Anticipez les besoins en consommation de bâtiments

Prédictions des consommations d'énergie et émissions de gaz à effet de serre des bâtiments non-résidentiels de Seattle

Date soutenance : 30/01/2024

Antoine Arragon

Plan :

Problématique

- La ville de Seattle s'est fixée un objectif de ville neutre en émissions de carbone en 2050.
  - ➢ Constitution de relevés minutieux sur la consommation totale d'énergie et les émissions de CO2 des bâtiments non-résidentiels de la ville ;
  - ➢ Problème : ces relevés sont très coûteux ;
  - ➢ Objectifs : Pouvoir se passer de ces relevés en prédisant au mieux les consommations en énergie et les émissions de CO2 de ces bâtiment.

- Notre but est donc de travailler sur le jeu de données fourni par la ville :
  - ➢ Nettoyage, analyse et préparation du jeu de données afin qu'il soit interprétable par des modèles de machine learning ;
  - ➢ Test de différents modèles afin d'obtenir les meilleurs prédictions pour les consommations d'énergie et émissions de gaz à effet de serre ;
  - ➢ Analyse des résultats ;
  - ➢ Intérêt de l'indicateur 'ENERGY STAR Score' pour la prédiction ?

1. Data cleaning et analyse exploratoire

Provenance des données : site de la ville de Seattle - téléchargement du jeu de données mis à disposition

Description rapide du jeu de données :

- 3376 lignes : le nombre total de bâtiments
- 46 colonnes : le nombre de variables, qui sont les caractéristiques de ces bâtiments.

On a des **variables numériques** (à la fois entières et décimales) ainsi que des variables **catégorielles** et une variable **booléenne**.

Dataset plutôt bien renseigné, certaines colonnes ont un taux de valeurs manquantes assez élevées mais elles ne seront pas forcément pertinentes pour notre analyse.

1. Data cleaning et analyse exploratoire

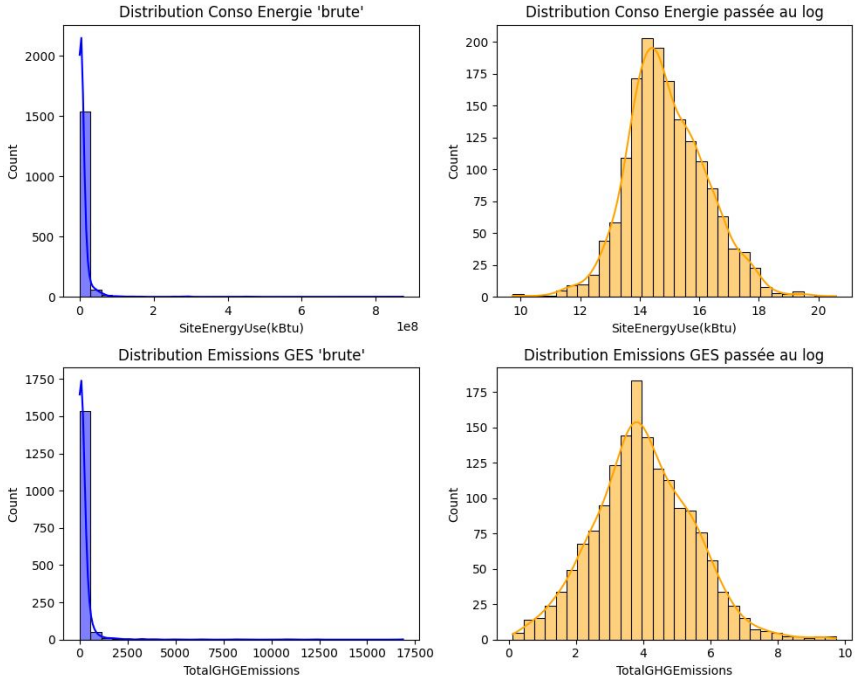
1.1. Datacleaning

1) Suppression des bâtiments résidentiels, non conforme à notre problématique
2) Sélection de nos deux variables cibles :
    a) **‘SiteEnergyUse(kBtu)’** -> consommation totale d’énergie
    b) **‘TotalGHGEmissions’** -> émission totale de gaz à effet de serre (GES)
3) Sélection et/ou création de variables explicatives pertinentes :
    a) **Localisation** : latitude / longitude ;
    b) **Type d’usage** : usage principal , nombre d’usages différents et proportion d’utilisation ;
    c) **Surface** : surface totale (bâtie + parking), %surface bâtie, % surface parking ;
    d) **Proportion des sources d’énergies utilisées** : %Elec, %Gaz, %Steam ;
    e) **Ancienneté du bâtiment** : transformation de l’année de construction ;
    f) **Nombre d’étages**
4) Traitement des données manquantes et des outliers

1. Data cleaning et analyse exploratoire

1.2. Analyse exploratoire

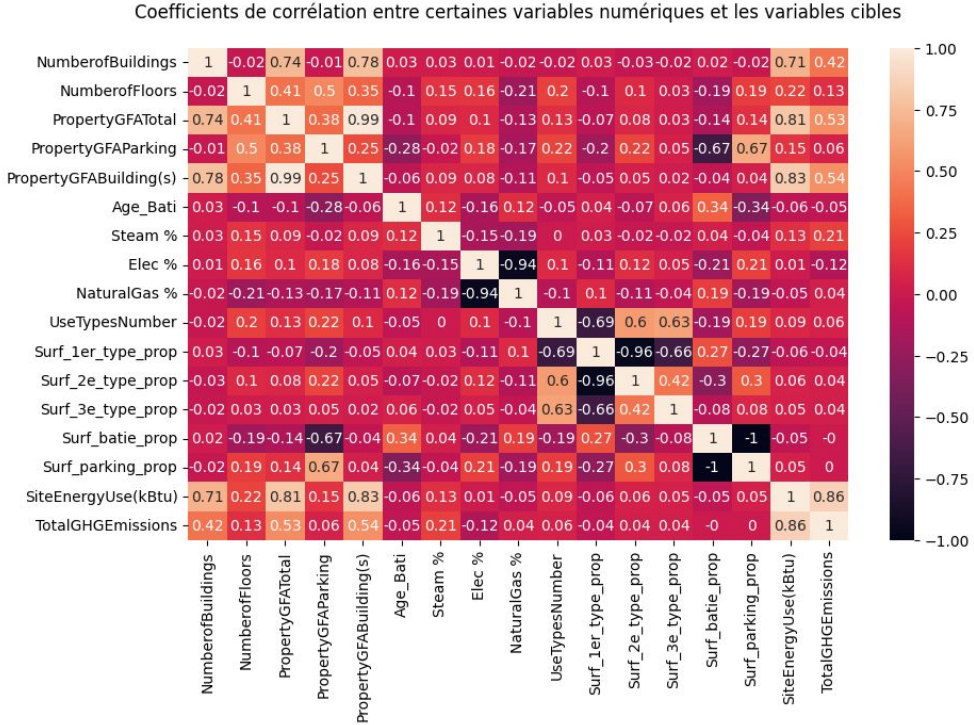
Le passage au log des variables cibles rapproche leur distribution de celle d'une loi normale :



1. Data cleaning et analyse exploratoire

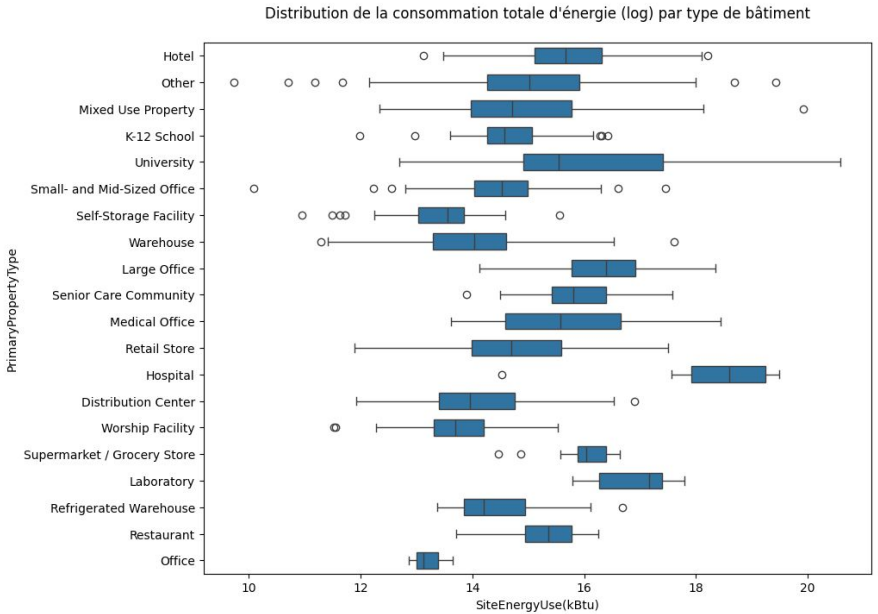
1.2. Analyse exploratoire

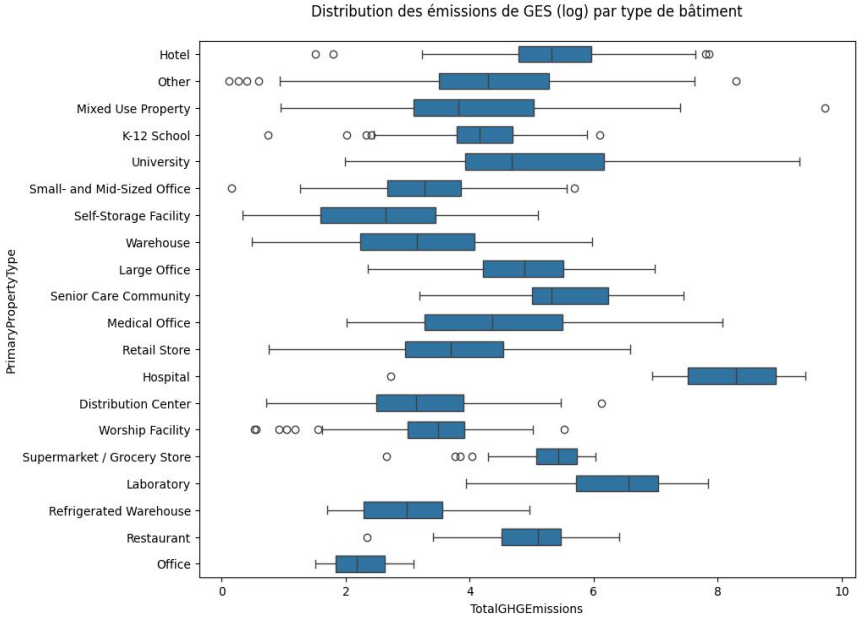
Matrice des corrélations :



1. Data cleaning et analyse exploratoire

1.2. Analyse exploratoire





2. Démarche de sélection et test de différents modèles

2.1. Approche de la modélisation

- Dimension du jeu de données : 1624 lignes et 23 colonnes
- Après plusieurs itérations : 15 variables explicatives : 14 numériques (passage de la surface totale au log) - 1 catégorielle (type d'usage)
- Transformation des données :
  - **a) Passage au log de la variable cible**
  - b) Séparation du jeu de données en **train_set** et **test_set** (test_size : 20%)
  - c) Application d'une normalisation **StandardScaler** aux variable numériques et d'un **OneHotEncoder** à la variable catégorielle
    - ➢ fit_transform sur X_train et .transform sur X_test
- Définition de fonctions :
  - **a. Cross Validation**
  - b. Affichage - Enregistrement des résultats
  - **c. GridSearchCV**
  - **d. Finetuning**
- Métriques d'évaluation : **MAE, RMSE, R2**, Temps d'entraînement

2. Démarche de sélection et test de différents modèles

2.1. Consommation d'énergie

2.1.1. Résultats

Entrainement de modèles avec **hyperparamètres** par défaut :

	Model	MAE	RMSE	R2_test	R2_train	R2_cv	Duration
0	Dummy Regression	1.0257	1.2809	-0.0003	0.0000	-0.0036	0.0063
1	LinearRegression	0.4926	0.6682	0.7278	0.7494	0.7369	0.0156
2	Ridge	0.4956	0.6728	0.7240	0.7505	0.7393	0.0152
3	Lasso	1.0056	1.2570	0.0367	0.0358	0.0318	0.0074
4	ElasticNet	0.8130	1.0358	0.3459	0.3385	0.3383	0.0095
5	SVR	0.9234	1.1698	0.1657	0.1642	0.1488	0.4844
6	RandomForest	0.5296	0.7100	0.6926	0.9592	0.7118	2.3396
7	GradientBoost	0.5023	0.6722	0.7246	0.8237	0.7324	0.4249
8	XGB	0.5497	0.7386	0.6674	0.9944	0.6940	2.8612

2. Démarche de sélection et test de différents modèles

2.1. Consommation d'énergie

2.1.1. Résultats

Recherche des meilleurs **hyperparamètres** via **GridSearchCV** :

	Model	Params	MAE	RMSE	R2_test	R2_train	R2_cv	Duration
0	LinearRegression	{'fit_intercept': False}	0.4939	0.6712	0.7253	0.7511	0.7382	0.0029
1	Ridge	{'alpha': 1, 'fit_intercept': True}	0.4956	0.6728	0.7240	0.7505	0.7393	0.0019
2	Lasso	{'alpha': 0.01, 'fit_intercept': True}	0.5246	0.7119	0.6910	0.7230	0.7167	0.0020
3	ElasticNet	{'alpha': 0.01, 'l1_ratio': 0.1, 'tol': 0.01}	0.5110	0.6931	0.7071	0.7387	0.7297	0.0075
4	SVR	{'C': 10, 'epsilon': 0.3, 'gamma': 0.01}	0.4802	0.6647	0.7306	0.7773	0.7380	0.0780
5	RandomForest	{'max_depth': 20, 'max_features': None, 'n_estimators': 500}	0.5299	0.7081	0.6943	0.9597	0.7141	7.0759
6	GradientBoost	{'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100}	0.5023	0.6722	0.7246	0.8237	0.7324	0.4144
7	XGB	{'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 0.5, 'n_estimators': 200}	0.4915	0.6653	0.7302	0.8556	0.7357	0.1506

Ici, le R2_cv correspond au meilleur score obtenu dans la cross_validation effectuée via la GridSearchCV

2. Démarche de sélection et test de différents modèles

2.1. Consommation d'énergie

2.1.1. Résultats

Finetuning et résultats finaux :

	Model	Params	MAE	RMSE	R2_test	R2_train	R2_cv	Duration
0	Ridge	{'alpha': 1, 'fit_intercept': True}	0.4956	0.6728	0.7240	0.7505	0.7387	0.0030
1	Lasso	{'alpha': 0.006, 'fit_intercept': True}	0.5107	0.6929	0.7073	0.7355	0.7258	0.0074
2	ElasticNet	{'alpha': 0.001, 'l1_ratio': 0.1, 'tol': 0.01}	0.4960	0.6733	0.7237	0.7503	0.7388	0.0292
3	SVR	{'C': 45, 'epsilon': 0.3, 'gamma': 0.0045}	0.4707	0.6559	0.7377	0.7782	0.7411	0.1232
4	RandomForest	{'max_depth': 18, 'min_samples_split': 2, 'n_estimators': 500}	0.5292	0.7072	0.6951	0.9581	0.7089	7.6783
5	GradientBoost	{'learning_rate': 0.1, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 120}	0.4989	0.6690	0.7271	0.8344	0.7333	0.5693
6	XGB	{'colsample_bytree': 0.8, 'gamma': 0.2, 'learning_rate': 0.03, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 550, 'reg_lambda': 1.1, 'subsample': 0.7}	0.4841	0.6542	0.7391	0.8502	0.7414	0.3673

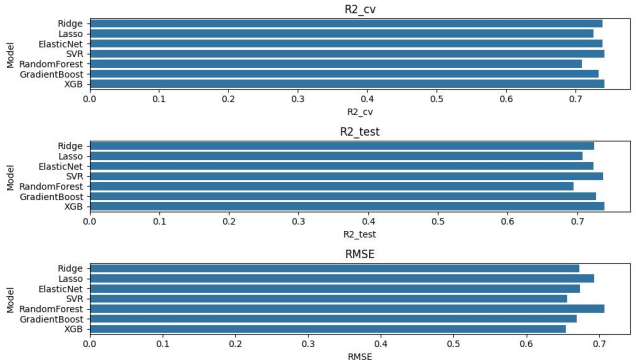
Ici, le R2_cv correspond au meilleur score obtenu dans la cross_validation effectuée via la GridSearchCV

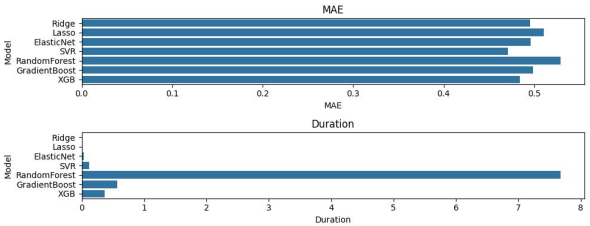
2. Démarche de sélection et test de différents modèles

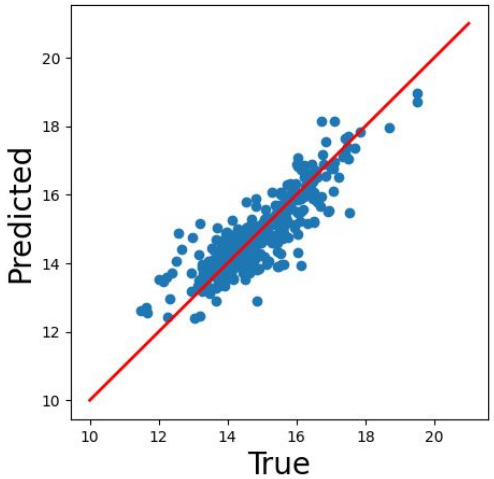
2.1. Consommation d'énergie

2.1.1. Résultats

Visualisation des résultats finaux :







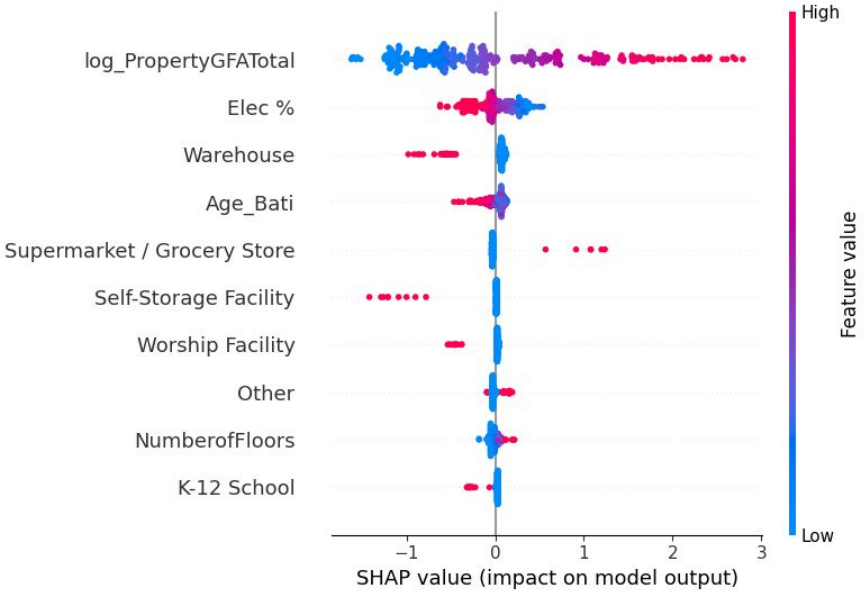
Choix du modèle : **XGBRegressor** {'colsample_bytree': 0.8, 'gamma': 0.2, 'learning_rate': 0.03, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 550, 'reg_lambda': 1.1, 'subsample': 0.7}

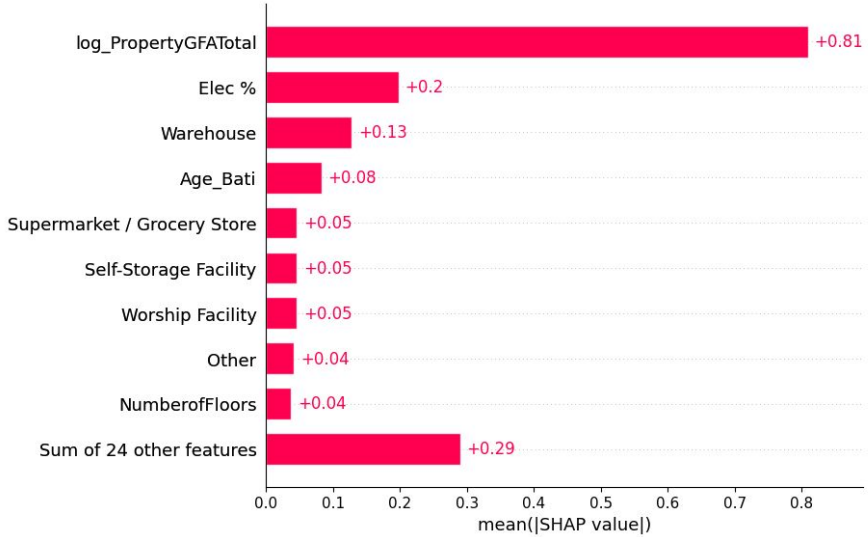
2. Démarche de sélection et test de différents modèles

2.1. Consommation d'énergie

2.1.2. Feature importance

Globale :



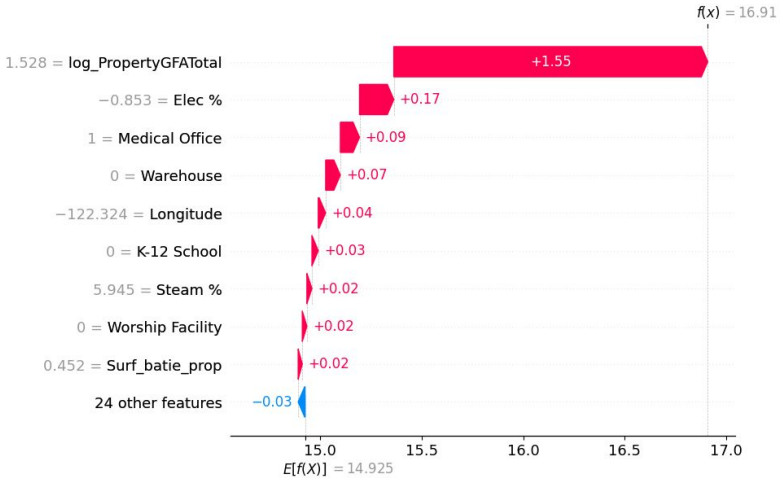


2. Démarche de sélection et test de différents modèles

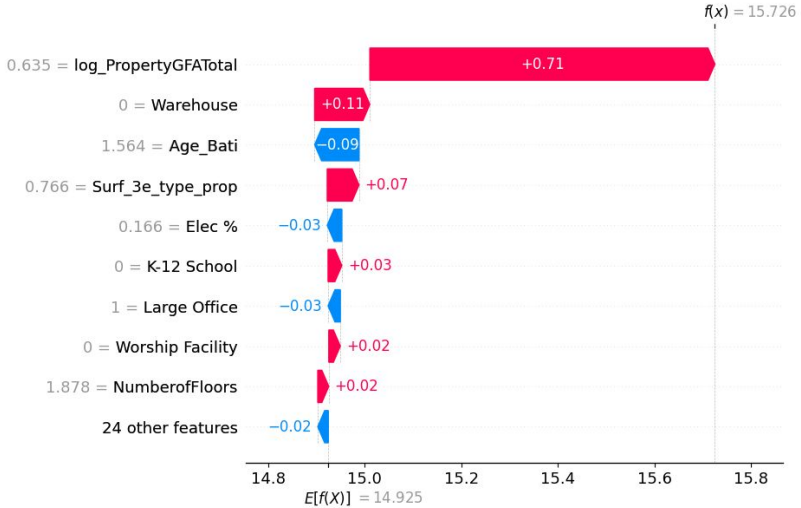
2.1. Consommation d'énergie

2.1.2. Feature importance

Locale :



Individu X_test[50]



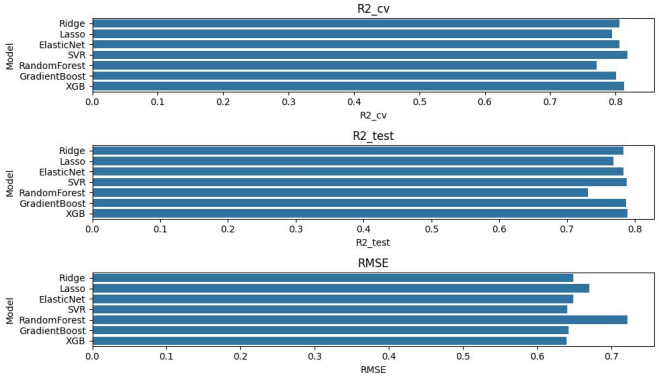
Individu aléatoire

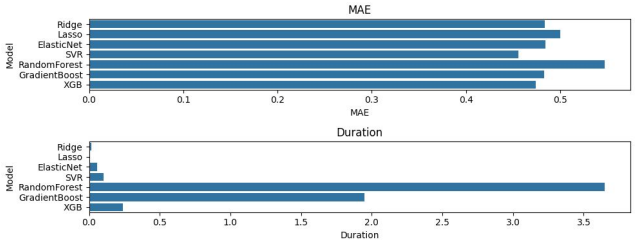
2. Démarche de sélection et test de différents modèles

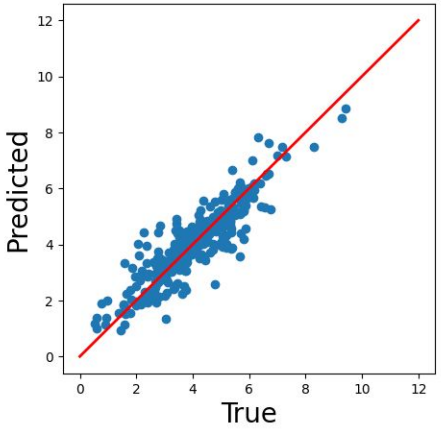
2.2. Émissions de GES

2.2.1. Résultats

Visualisation des résultats finaux :







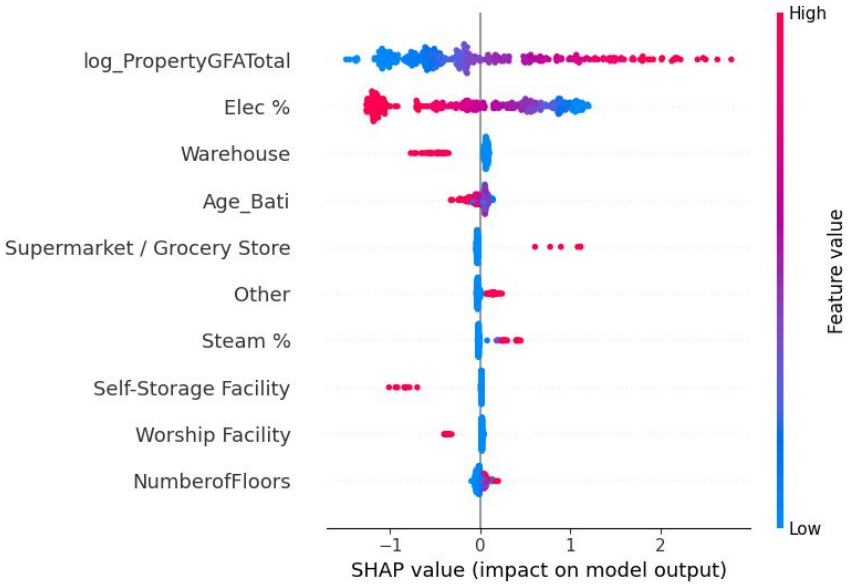
Choix du modèle : **XGBRegressor** {'gamma': 0.3, 'learning_rate': 0.03, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 500, 'subsample': 0.7}
Scores : MAE : 0.4740 / RMSE : 0.6394 / R2 : 0.7891 (R2_cv : 0.8131) / Duration : 0.2403.

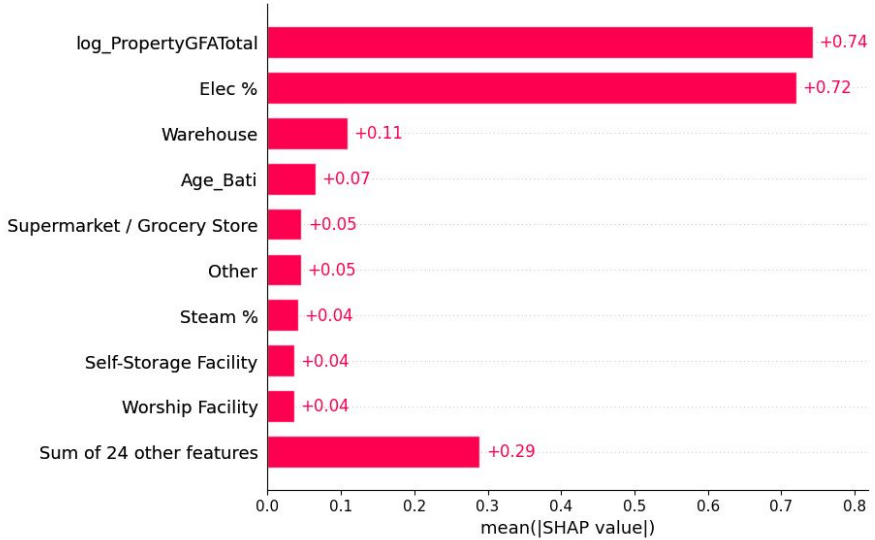
2. Démarche de sélection et test de différents modèles

2.2. Émissions de GES

2.2.2. Feature importance

Globale :



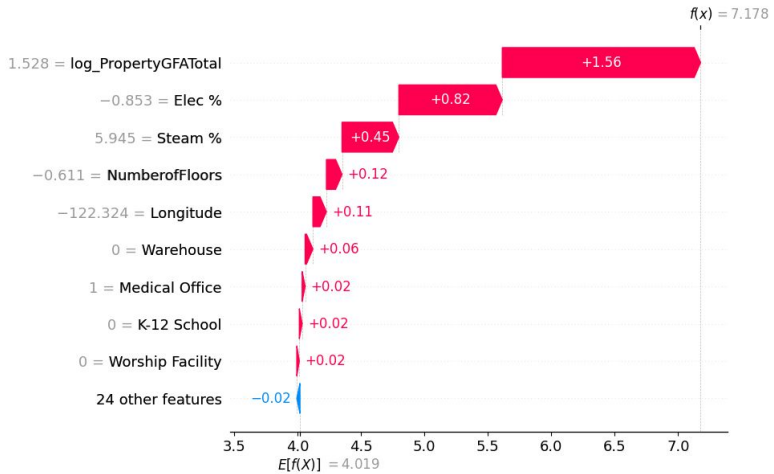


2. Démarche de sélection et test de différents modèles

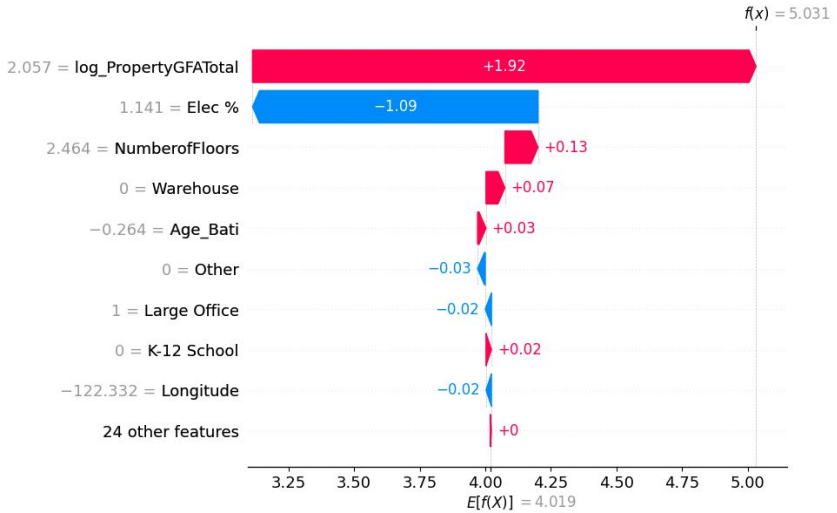
2.2. Émissions de GES

2.2.2. Feature importance

Locale :



Individu X_test[50]

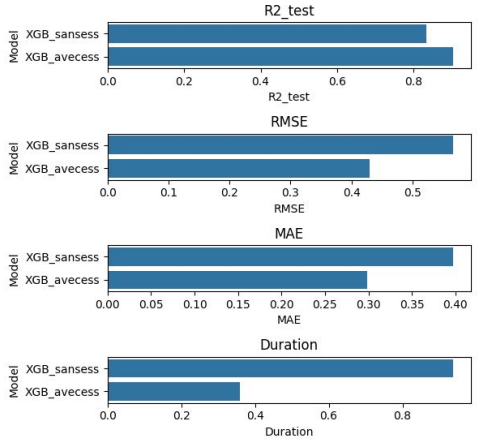


Individu aléatoire

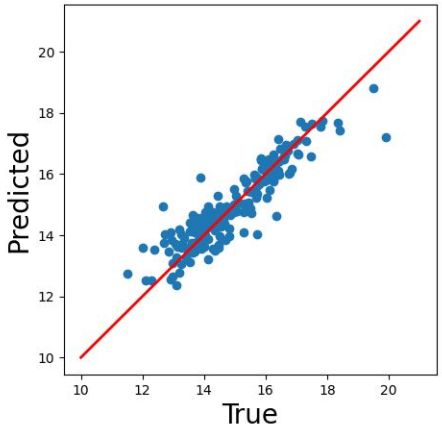
3. Impact de l'intégration de l'Energy Star Score

3.1. Consommation d'énergie

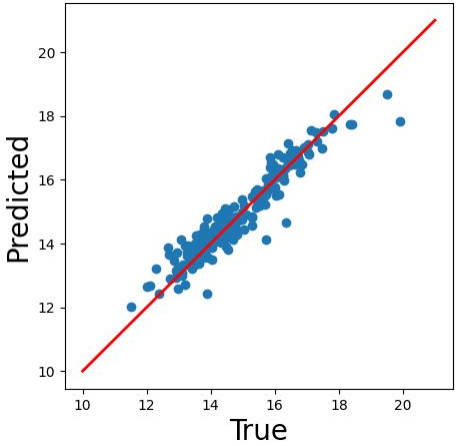
3.1.1. Résultats



Sans :



Avec :



	Model	MAE	RMSE	R2_test	R2_train	R2_cv	Duration
0	XGB_sanssess	0.3978	0.5671	0.8328	0.9037	0.7852	0.9384
1	XGB_avecess	0.2990	0.4296	0.9041	0.9544	0.8831	0.3595

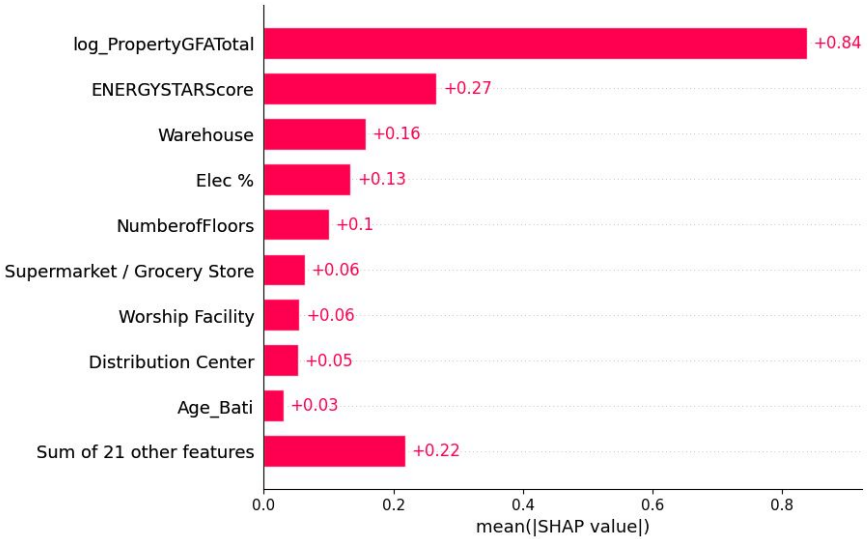
Dataleakage avec Energy Star Score ?

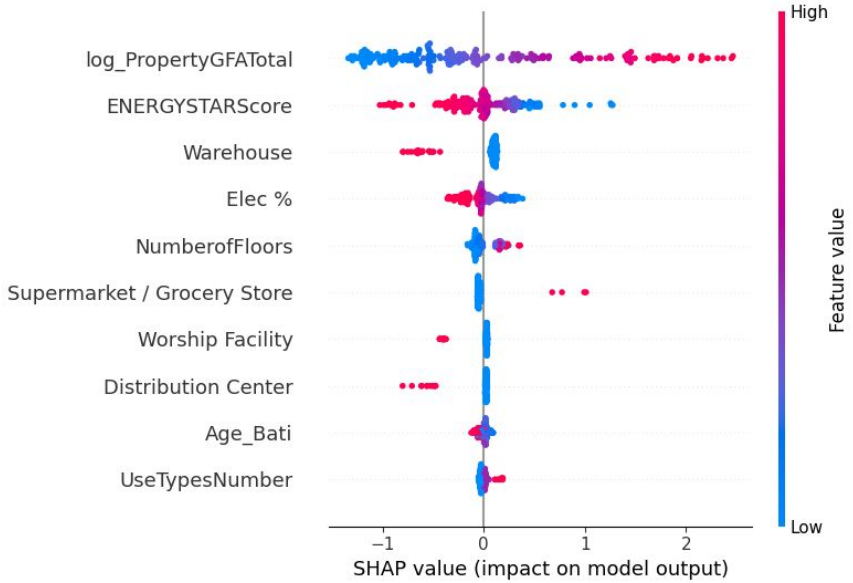
3. Impact de l'intégration de l'Energy Star Score

3.1. Consommation d'énergie

3.1.2. Feature importance

Globale :



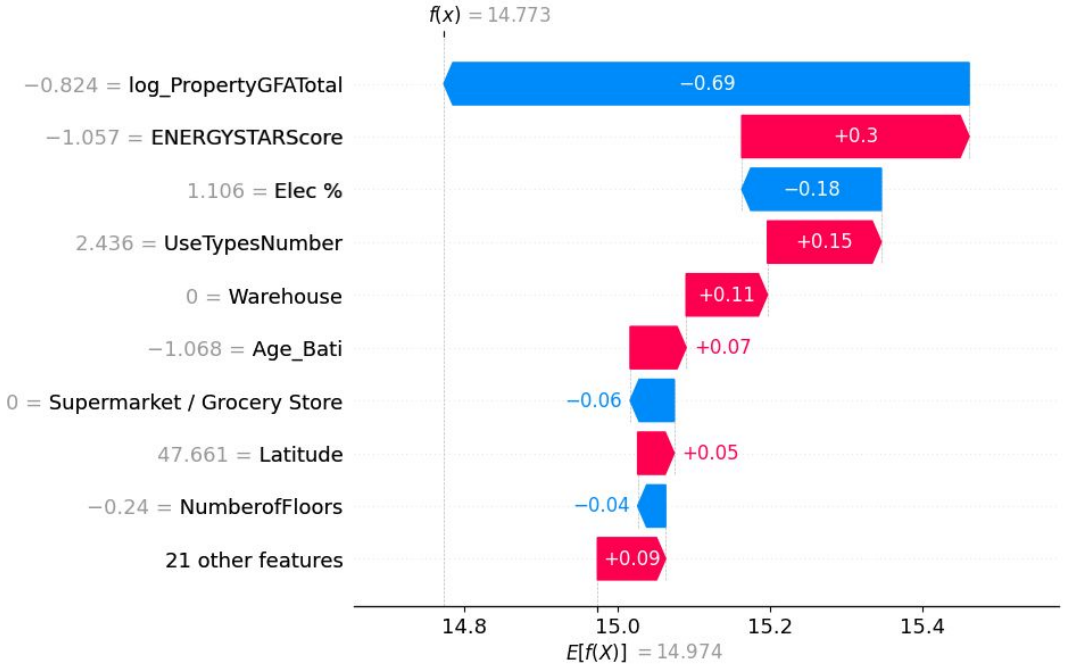


3. Impact de l'intégration de l'Energy Star Score

3.1. Consommation d'énergie

3.1.2. Feature importance

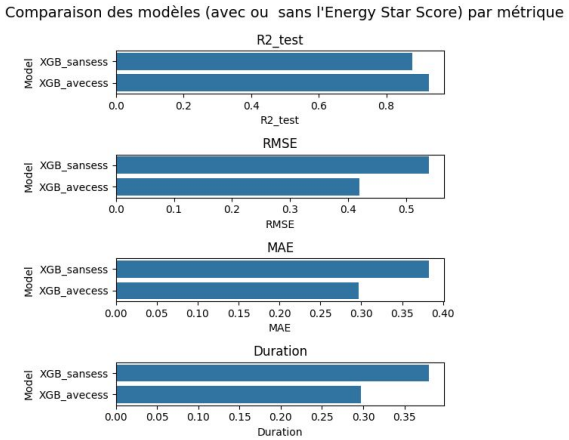
Locale :



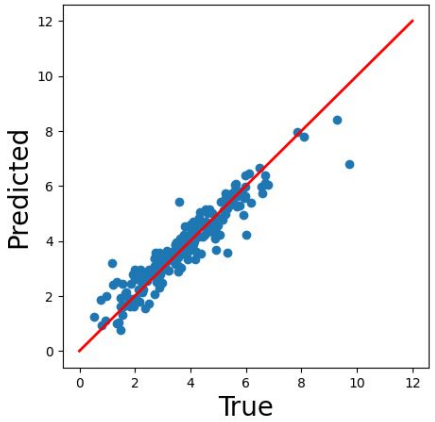
3. Impact de l'intégration de l'Energy Star Score

3.2. Émissions de GES

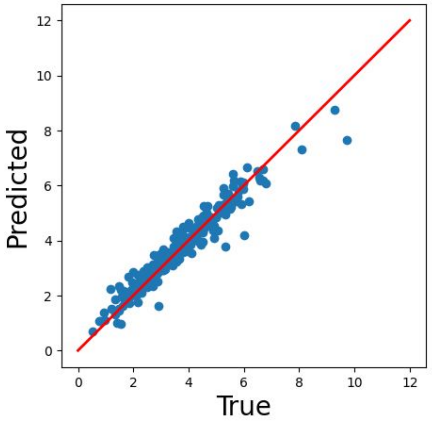
3.2.1. Résultats



Sans :



Avec :



	Model	MAE	RMSE	R2_test	R2_train	R2_cv	Duration
0	XGB_sanseSS	0.3825	0.5390	0.8770	0.9236	0.8298	0.3795
1	XGB_avecess	0.2966	0.4195	0.9255	0.9601	0.9004	0.2972

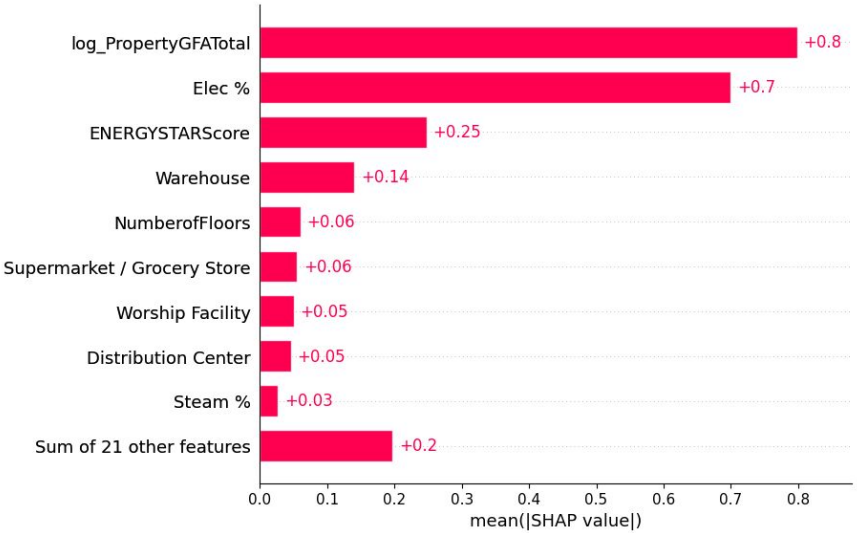
Dataleakage avec Energy Star Score ?

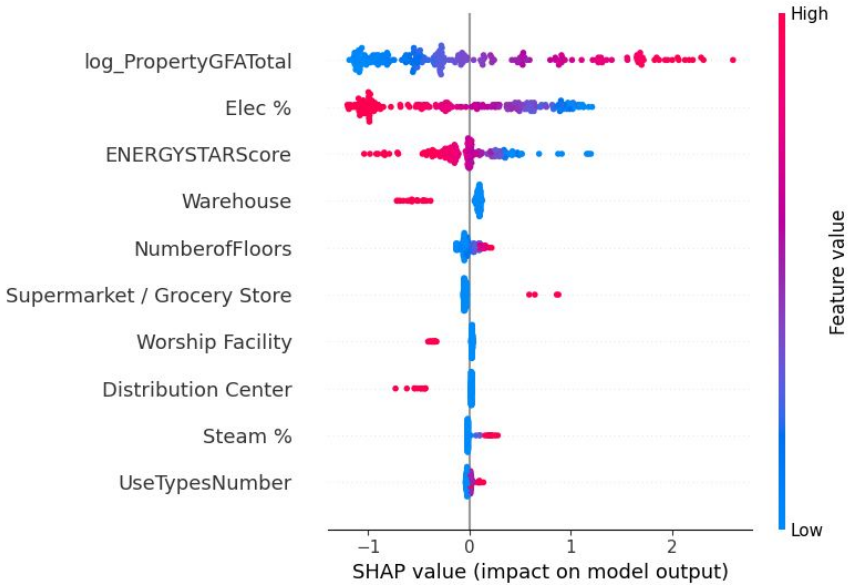
3. Impact de l'intégration de l'Energy Star Score

3.2. Émissions de GES

3.2.2. Feature importance

Globale :



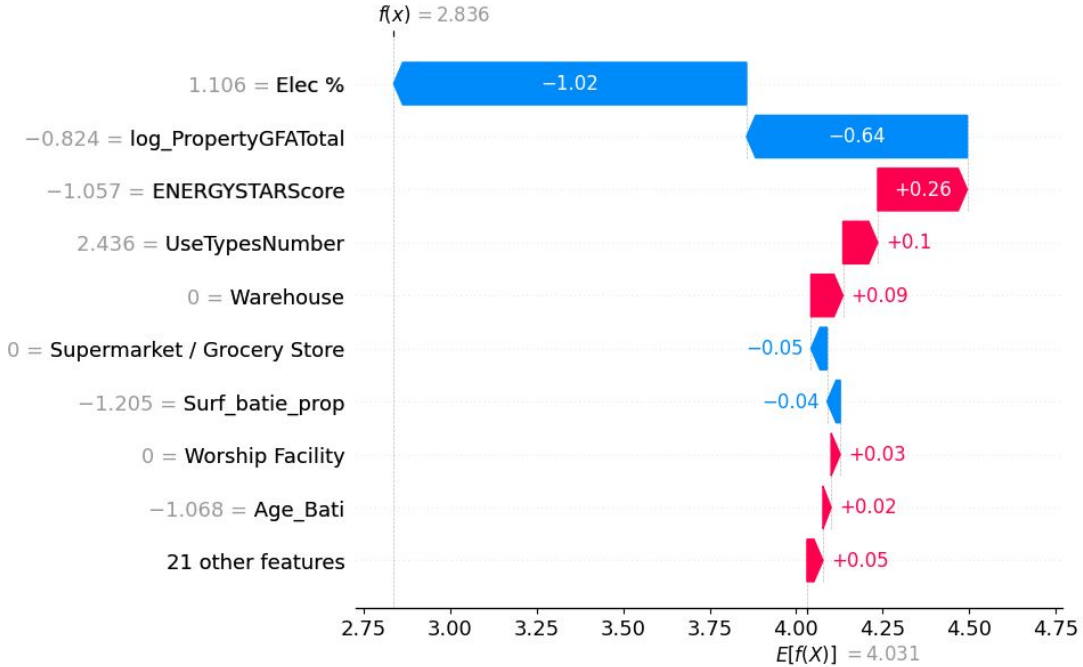


3. Impact de l'intégration de l'Energy Star Score

3.2. Émissions de GES

3.2.2. Feature importance

Locale :



Conclusion

Nous avons montré dans ce travail la possibilité de **prédire**, avec une marge d'erreur raisonnable, les consommations énergétiques et les émissions de gaz à effet de serres de bâtiments de la ville de Seattle, en fonction de certaines de leurs caractéristiques.

Il a été nécessaire pour cela de **retravailler le jeu de données** mis à disposition, de manière à l'**optimiser** en vue de l'entraînement de différents **modèles de Machine Learning**.

Les **résultats** obtenus par nos différents modèles, après **plusieurs itérations** et **ajustements des hyperparamètres** nous semblent satisfaisants. On a pu identifier des variables ayant un **poids** important dans la **détermination de la prédiction** des variables cibles, telles que la **surface, le type d'énergie utilisé et certains types d'usage des bâtiments**.

L'inclusion de l'**Energy Star Score** a permis d'améliorer sensiblement les résultats. Cette variable ressort également dans les analyses de feature importance avec un poids non négligeable. Mais il faut prendre ces résultats avec prudence, cela étant potentiellement dû à du **dataleakage**, occasionnant des **prédictions particulièrement optimistes**.

Librairies utilisées

Python - Version 3.10.12

Pandas - Version 1.5.3

Numpy - Version 1.23.5

Matplotlib - Version 3.7.1

Seaborn - Version 0.13.1

Scikit-learn - Version 1.2.2

XGBoost - Version: 2.0.3

SHAP - Version: 0.44.0

Ressources

Provenance des données :

https://data.seattle.gov/dataset/2016-Building-Energy-Benchmarking/2bpz-gwpy/about_data

(téléchargement direct via le site d'OC)

Informations sur l'Energy Star Score :
https://www.energystar.gov/buildings/facility-owners-and-managers/existing-buildings/use-portfolio-manager/interpret-your-results/what