Anticipez les besoins en consommation de bâtiments

Appel à projet de Kaggle

Joyce Kuoh Moukouri,
P4, Soutenance du 31/03/2023

Ordre du jour

Anticipez les besoins en consommation de bâtiments

1. La mission
2. Présentation du jeu de données
3. Feature engineering
4. Approche de modélisation
5. Résultats

Conclusion

1. La mission

La mission

Rappel des objectifs fixés par Douglas

- Prédire **les émissions de CO2** et **la consommation totale d'énergie des bâtiments non destinés** à l'habitation de la ville de Seattle

- Analyser l'incidence de l'ENERGYSTARScore sur la modélisation

2. La base de données

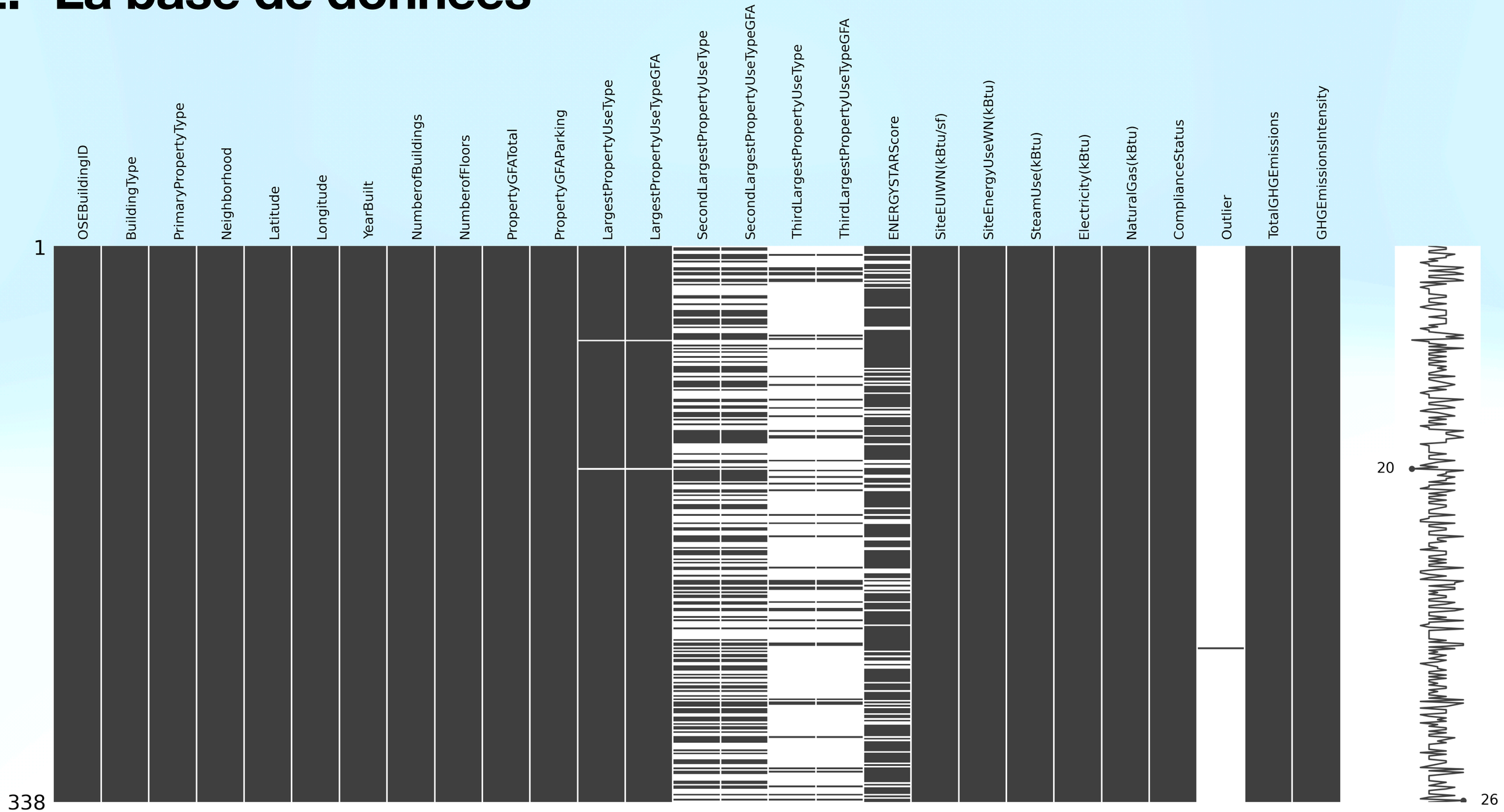
2. La base de données

- Base de données SEATTLE OPEN DATA
- Le rapport du Building Energy Benchmark de 2016, en vue de l'objectif NET ZERO de Seattle d'ici 2050
- Données accessibles en Open Data et gérer par la ville de Seattle
- Clé : Identification des immeubles

Qualité du jeu de données : très bonne

- Valeurs manquantes : seul le score ENERGYSTARScore comporte 36% de valeurs manquantes
- Les autres valeurs manquantes sont « expliqués » ou monotones
- Absence de doublons et pas de valeurs aberrantes dans les individus 'COMPLIANT'

2. La base de données



2. La base de données

Traitement des valeurs manquantes, ENERGYSTARScore

Features	Type	Méthode
PropertyUseType (Fonction secondaire ou tertiaire de l'immeuble)	MNAR monotones	Remplacement par le mention 'NON APPLICABLE'
PropertyUseTypeGFA (Aire attribué à la fonction secondaire ou tertiaire de l'immeuble)	MNAR monotones	Remplacement par 0
ENERGYSTARScore	MAR	KNNImputer avec k = 2, étude de la distribution pour choisir k

3. Feature Engineering

3. Feature Engineering

Trois types de feature engineering utilisés

TYPE 1	Encodage des variables catégorielles	
TYPE 2	Transformation des variables	Création de nouvelles variables
TYPE 3	Scaling des features	

3. Feature Engineering

TYPE 1

Encodage des variables catégorielles

Trois types d'encodage ont été testés

Set 1	Nominal/ Ordinal	Encoding
TYPE	Nominal	One-Hot encoding
FONCTION_1	Nominal	One-Hot encoding
QUARTIER	Nominal	One-Hot encoding

Set 2	Nominal/ Ordinal	Encoding
TYPE	Nominal	One-Hot encoding
FONCTION_1	Nominal	Target Encoding
FONCTION_2	Nominal	Target Encoding
FONCTION_3	Nominal	Target Encoding
QUARTIER	Nominal	One-Hot encoding

Set 3	Nominal/ Ordinal	Encoding
TYPE	Nominal	One-Hot encoding
FONCTION_1	Nominal	One-Hot encoding x part surfacique des usages
QUARTIER	Nominal	One-Hot encoding

3. Feature Engineering

TYPE 2 | Transformation des variables | Création de nouvelles variables

Variables crées	Description	Variables utilisées
PARKING	La surface du parking sur la surface totale	• Surface parking • Surface totale
Energy_part	La part de chaque type d'énergie consommée	• Variables de relevé

3. Feature Engineering

TYPE 3

Scaling des features

- Normalisation des variables avec MinMaxScaler() pour les modèles basés sur les distances entre individus

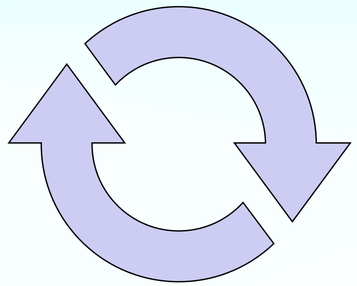
4. Approche de modélisation

4. Approche de modélisation

PHASE 1 — Baseline

- Modélisation Baseline / DummyRegressor()
- Validation croisée sur k = 15 folds
- Score R2 et MAE moyen

Pour chaque modèle



PHASE 2 — Pipeline

- Scaling
  Target_transforming
- Choix du modèle de régression
- GridSearchCV, recherche des hyperparamètres optimaux
  +
  Validation croisée sur k = 15 folds

PHASE 3 — Évaluation

- Score R2 et MAE moyen
- Comparaison à la Baseline

PHASE 4

- Choix du meilleur modèle

4. Approche de modélisation

Les modèles testés
LinearRegression()
Ridge()
Lasso()
ElasticNet()
LinearSVR()
DecisionTreeRegressor()
RandomForest
XGBoost

5. Résultats

5. Résultats

Prédiction de la consommation d'énergie surfacique - set 1

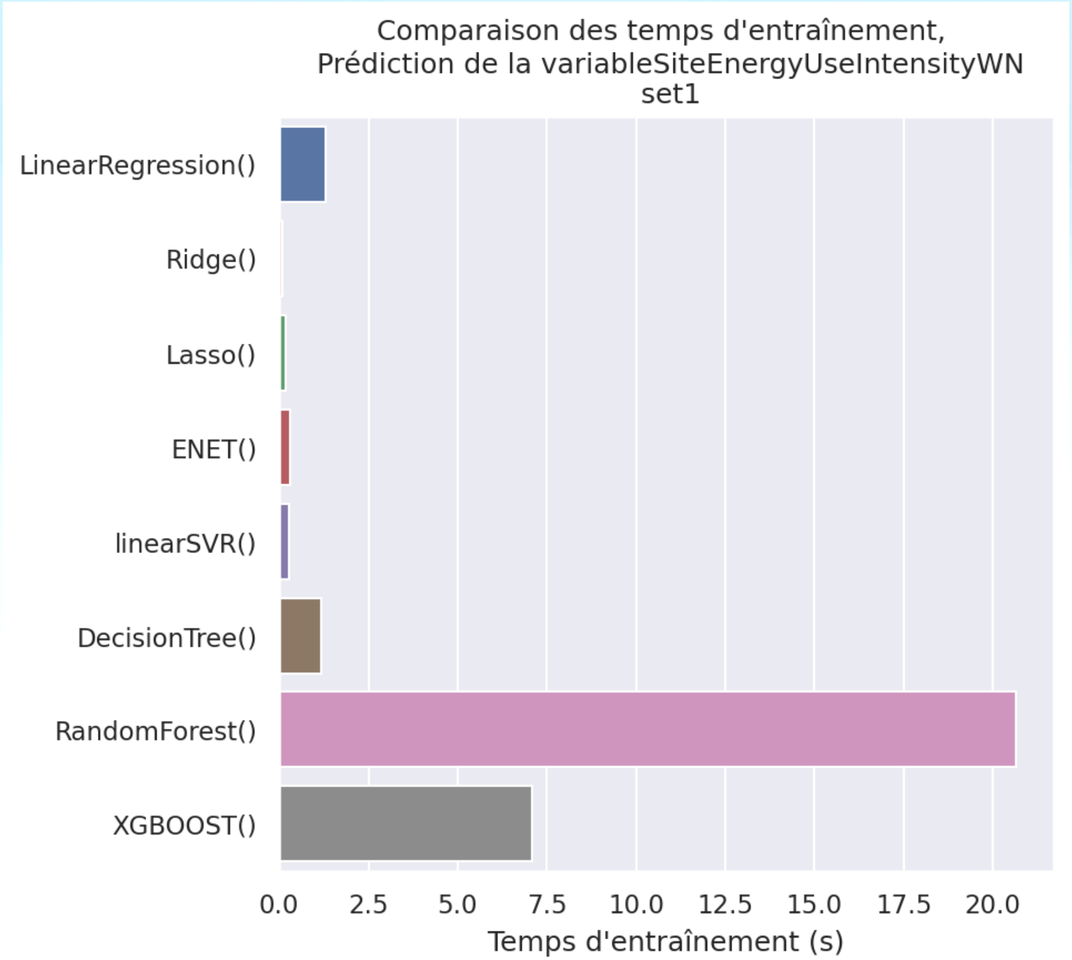
Modèle	best_param	Temps d'entraînement (en s)	MAE moyen Validation croisée	R2 moyen Validation croisée	RMSE Validation croisée
LinearRegression()		1,29	-32,90	0,44	52,28
Ridge()	{'alpha': 0,1}	0,09	-32,66	0,44	52,11
Lasso()	{'alpha': 0,1}	0,19	-32,01	0,45	52,02
ENET()	{'alpha': 0,1, 'l1_ratio': 1,0, 'max_iter': 10}	0,31	-31,94	0,45	52,01
LinearSVR()	{'C': 1000,0}	0,26	-29,71	0,40	54,65
DecisionTree()	{'criterion': 'squared_error', 'max_depth': 3}	1,18	-38,06	0,27	60,02
RandomForest()	{'max_depth': 28, 'n_estimators': 100}	20,67	-29,28	0,50	49,19
XGBOOST()	{'learning_rate': 0,1, 'max_depth': 10, 'n_estimators': 50, 'subsample': 0,8}	7,09	-26,86	0,58	45,39

Modélisation de la variable SiteEUIWN(kBtu/sq) - set 1
Comparaison des modèles testés

5. Résultats

Prédiction de la consommation d’énergie surfacique - set 1

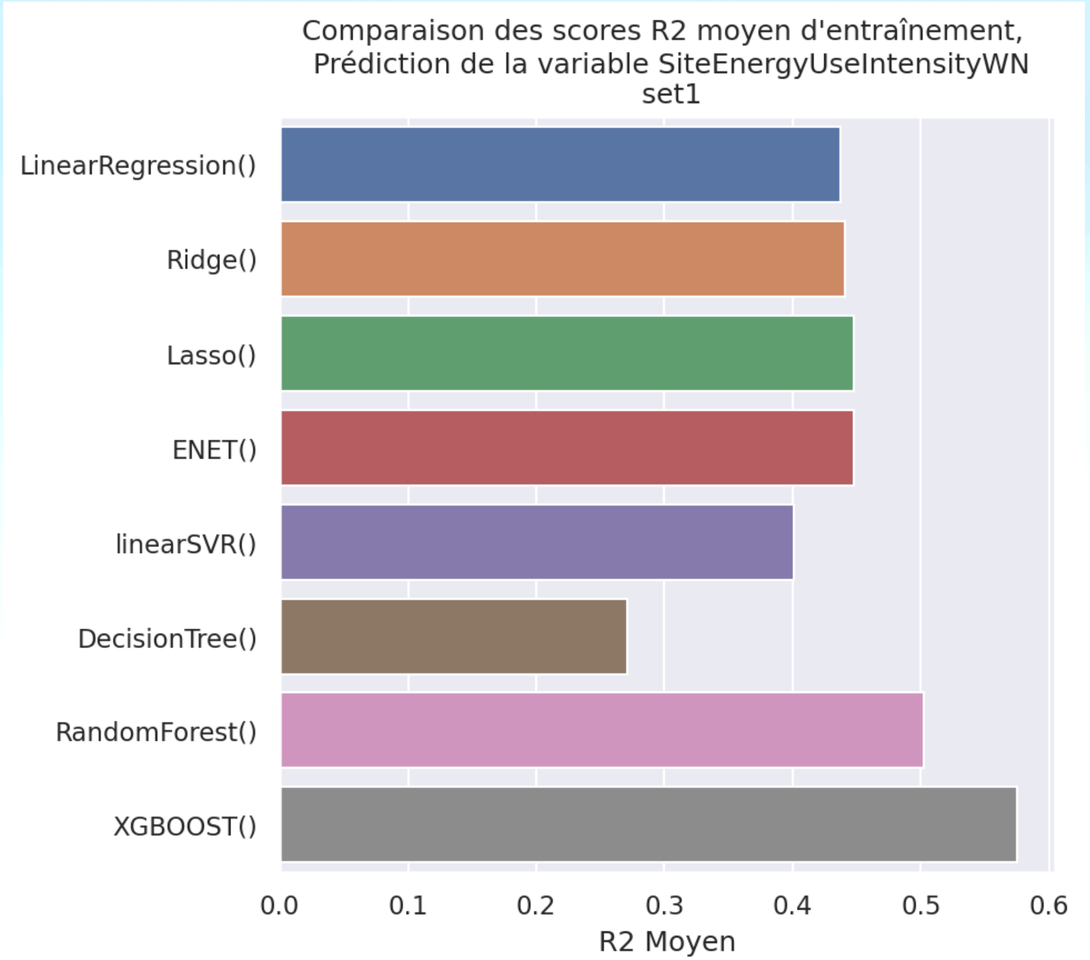
Comparaison des performances de chaque modèles



5. Résultats

Prédiction de la consommation d’énergie surfacique - set 1

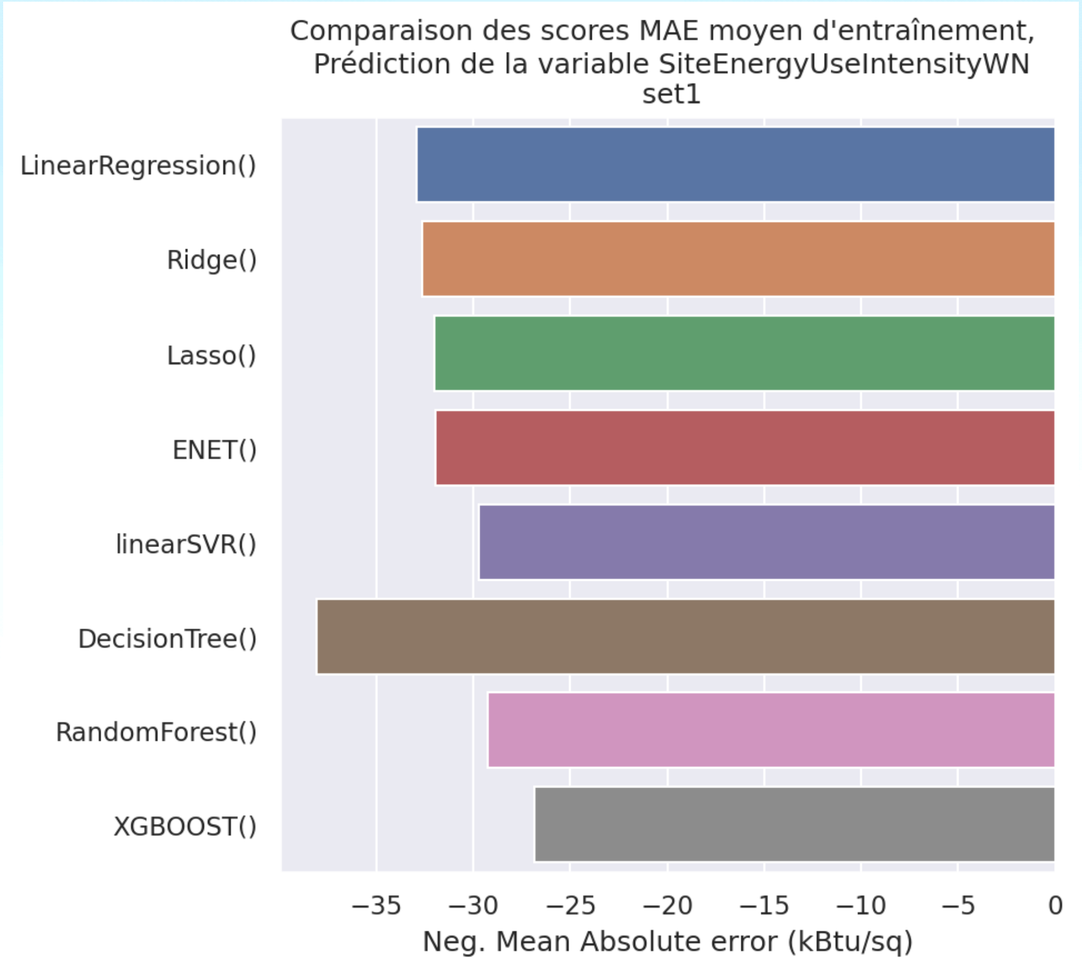
Comparaison des performances de chaque modèles



5. Résultats

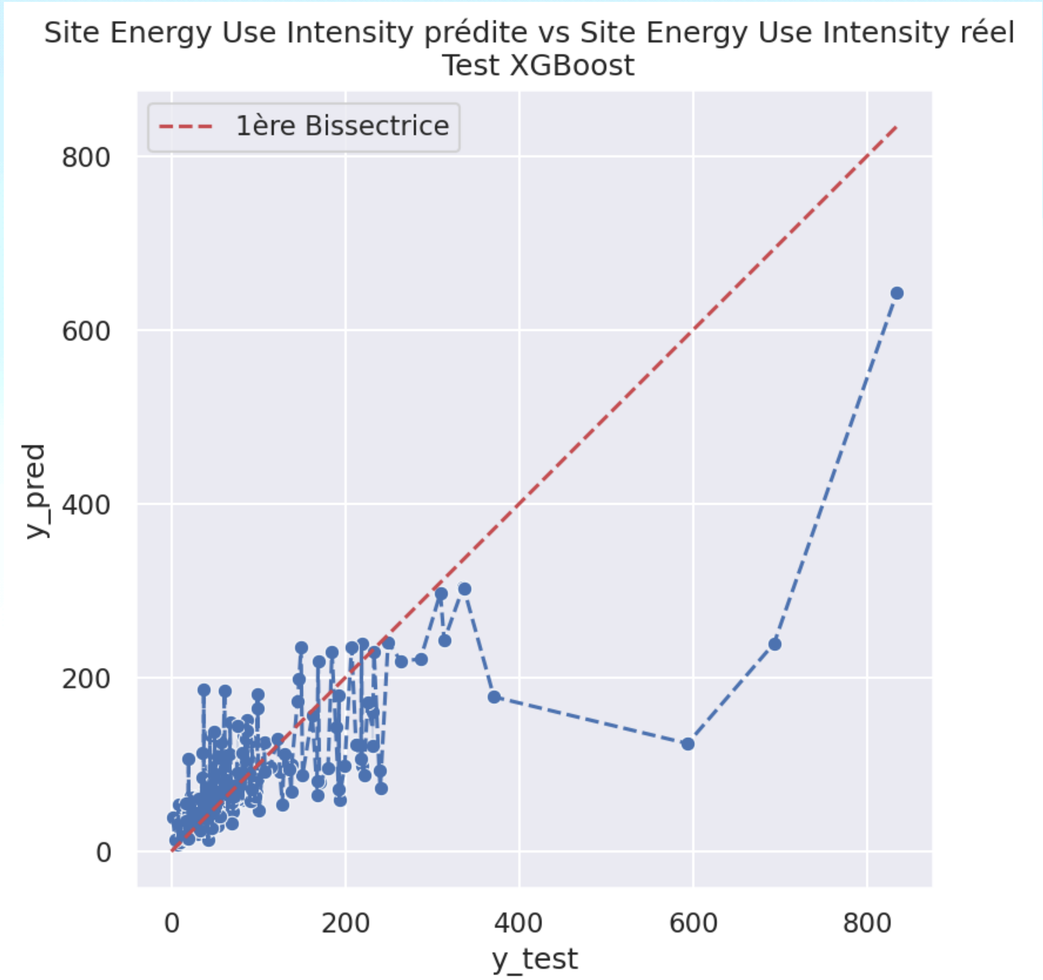
Prédiction de la consommation d’énergie surfacique - set 1

Comparaison des performances de chaque modèles



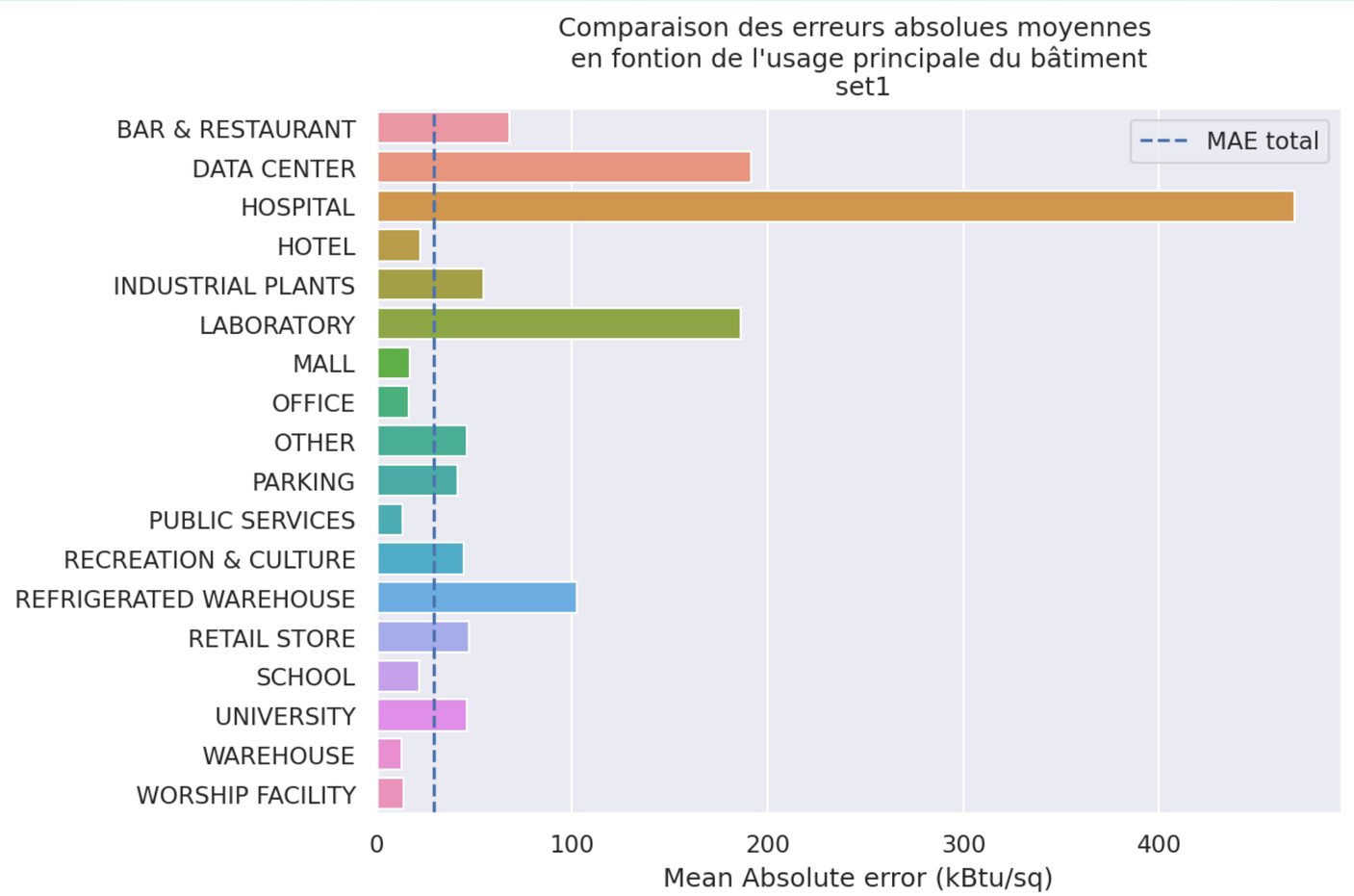
5. Résultats du modèle le plus performant | XGBOOST
Prédiction de la consommation d'énergie surfacique

Modèle XGBOOST



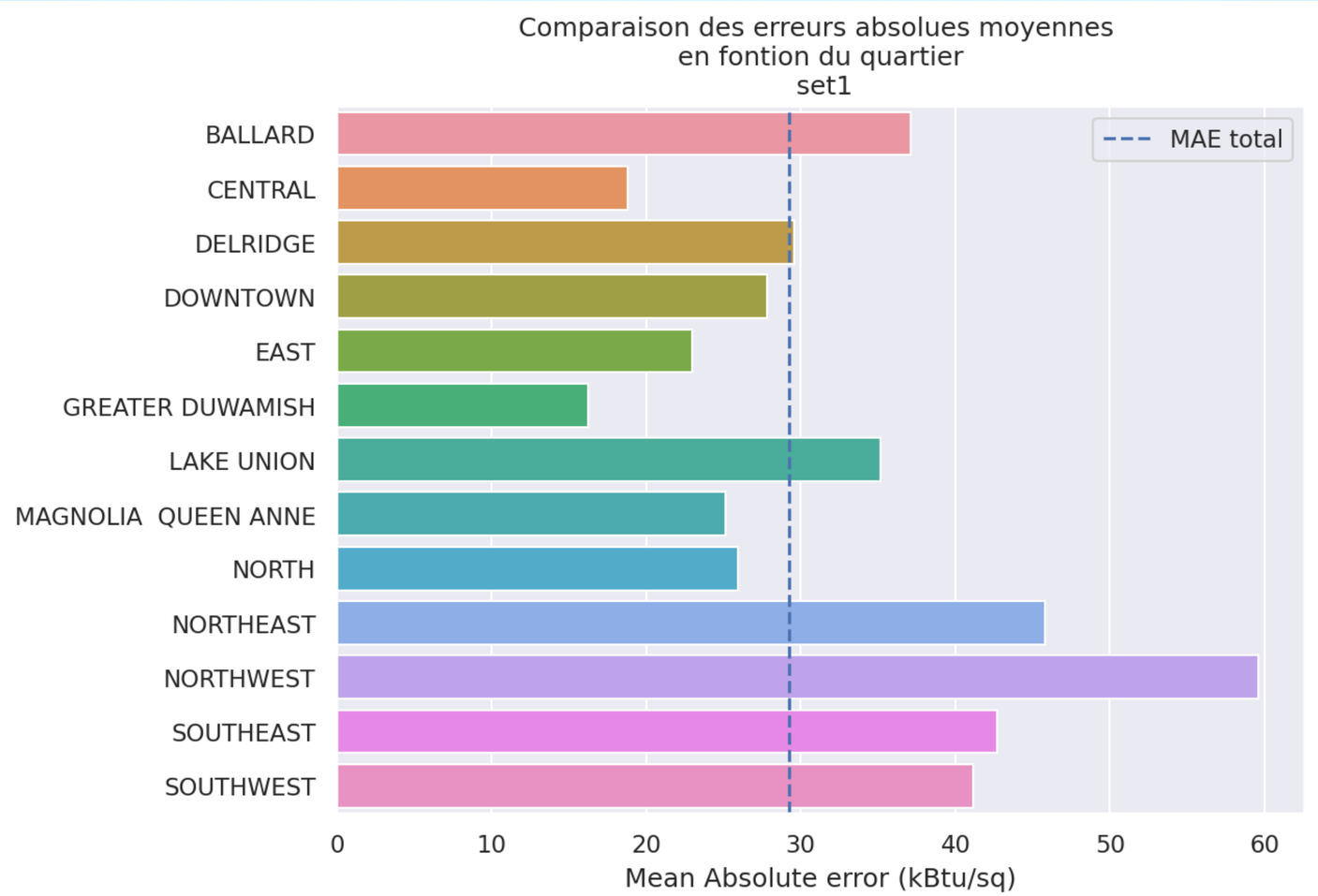
5. Résultats du modèle le plus performant | XGBOOST
Prédiction de la consommation d'énergie surfacique

Modèle XGBOOST



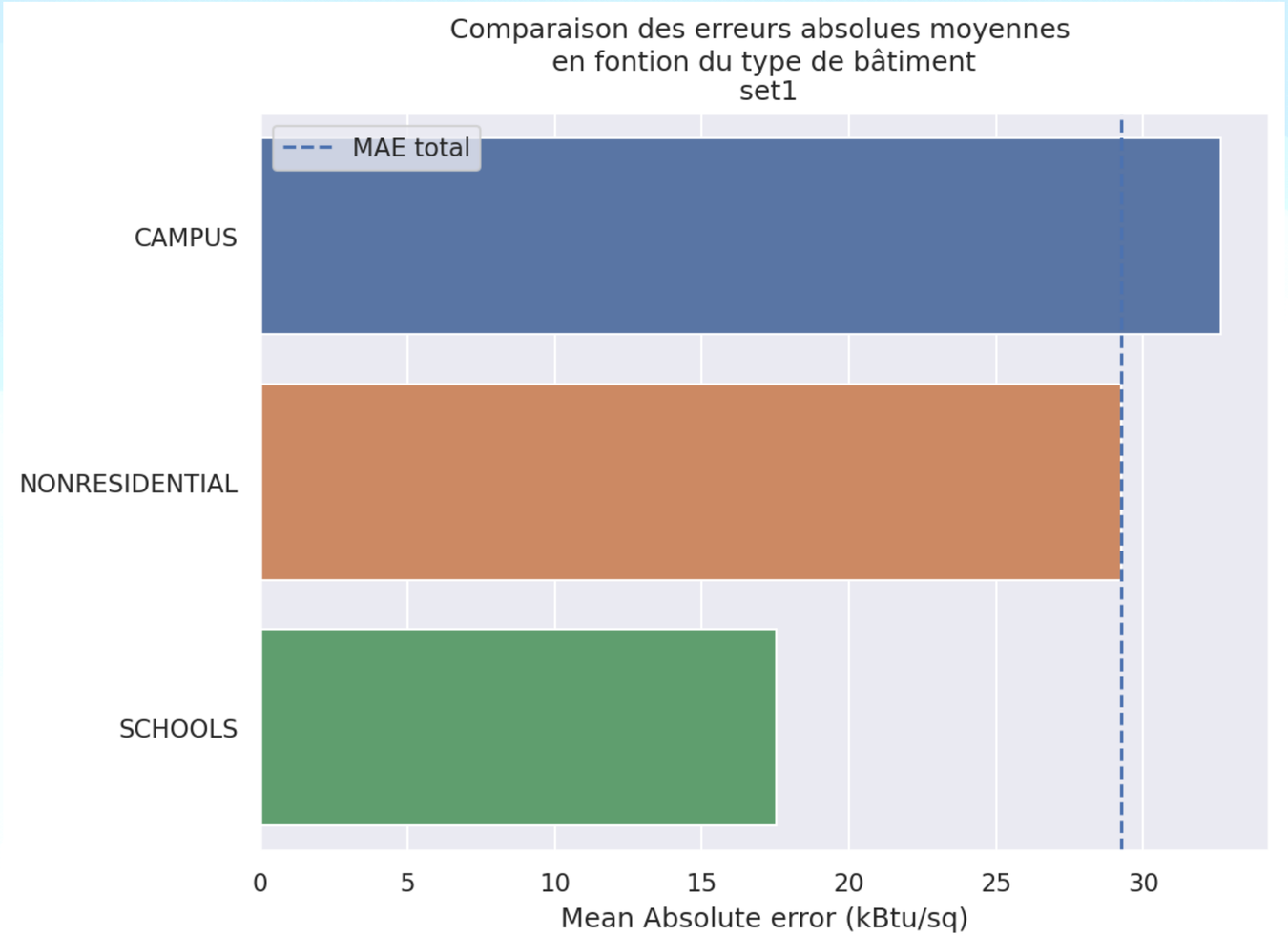
5. Résultats du modèle le plus performant | XGBOOST
Prédiction de la consommation d'énergie surfacique

Modèle XGBOOST



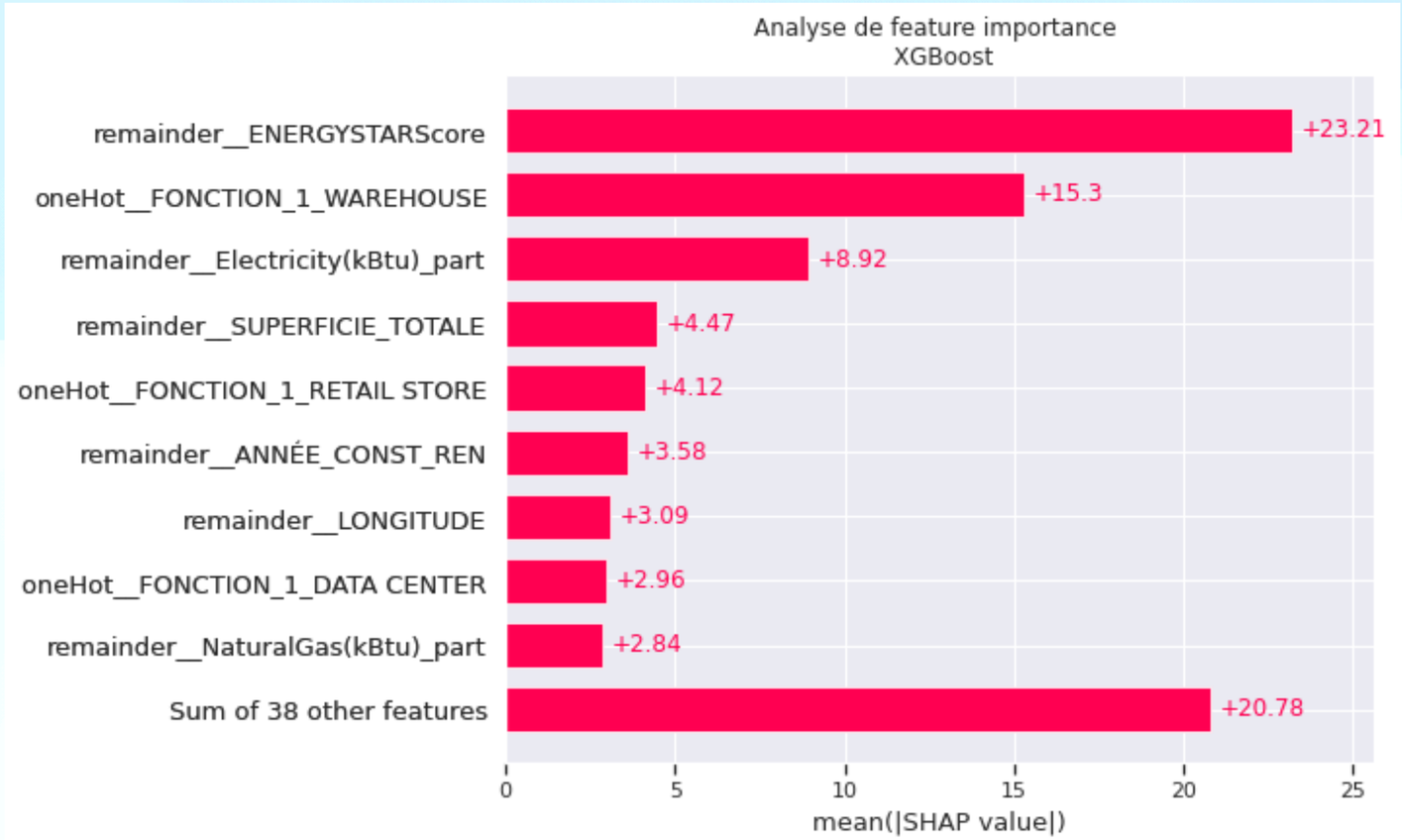
5. Résultats du modèle le plus performant | XGBOOST
Prédiction de la consommation d'énergie surfacique

Modèle XGBOOST



5. Résultats du modèle le plus performant | XGBOOST
Prédiction de la consommation d'énergie surfacique

Modèle XGBOOST



5. Résultats

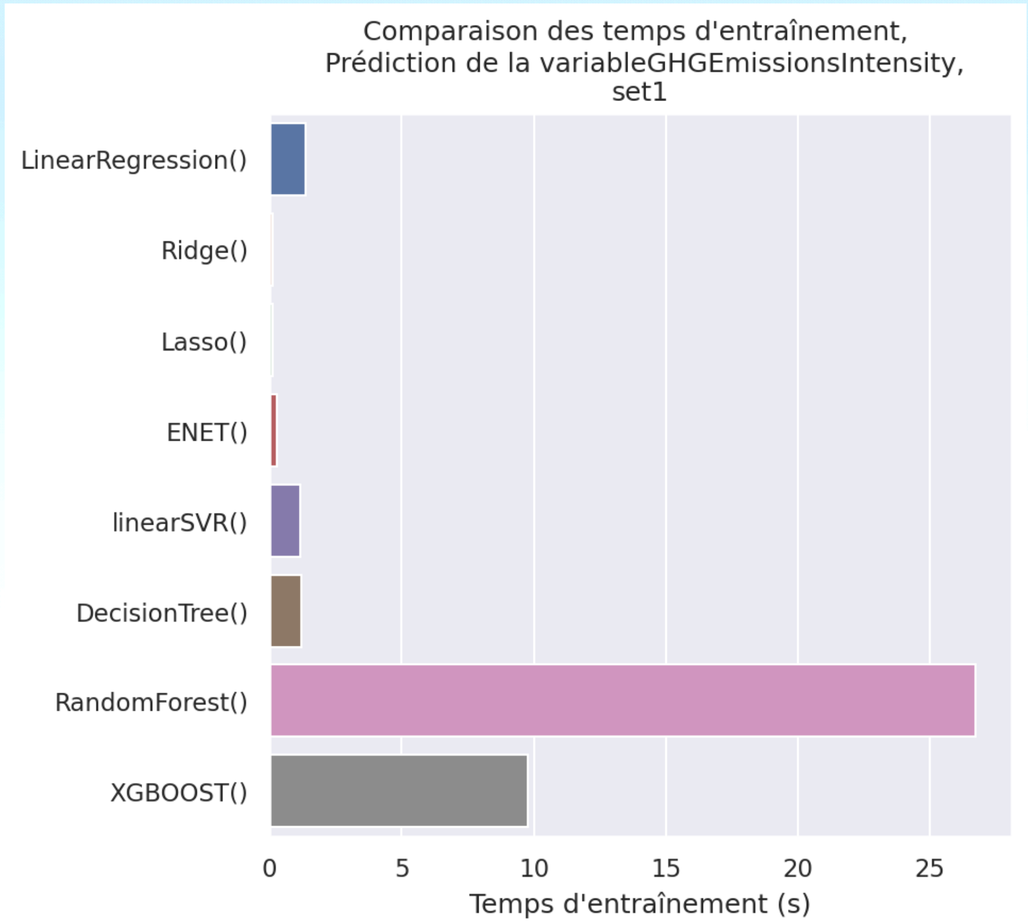
Prédiction de l'intensité des émissions de gaz à effet de serre - set 1

Modèle	best_param	Temps d'entraînement (en s)	MAE moyen Validation croisée	R2 moyen Validation croisée	RMSE Validation croisée
LinearRegression()		1,48	0,67	0,54	1,37
Ridge()	{'regressor__alpha': 5}	0,26	0,67	0,57	1,35
Lasso()	{'regressor__alpha': 0,01}	0,24	0,73	0,48	1,48
ENET()	{'regressor__alpha': 0,001, 'regressor__l1_ratio': 0,2, 'regressor__max_iter': 1000}	0,35	0,67	0,56	1,36
LinearSVR()	{'regressor__C': 0,1}	1,12	0,68	0,54	1,39
DecisionTree()	{'criterion': 'absolute_error', 'max_depth': 2}	1,23	0,91	0,25	1,78
RandomForest()	{'max_depth': 25, 'n_estimators': 125}	28,98	0,74	0,50	1,40
XGBOOST()	{'learning_rate': 0,1, 'max_depth': 15, 'n_estimators': 40, 'subsample': 0,5}	10,49	0,70	0,55	1,32

Modélisation de la variable GHGEmissionsIntensity (tCo2/sq) - set 1
Comparaison des modèles testés

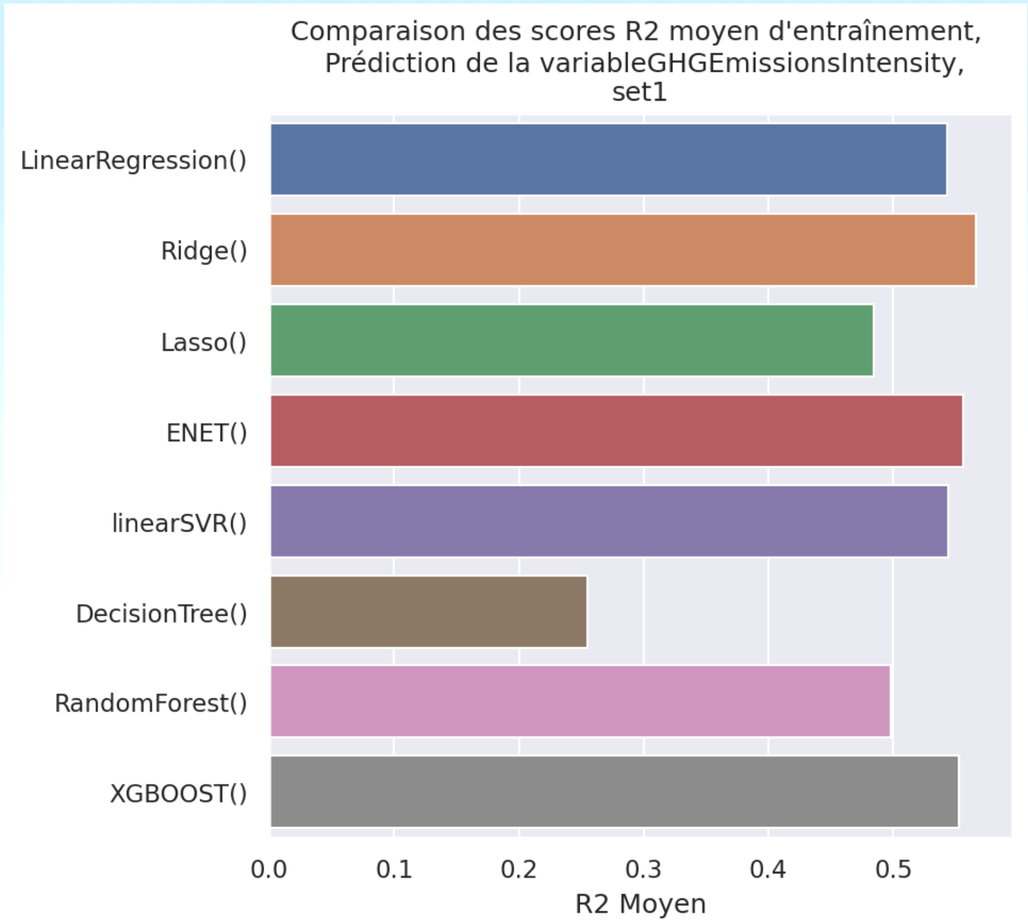
5. Résultats

Prédiction de l’intensité des émissions de gaz à effet de serre - set 1



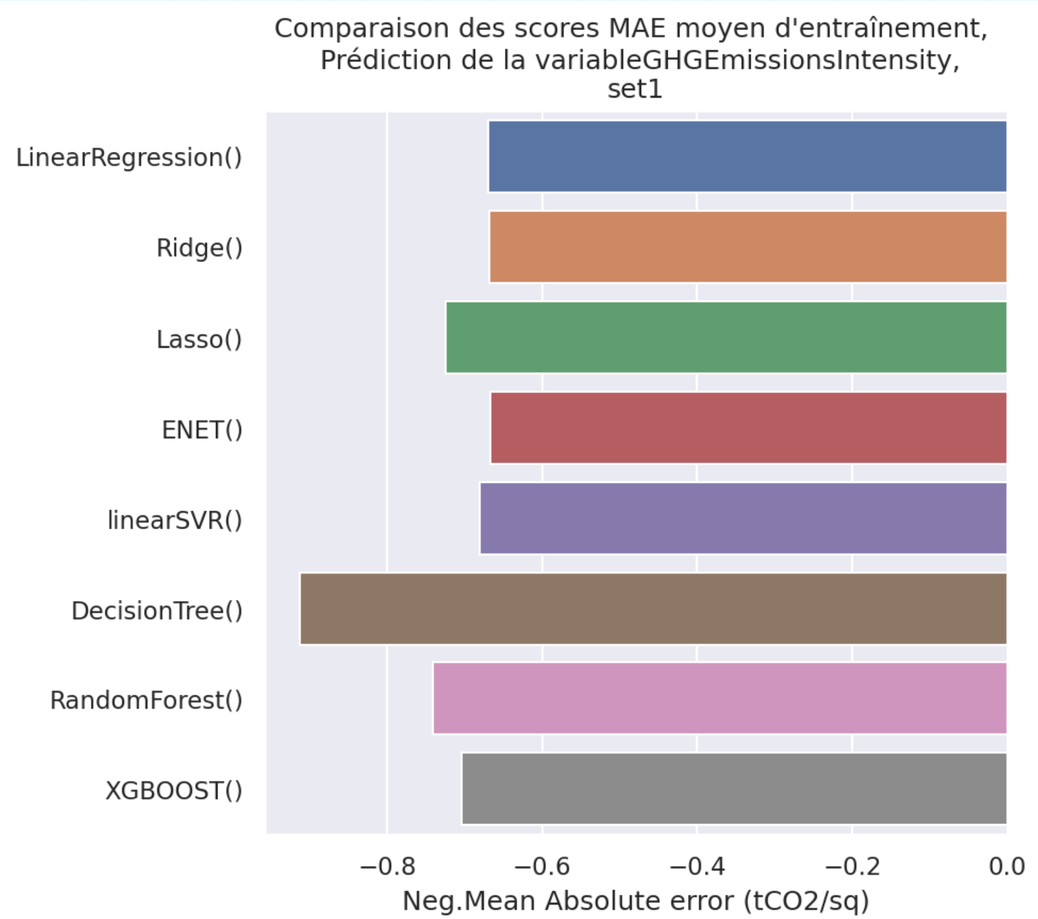
5. Résultats

Prédiction de l’intensité des émissions de gaz à effet de serre - set 1



5. Résultats

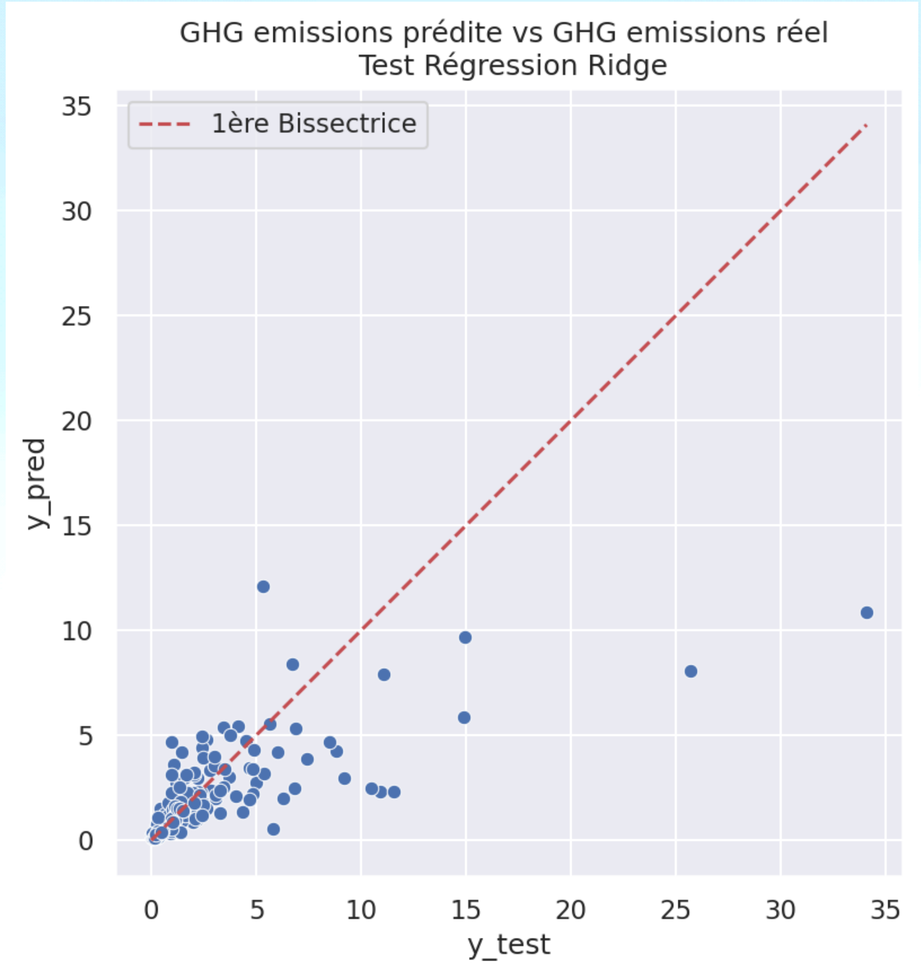
Prédiction de l’intensité des émissions de gaz à effet de serre - set 1



5. Résultats du modèle le plus performant | Ridge avec alpha = 5

Prédiction de l'intensité des émissions de gaz à effet de serre

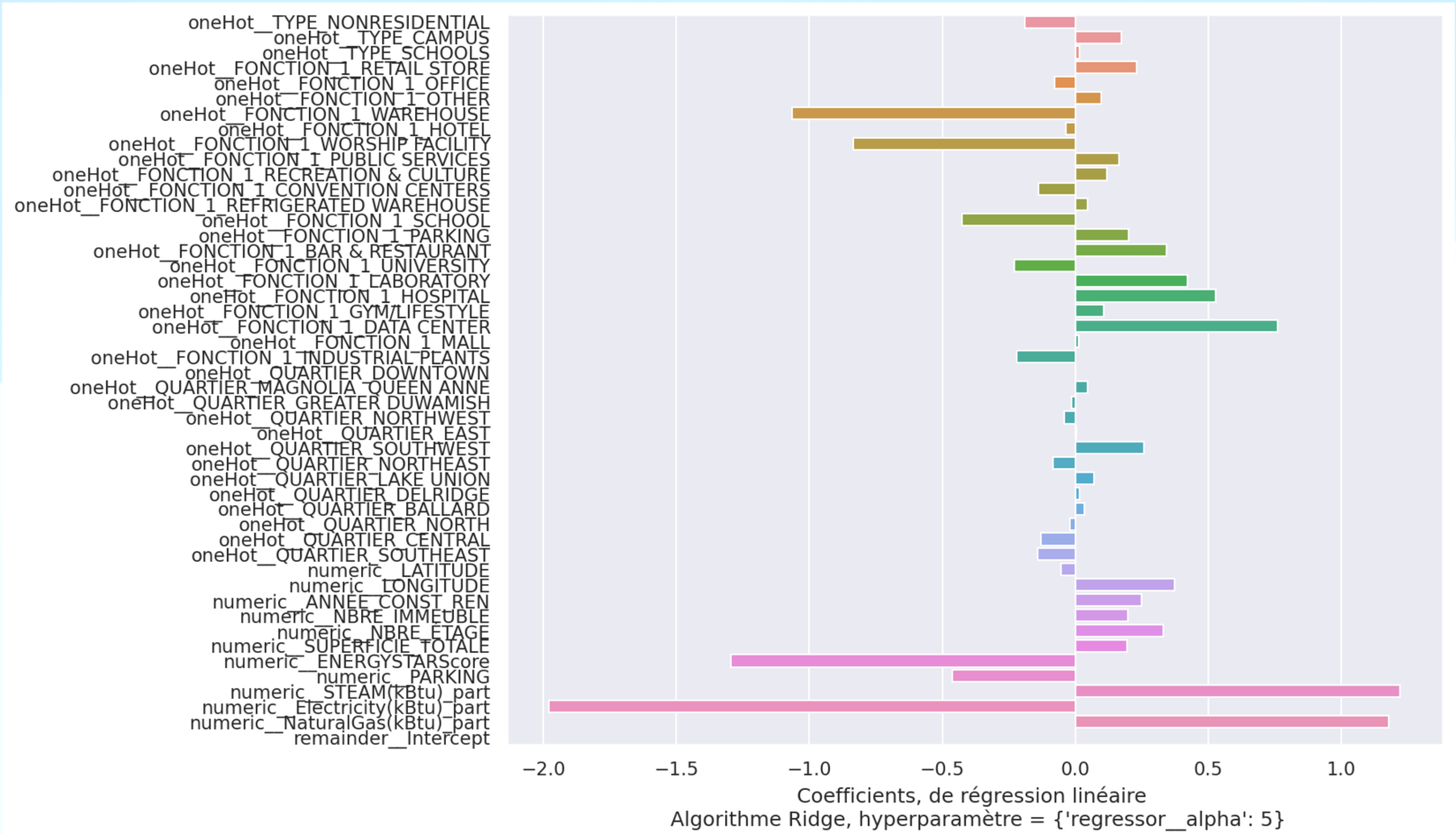
Modèle Ridge



5. Résultats du modèle le plus performant | Ridge avec alpha = 5

Prédiction de l'intensité des émissions de gaz à effet de serre

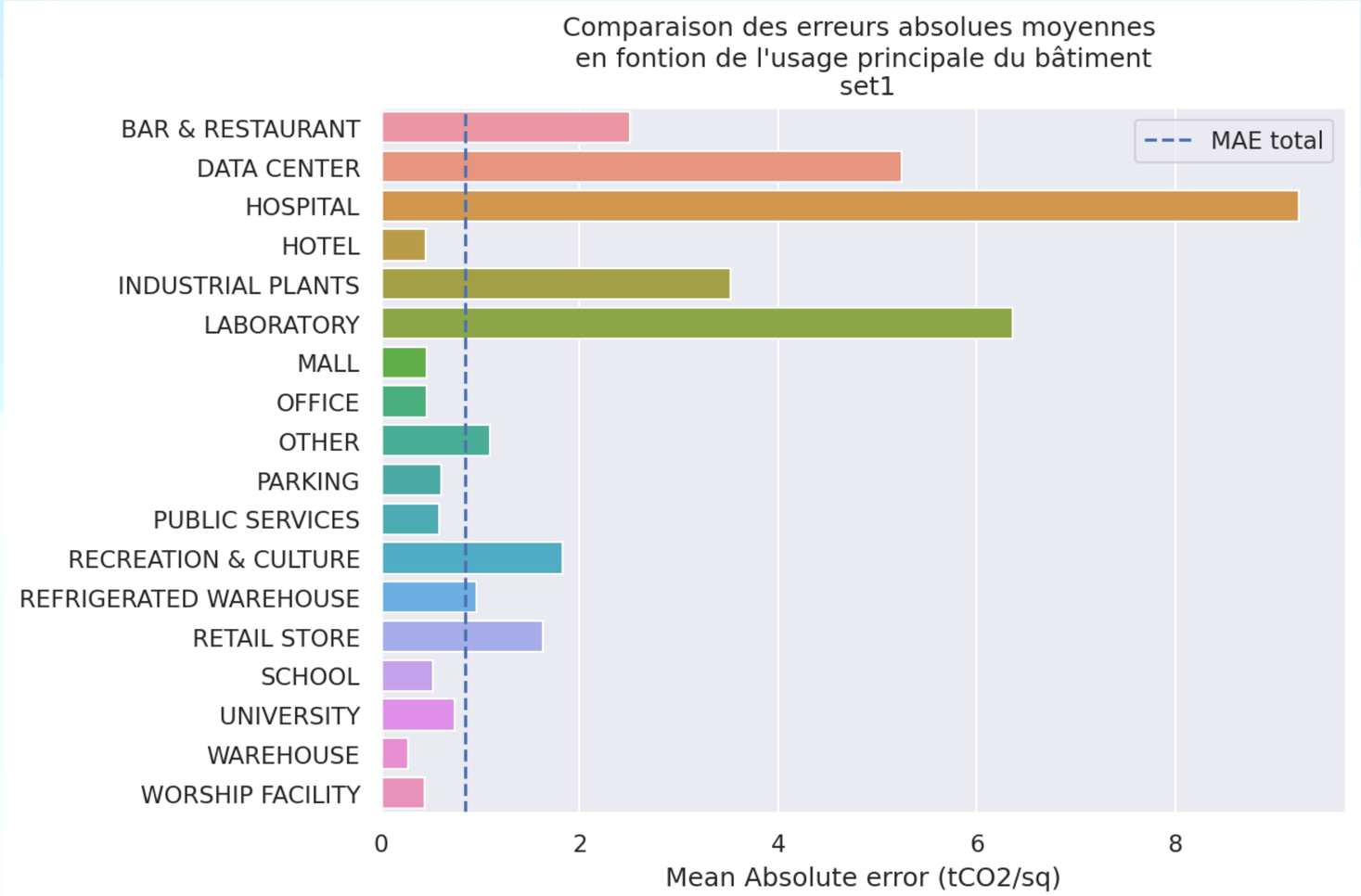
Modèle Ridge



5. Résultats du modèle le plus performant | Ridge avec alpha = 5

Prédiction de l’intensité des émissions de gaz à effet de serre

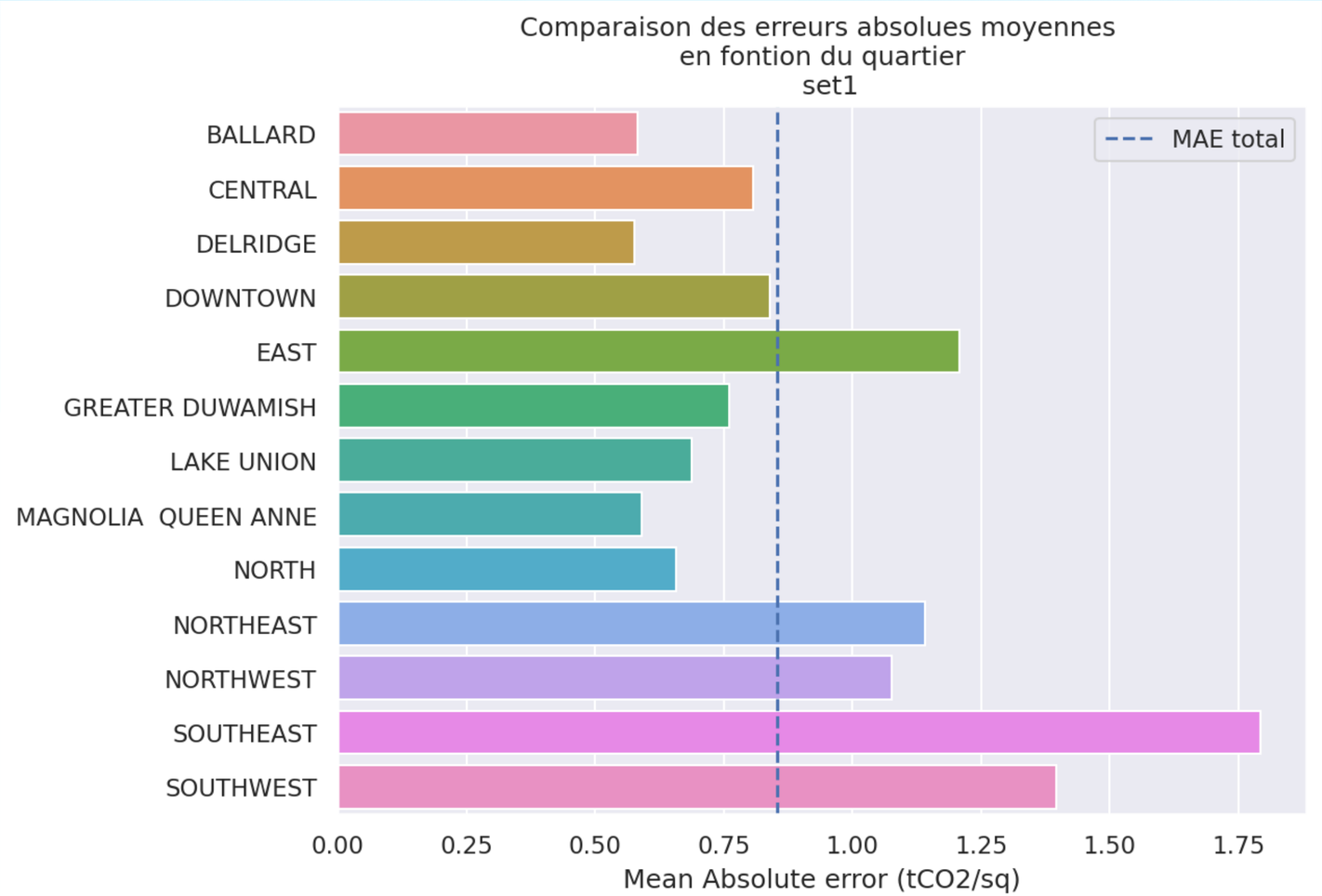
Modèle Ridge



5. Résultats du modèle le plus performant | Ridge avec alpha = 5

Prédiction de l'intensité des émissions de gaz à effet de serre

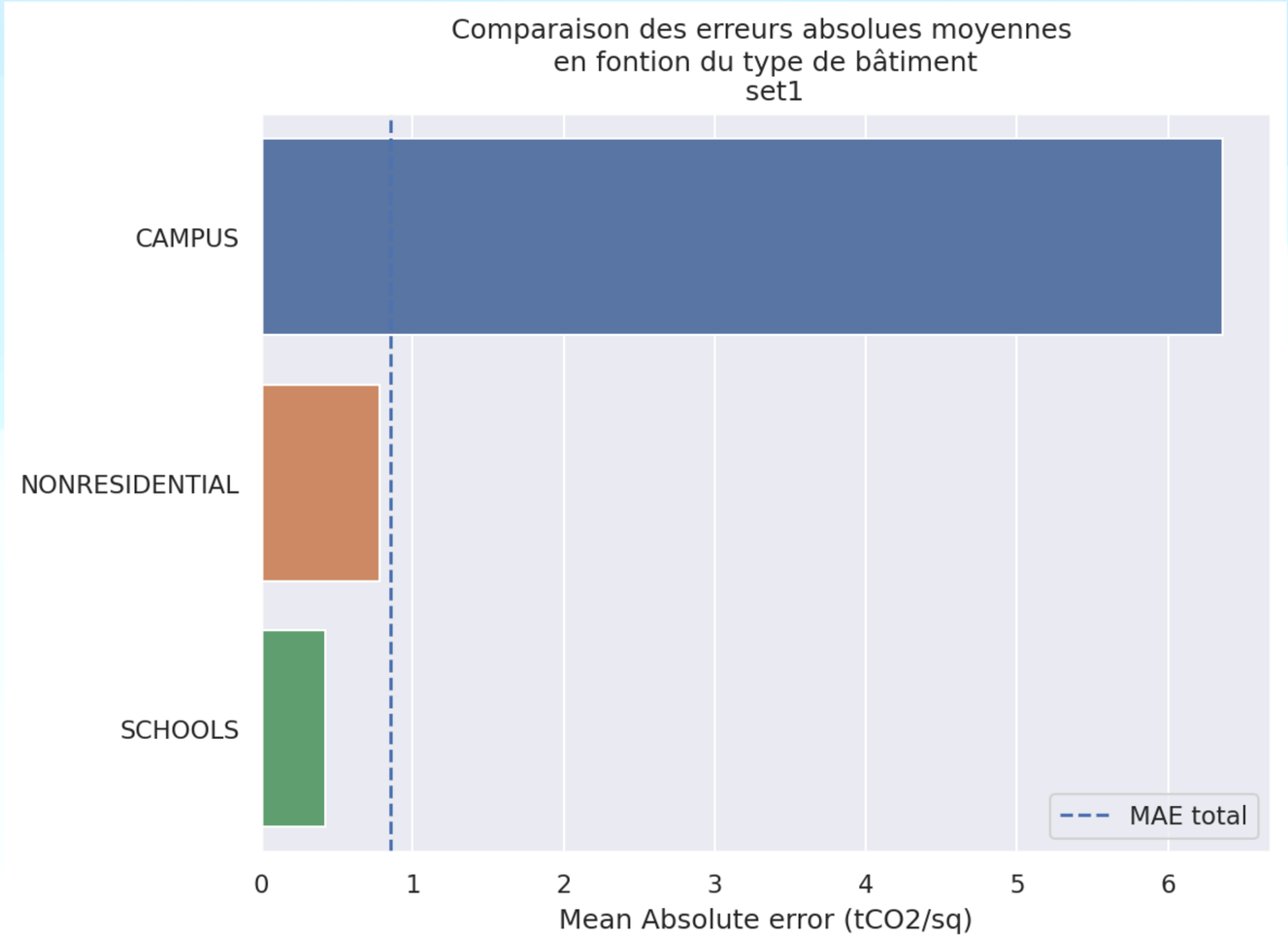
Modèle Ridge



5. Résultats du modèle le plus performant | Ridge avec alpha = 5

Prédiction de l'intensité des émissions de gaz à effet de serre

Modèle Ridge



5. Résultats

Piste d’amélioration

- Utilisation de l’analyse de feature importance pour affiner le feature engineering

- Utilisation du calcul d’erreur par catégorie, pour affiner l’encodage des variables catégorielles

Conclusion

- **8 algorithmes** ont été testés par validation croisée et recherche des hyperparemètres optimaux.
- L'algorithme XGBoost est le plus performant dans la prédiction de **la consommation totale d'énergie des bâtiments non destinés** à l'habitation de la ville de Seattle.
- L'algorithme Ridge est le plus adaptés à la prédiction des **émissions de gaz à effet de serre.**
- **L'ENERGYSTARScore influe fortement sur les deux target.**
- Les résultats pourraient être améliorés par un feature engineering plus fin axé sur les catégorie pour lesquelles l'algorithme commet le plus d'erreur.