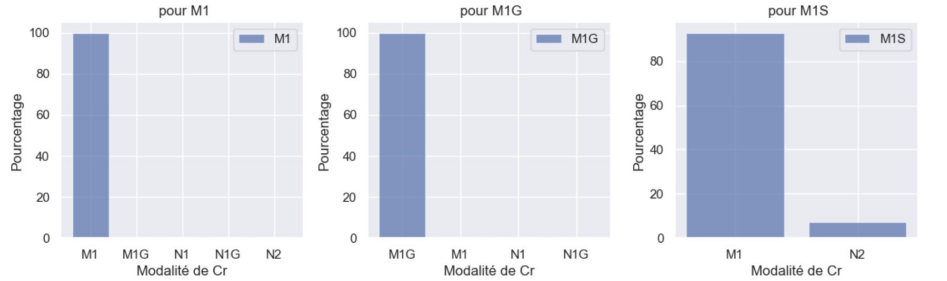
- **Imputation de Electric Range (km) :** Médiane avec groupement, séparant les véhicules électriques. Cette approche vise à améliorer la précision en prenant en compte les caractéristiques distinctives de cette catégorie spécifique de véhicules.

- **Imputation de Fuel Consumption, Erwltp et autres variables quantitatives :** Méthode d'imputation par médiane. L'uniformité de cette méthode sur l'ensemble des variables quantitatives assure cohérence et fiabilité dans le processus d'imputation.

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Imputation





- **Imputation de Fm :** Mode avec groupement par catégorie de Ft pour renforcer la précision en considérant les variations spécifiques de Ft.
- **Imputation de Ct :** Mode avec groupement par catégorie de Cr pour une adaptation précise aux caractéristiques distinctives de Cr.
- **Imputation des Autres Variables Catégorielles :** Imputation par le mode pour le reste des variables, assurant uniformité, cohérence, et stabilité.

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Imputation

Création de deux nouvelles variables : Imputation par Random Forest pour Mp (classification) et Fuel Consumption (régression), exploitant la puissance prédictive du modèle pour une imputation précise et adaptée.

```python
def imputer_valeurs_manquantes_random_forest_classif(df_apred_dumFt, df_dumFt, colonnes_a_imputers):
    df_imputed1 = df_dumFt.copy()
    df_imputed2 = df_apred_dumFt.copy()  # Place de encoded
    df_concatenated = pd.concat([df_dumFt, df_apred_dumFt])
    df_imputed = df_concatenated.copy()

    # Remarque: pour ne pas imputer les prochaines colonnes sur la base des colonnes imputées
    df_dumFt = supr_col(df_dumFt)
    df_apred_dumFt = supr_col(df_apred_dumFt)
    df_concatenated = supr_col(df_concatenated)

    for col in colonnes_a_imputers:
        Ensemble_test_target, Ensemble_train_target = selection_ensemble_de_test(df_apred_dumFt, df_dumFt, [col])
        colonnes_avec_valeurs_manquantes_target = colonnes_non_candidates(Ensemble_test_target)

        Ensemble_test_target_Y, Ensemble_test_target_X, Ensemble_train_target_Y, Ensemble_train_target_X = train_test_X_Y(col, Ensemble_test_target,

        # Use RandomForestClassifier instead of RandomForestRegressor
        model = RandomForestClassifier(n_estimators=120, random_state=42) # Parametre a ameliorer si ressources technique disponible
        print(f"Taille de l'ensemble d'entraînement pour la colonne {col}: {Ensemble_train_target_X.shape}")
        model.fit(Ensemble_train_target_X, Ensemble_train_target_Y)
        predicted_values = model.predict(Ensemble_test_target_X)

        df_imputed.loc[df_imputed[col].isna(), col] = predicted_values
        # utilisation de df_imputed
        # pour les performances

        print(col)
        print(Ensemble_train_target_X.columns)
        print(model.feature_importances_)  # Check feature importances if needed

        # Utiliser accuracy_score pour évaluer la précision du modèle
        accuracy = accuracy_score(Ensemble_train_target_Y, model.predict(Ensemble_train_target_X))
        print(f'Précision estimée sur l\'ensemble train pour la colonne {col}: {accuracy}')

    return df_imputed
```

```
Taille de l'ensemble d'entraînement pour la colonne Mp: (148849, 23)
Mp
Index(['Ft_DIESEL', 'Ft_DIESEL/ELECTRIC', 'Ft_E85', 'Ft_ELECTRIC',
       'Ft_HYDROGEN', 'Ft_LPG', 'Ft_NG', 'Ft_NG-BIOMETHANE', 'Ft_PETROL',
       'Ft_PETROL/ELECTRIC', 'Ft_UNKNOWN', 'Fm_B', 'Fm_E', 'Fm_F', 'Fm_H',
       'Fm_M', 'Fm_P', 'Ct_M1', 'Ct_M1G', 'Ct_N1', 'electric_range_binary',
       'fuel_consumption_binary', 'Enedc (g/km)_binary'],
      dtype='object')
[0.         0.42408516 0.         0.         0.         0.
 0.         0.         0.         0.43543817 0.         0.
 0.         0.         0.         0.         0.         0.07116679
 0.06930988 0.         0.         0.         0.         ]
Précision estimée sur l'ensemble train pour la colonne Mp: 0.33306236521575555
```

```
Taille de l'ensemble d'entraînement pour la colonne Fuel consumption : (166577, 33)
Fuel consumption
Index(['Ft_DIESEL', 'Ft_DIESEL/ELECTRIC', 'Ft_E85', 'Ft_ELECTRIC',
       'Ft_HYDROGEN', 'Ft_LPG', 'Ft_NG', 'Ft_NG-BIOMETHANE', 'Ft_PETROL',
       'Ft_PETROL/ELECTRIC', 'Ft_UNKNOWN', 'Fm_B', 'Fm_E', 'Fm_F', 'Fm_H',
       'Fm_M', 'Fm_P', 'Ct_M1', 'Ct_M1G', 'Ct_N1', 'Mp_BMW', 'Mp_FORD',
       'Mp_HYUNDAI MOTOR EUROPE', 'Mp_KIA', 'Mp_MAZDA-SUBARU-SUZUKI-TOYOTA',
       'Mp_MERCEDES-BENZ', 'Mp_RENAULT-NISSAN-MITSUBISHI', 'Mp_STELLANTIS',
       'Mp_TESLA-HONDA-JLR', 'Mp_VOLKSWAGEN', 'electric_range_binary',
       'fuel_consumption_binary', 'Enedc (g/km)_binary'],
      dtype='object')
[0.         0.01024157 0.         0.         0.         0.
 0.         0.         0.         0.03885662 0.         0.
 0.         0.         0.         0.         0.         0.14754641
 0.         0.         0.09657839 0.08089215 0.05417866 0.06547258
 0.06694218 0.10563184 0.01851722 0.2351204  0.0143066  0.06571538
 0.         0.         0.         ]
MAE estimée sur l'ensemble train pour la colonne Fuel consumption : 0.23262708655098577
```

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Encodage des variables catégorielles

➢ **Encodage One-Hot :**
- Pour les colonnes catégoriques avec moins de 29/30 modalités.
- Création de nouvelles colonnes binaires pour chaque modalité.
- Facilite la compréhension des relations entre les différentes modalités par le modèle.

➢ **Encodage par Comptage :**
- Pour les colonnes catégoriques avec plus de 29/30 modalités.
- Utilisation du Count Encoder de **category_encoders**.
- Compte simplement le nombre d'occurrences de chaque modalité, évitant la création excessive de nouvelles colonnes.

Objectif : Préparer nos données catégorielles de manière optimale pour une intégration efficace dans nos modèles d'apprentissage automatique.

Sommaire

XG-Boost

eXtreme Gradient Boosting

Algorithme d'apprentissage automatique utilisé pour les problèmes de régression et classification

Appartient aux **méthodes ensemblistes** : plusieurs modèles sont combinés pour former un modèle global plus robuste et performant

- **Ensemble Boosting** : approche où plusieurs modèles faibles s'assemblent pour créer un modèle fort
- **Construction des Arbres** : utilise des arbres de décision successifs pour corriger les erreurs résiduelles des modèles précédents
- **Optimisation Rapide** : "rapide" et efficace, adapté à de grands ensembles de données
- **Régularisation Intégrée** : intègre des mécanismes de régularisation pour éviter le surajustement.
- **Évaluation de l'Importance des Caractéristiques** : évalue l'importance des variables, aidant à identifier les caractéristiques les plus influentes dans le modèle.
- **Adaptabilité Universelle** : convient à la fois à la classification et à la régression
- **Facilité de Parallélisation** : prend en charge la parallélisation, facilitant le traitement de grands ensembles de données sur des systèmes distribués

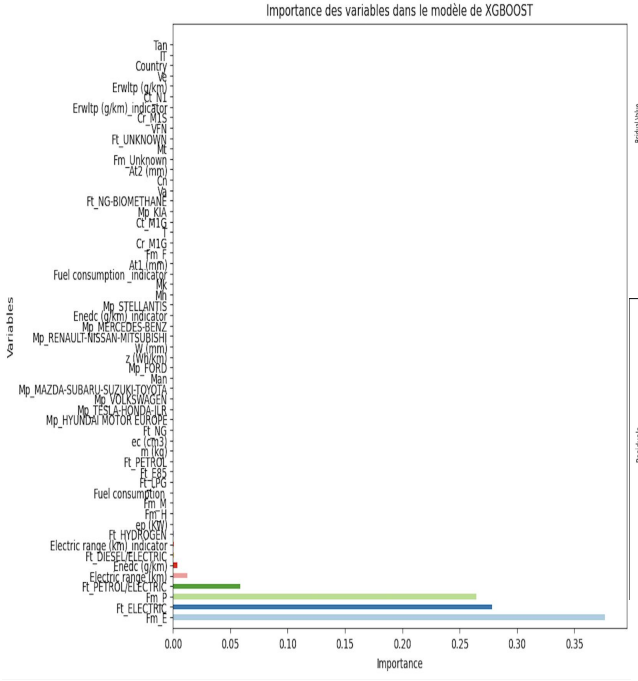
XG-Boost

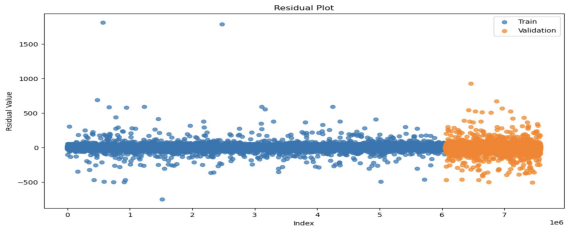
```python
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
import xgboost as xgb

# 1. Diviser les données d'entraînement
X_train, X_val, y_train, y_val = train_test_split(df_encoded_train, y_train_df, test_size=0.2, random_state=42)

# 2. Paramètres xgboost
xgb_model = xgb.XGBRegressor(
    n_estimators=4500,
    max_depth=35,
    learning_rate=0.001,
    colsample_bytree=0.75,
    gamma=8,
    reg_alpha=0.7,
    reg_lambda=0.2,
    objective='reg:squarederror',
    tree_method='hist',
    n_jobs=-1)

# 3. Entraîner le modèle
xgb_model.fit(X_train, y_train)

# 4. Évaluer le modèle sur l'ensemble de validation
y_val_pred = xgb_model.predict(X_val)
mae_val = mean_absolute_error(y_val, y_val_pred)
print('Mean Absolute Error on validation set:', mae_val)
```

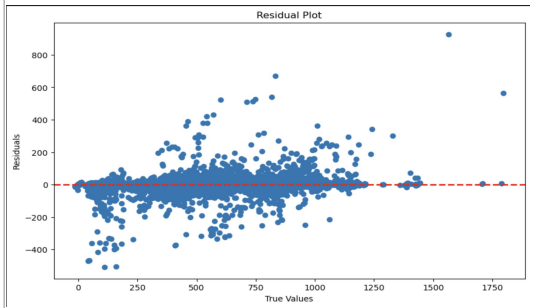
- **n_estimators** : Nombre d'arbres dans le modèle. Plus il y en a, plus le modèle est complexe.
- **max_depth** : Profondeur maximale des arbres. Limite la complexité du modèle.
- **learning_rate** : Taux d'apprentissage. Plus il est petit, plus l'apprentissage est prudent.
- **colsample_bytree** : Fraction de caractéristiques utilisées pour chaque arbre. Contrôle la variabilité des caractéristiques entre les arbres.
- **gamma** : Un seuil pour la division d'un nœud basé sur la réduction de la perte. Aide à éviter le surajustement.
- **reg_alpha** : Contrôle la régularisation L1 (norme L1) en ajoutant une pénalité aux poids des nœuds des arbres.
- **reg_lambda** : Contrôle la régularisation L2 (norme L2) en ajoutant une pénalité aux poids des nœuds des arbres.
- **objective** : Type de problème à résoudre (classification, régression, etc.).
- **tree_method** : Spécifie la méthode utilisée pour construire les arbres
- **n_jobs** : Le nombre de threads à utiliser lors de la construction des arbres. Permet d'accélérer le processus sur des systèmes multicœurs.

Sommaire

Résultats







MAE sur l'Ensemble d'Entraînement : **2.58**

MAE sur l'Ensemble de Validation : **2.798**

MAE sur l'Ensemble de Test : **50% 2.7993 et 50% 2.7924**

Observations :

- La performance du modèle, évaluée par le MAE, est cohérente sur l'ensemble de validation et de test.
- Les résultats indiquent une stabilité et une généralisation du modèle sur de nouvelles données.

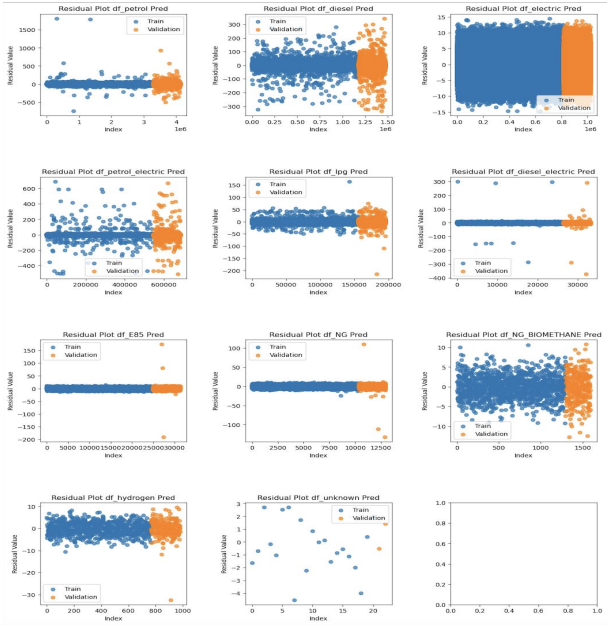
UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

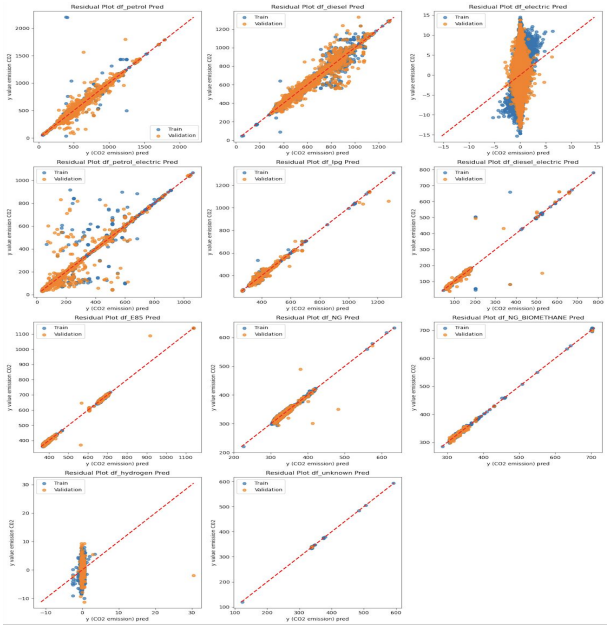
Résultats

Visualisation des Résidus :

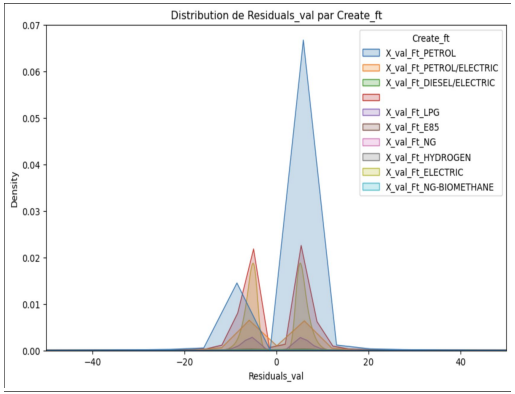
- Entraînement : Analyse des résidus par catégorie de véhicule.
- Validation : Extension de l'analyse pour évaluer la stabilité du modèle.

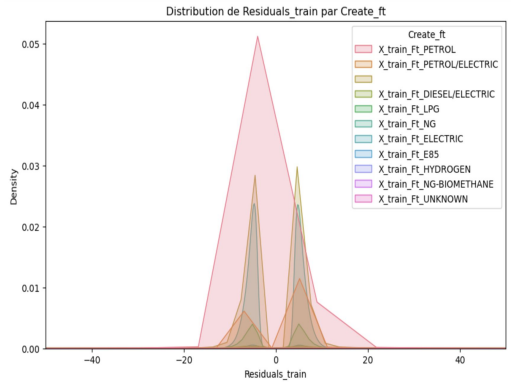
Objectif : Identifier des tendances dans les résidus pour évaluer la performance et la robustesse du modèle.





Résultats





Observations :

- Véhicules à Essence : Risque de surapprentissage.
- Autres Catégories : Distribution Bimodale.

Implications :

- Ajustements nécessaires pour atténuer le surapprentissage chez les véhicules à essence.
- Explorer les caractéristiques influençant la distribution bimodale.

Prochaines étapes :

- Calibration pour les véhicules à essence.
- Exploration approfondie des caractéristiques pour les autres catégories.

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Sommaire

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Perspectives

Perspectives pour une meilleure compréhension et amélioration des résultats d'imputation :

- Explorer l'analyse de la distribution des résidus par modalité pour chaque variable afin de visualiser les caractéristiques des moins bonnes prédictions

- Utilisation de méthodes avancées telles que le stacking, le blending, la régression linéaire, et la régression ridge sur les cibles prédites dans l'ensemble d'entraînement, avec une prédiction ultérieure sur la cible prédite de l'ensemble de validation (nécessitant des ressources).

- Exploitation de méthodes d'optimisation de paramètres telles que la grid search, les méthodes bayésiennes (nécessitant des ressources).

- Application de la cross-validation k-fold pour une évaluation plus robuste des performances (nécessitant également des ressources).

Limites

- Ressources Limitées :
  - Contraintes en termes de puissance informatique et de mémoire.
  - Impact sur le temps d'exécution des opérations.

- Base de Données de 7 500 000 Lignes :
  - Limitations dans la manipulation efficace de grands ensembles de données.
  - Notamment, difficultés dans l'analyse détaillée des données massives.

Merci de votre attention