Projet 4

Anticipez la consommation électrique de bâtiments

Seattle

Problématique

Données

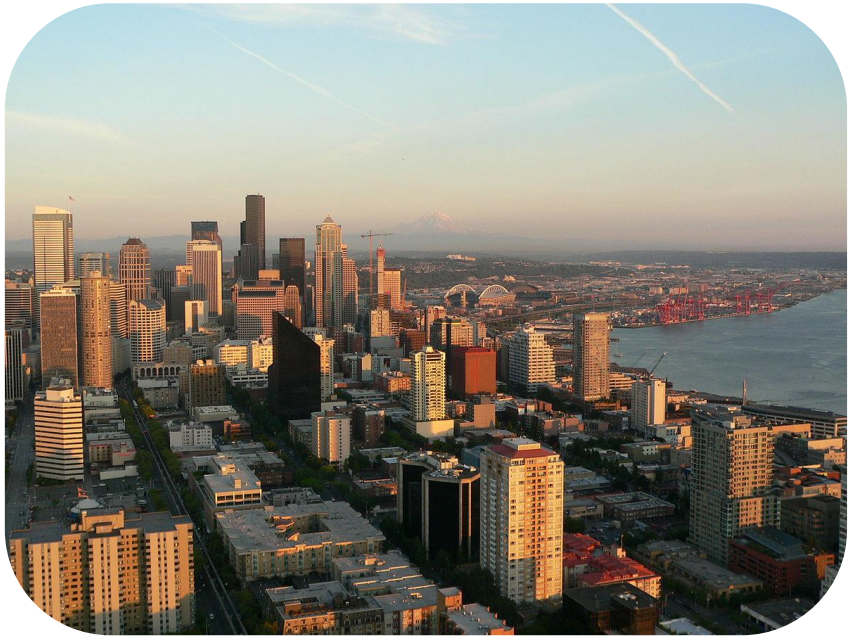
Modélisation

Conclusions

Camille BRODIN

Problématique

Missions confiées par la ville de Seattle:



Base de données de relevés par les agents de la ville :

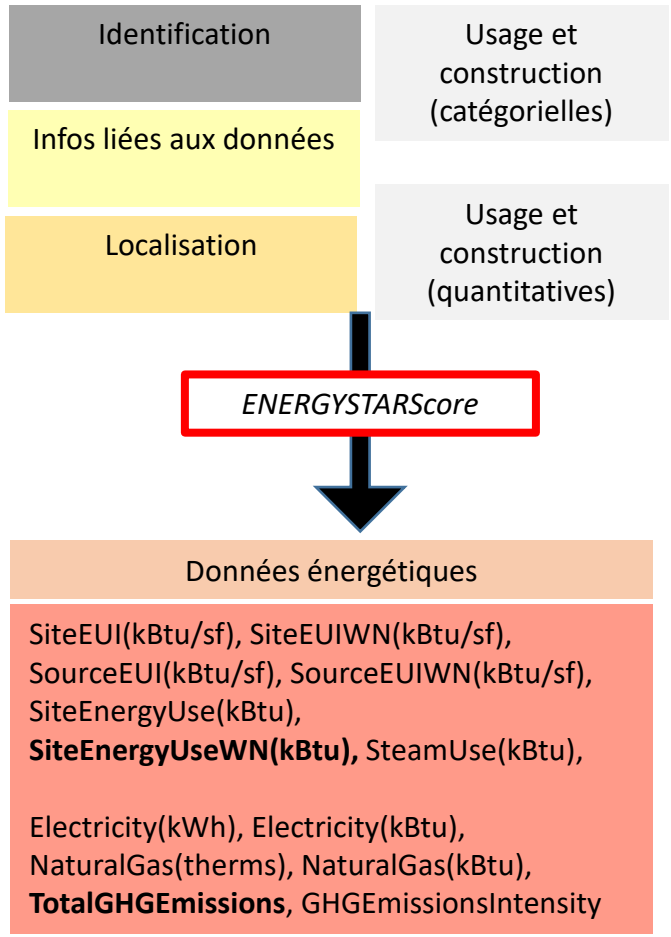
- Effectués en 2016
- Relevés coûteux à obtenir
- 3 376 bâtiments décrits par 46 colonnes

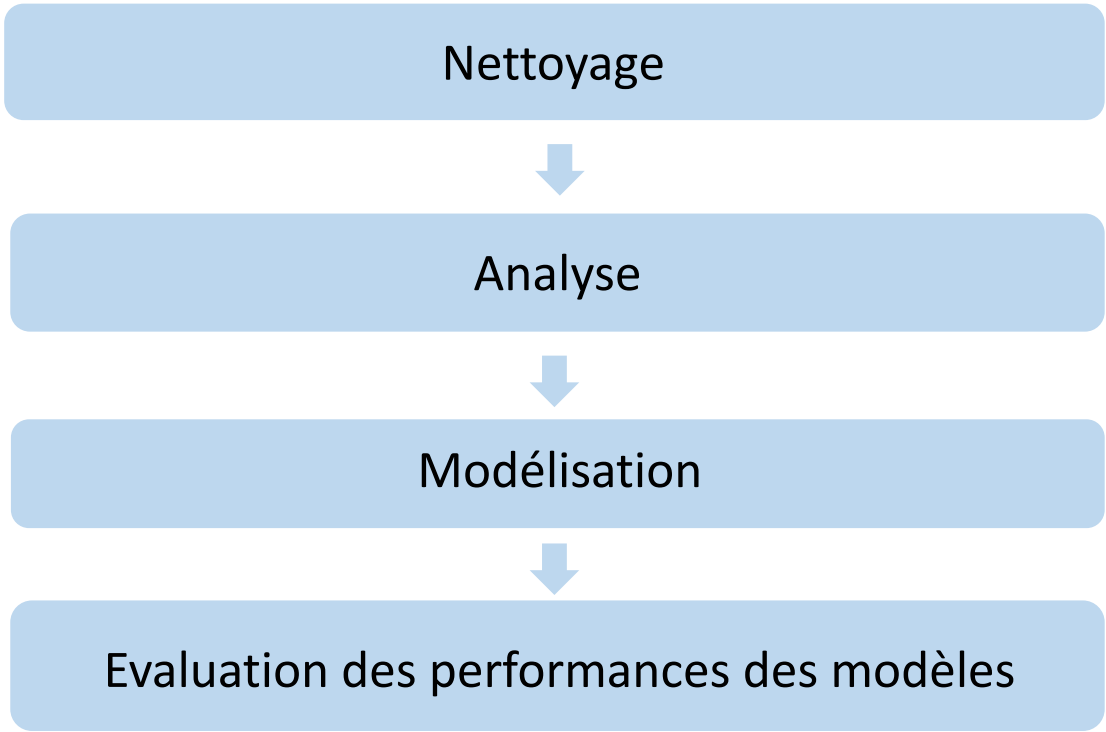
Trois missions :

- Prédire les émissions de CO2 des bâtiments hors habitations. *Total GreenHouse Gases (GHG)*
- Prédire la consommation totale d'énergie de ces bâtiments. *Site Energy Use (SEU)*
- Evaluer l'intérêt de l'"ENERGYSTARScore" pour la prédiction d'émissions

➢ **Pour atteindre son objectif de ville neutre en émissions de carbone en 2050**

Problématique





Données : Nettoyage

Justifications	Lignes/colonnes restantes	Méthodes
0. Harmonisation des variables	3376 / 46	data[].apply(lambda x: split_dates(x)) data[].applymap(str.upper)
1. Filtrage projet : Bâtiments hors habitations	3376 / 46 -> 1668 / 46	data[data['BuildingType'].isin([])]
2. Elimination des lignes inexploitables + colonnes a var nulle et doublons	1668 / 46 -> 1597, 42	data.drop(index=data[data[]==0].index) data.drop(columns=["City", "State", "DataYear“, “YearsENERGYSTARCertified”])
3. Outliers : Exploration individualisée et connaissances métiers (voir détails annexes)	1597, 42 -> 1570, 41	data.loc[(data["Outlier"] == 'not’)] data.loc[data['PropertyGFATotal']<= 1800000.00]
4. Imputations des données a) Remplacer par la valeur 0/None (+50%) b) 40 valeurs à imputer sur 3 colonnes c) ENERGYSTARScore non imputé (rempli à 60%)	1570, 41 -> 1049, 41	a)data[c] = data[c].fillna(0) ou .fillna(‘None’) b)knn_impute(data, var_model = filled_cols, var_target='LargestPropertyUseType’, ‘LargestPropertyUseTypeGFA’ , ‘ZipCode’) **c)data.dropna(subset=['ENERGYSTARScore'])**

- **3376 lignes et 46 colonnes sur le jeu de données brut**
- **1049 lignes et 41 colonnes sur le jeu de données nettoyé**

Données : Sélection et création de variables (feature engineering)

2 variables quantitatives cibles

- *SiteEnergyUseWN(kBtu)*
- *TotalGHGEmissions*

+2 nouvelles variables quantitatives

➢ Age des bâtiments (« **BuildingAge** »)
2016 – Année de construction

➢ Surface moyenne /étage (« **MeanGFAperFloor** »)
surface totale/(nb d'étages +1)

+1 nouvelle variable catégorielle

➢ Principale énergie consommée (« **MainEnergy** »)
Steam, Electricity, NaturalGas

+ 9 variables quantitatives existantes et exploitables

- **Profil énergétique**

ENERGYSTARScore

- **Usages des bâtiments**

LargestPropertyUseTypeGFA, SecondLargestPropertyUseTypeGFA, ThirdLargestPropertyUseTypeGFA,

- **Surfaces et état du bâtiment**

NumberofBuildings, NumberofFloors, PropertyGFATotal, PropertyGFAParking, PropertyGFABuilding(s)

+ 7 variables catégorielles existantes et exploitables

- **Usages des bâtiments**

BuildingType, PrimaryPropertyType, LargestPropertyUseType, SecondLargestPropertyUseType, ThirdLargestPropertyUseType,

- **Emplacement des bâtiments**

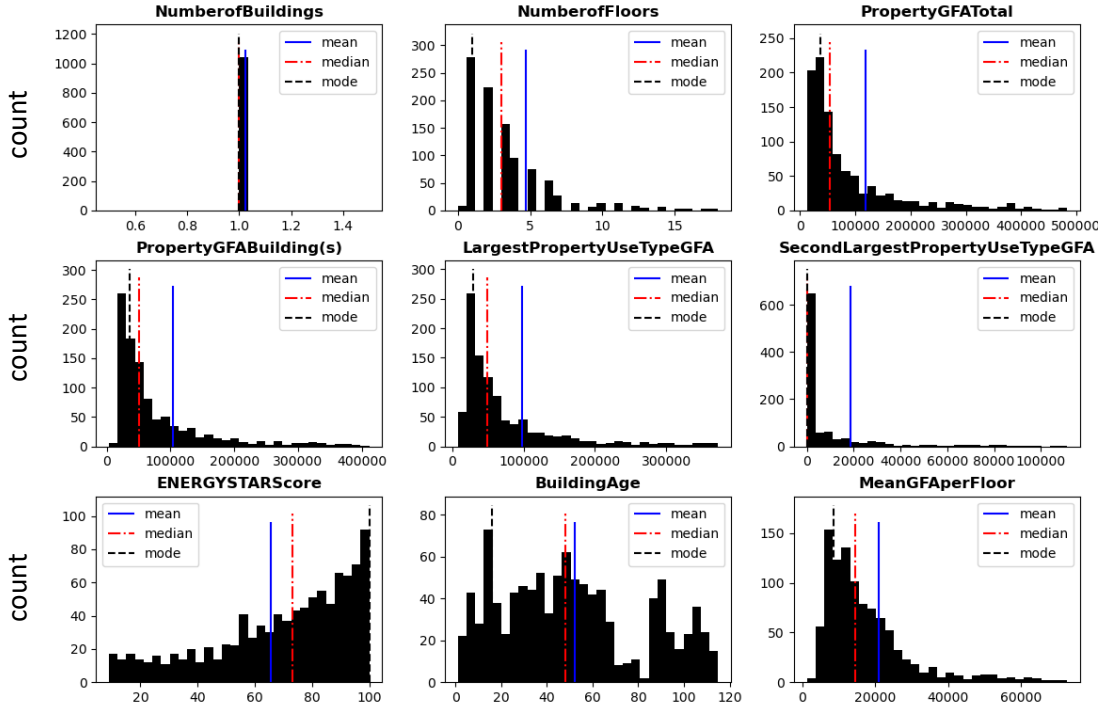
CouncilDistrictCode, Neighborhood,

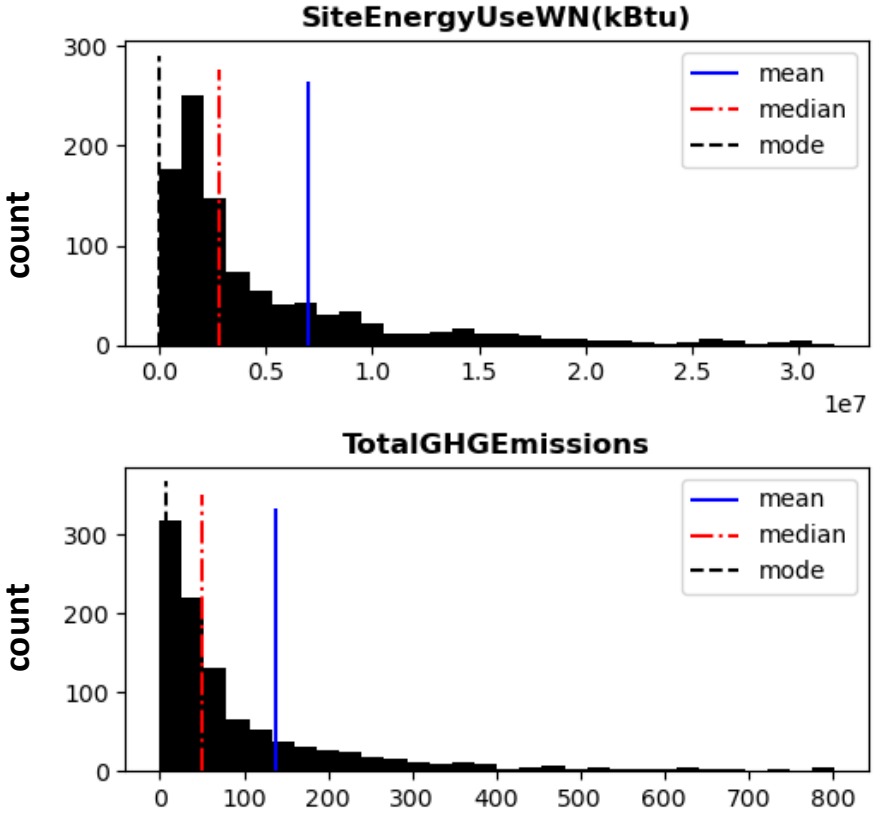
➢ **1049 lignes et 41 (+3) = 44 colonnes sur le jeu de données nettoyé**
➢ **1049 lignes et 19 features + 2 cibles sur le jeu de données (+3 variables d'ID)**

Analyses univariées des données

❖ **Variables quantitatives :**

→ **Test de normalité (histogramme, Shapiro-Wilk)**

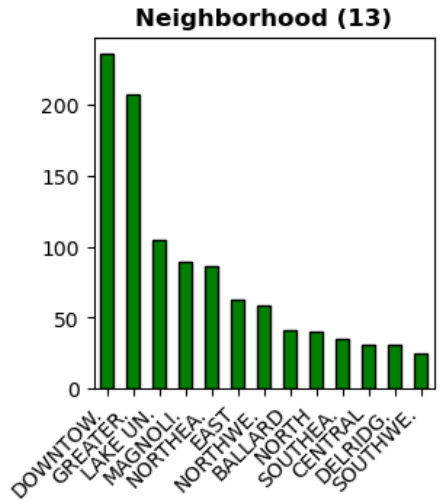


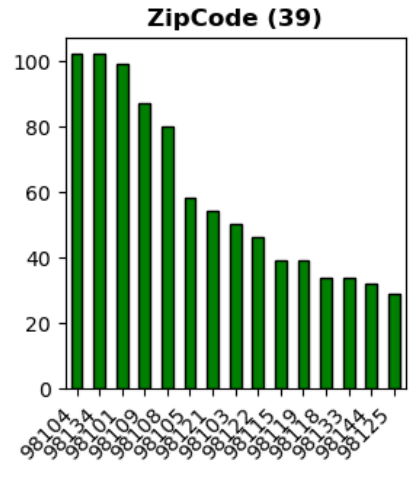


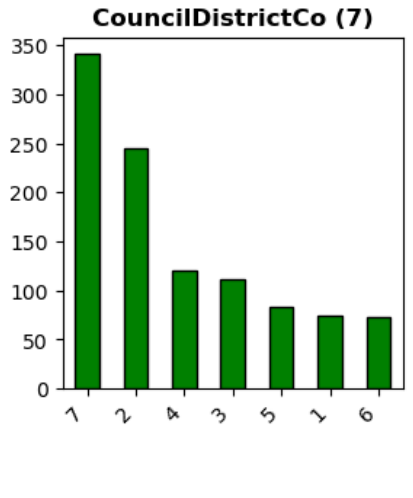
- **Distributions non gaussiennes très rassemblées sur la gauche dans les variables d'études p<0,05.**
- **Nous testons deux conditions sur le df : scaling MinMax (A) ou transformation logarithme népérien +1 (B)**

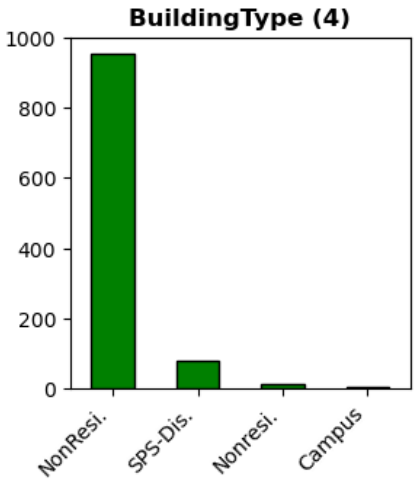
Analyses univariées des données

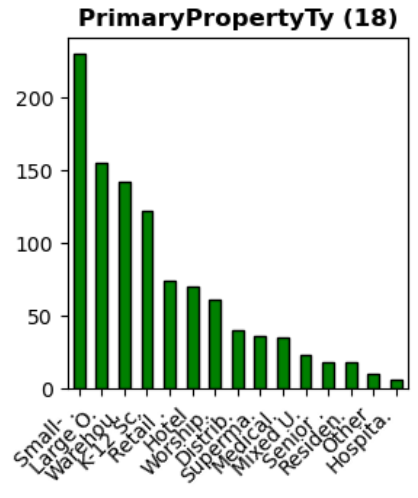
❖ Variables qualitatives

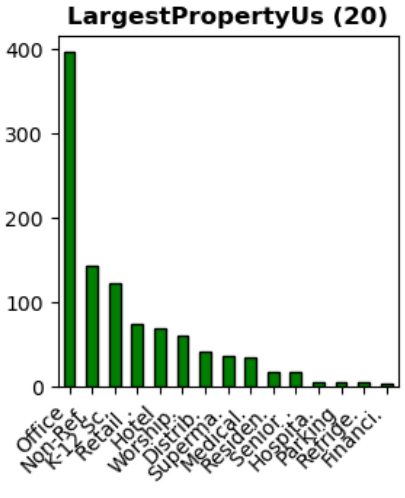


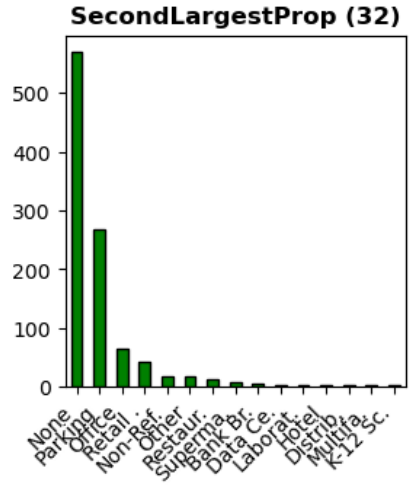


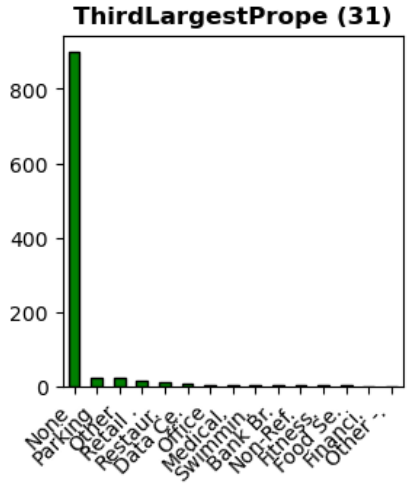


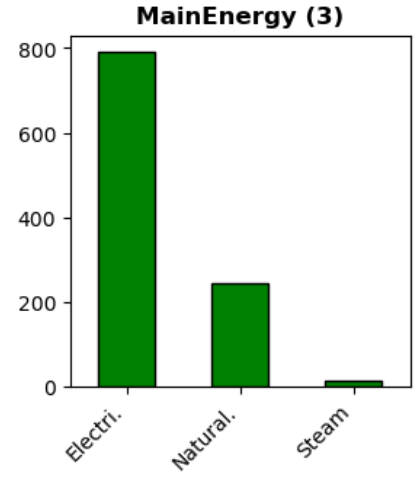






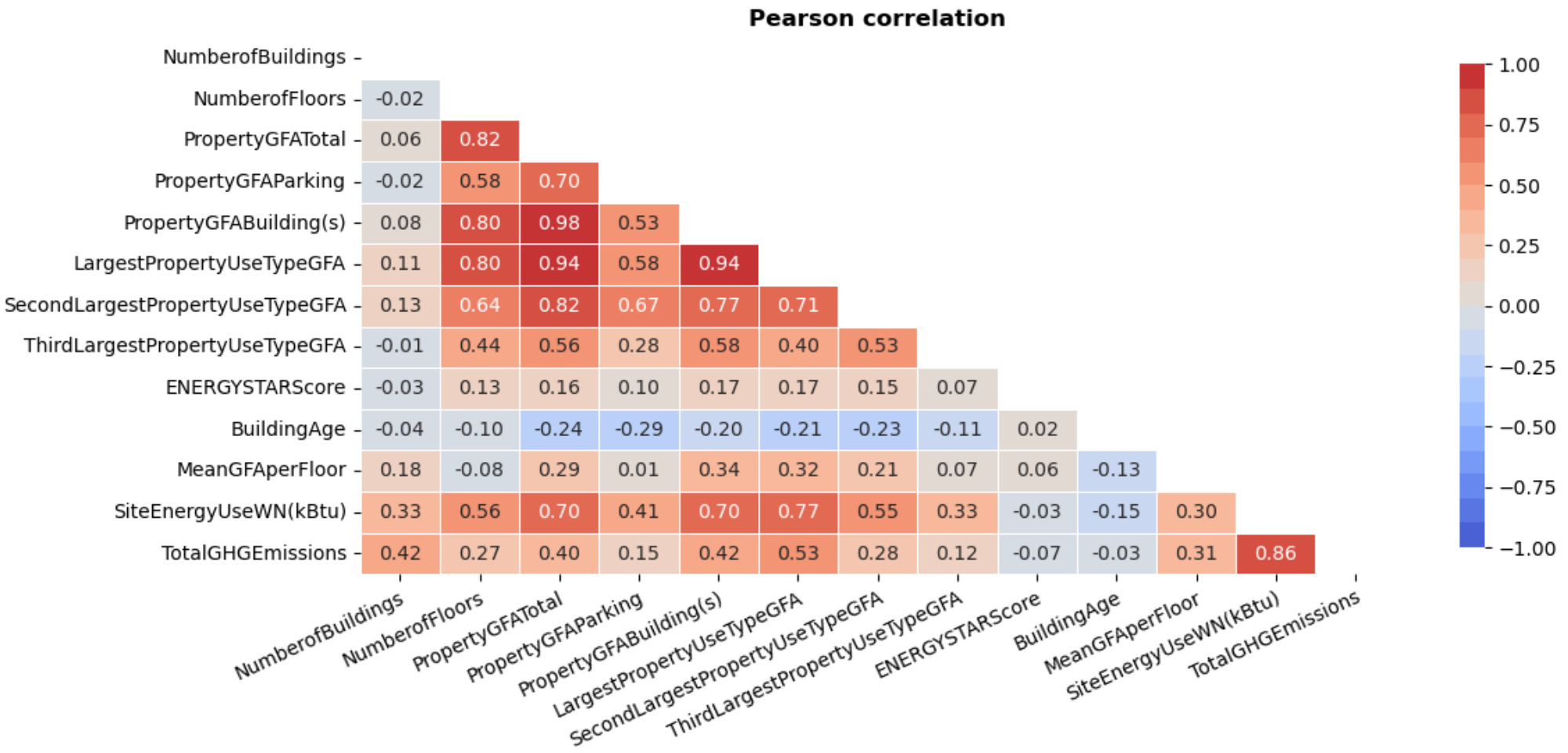






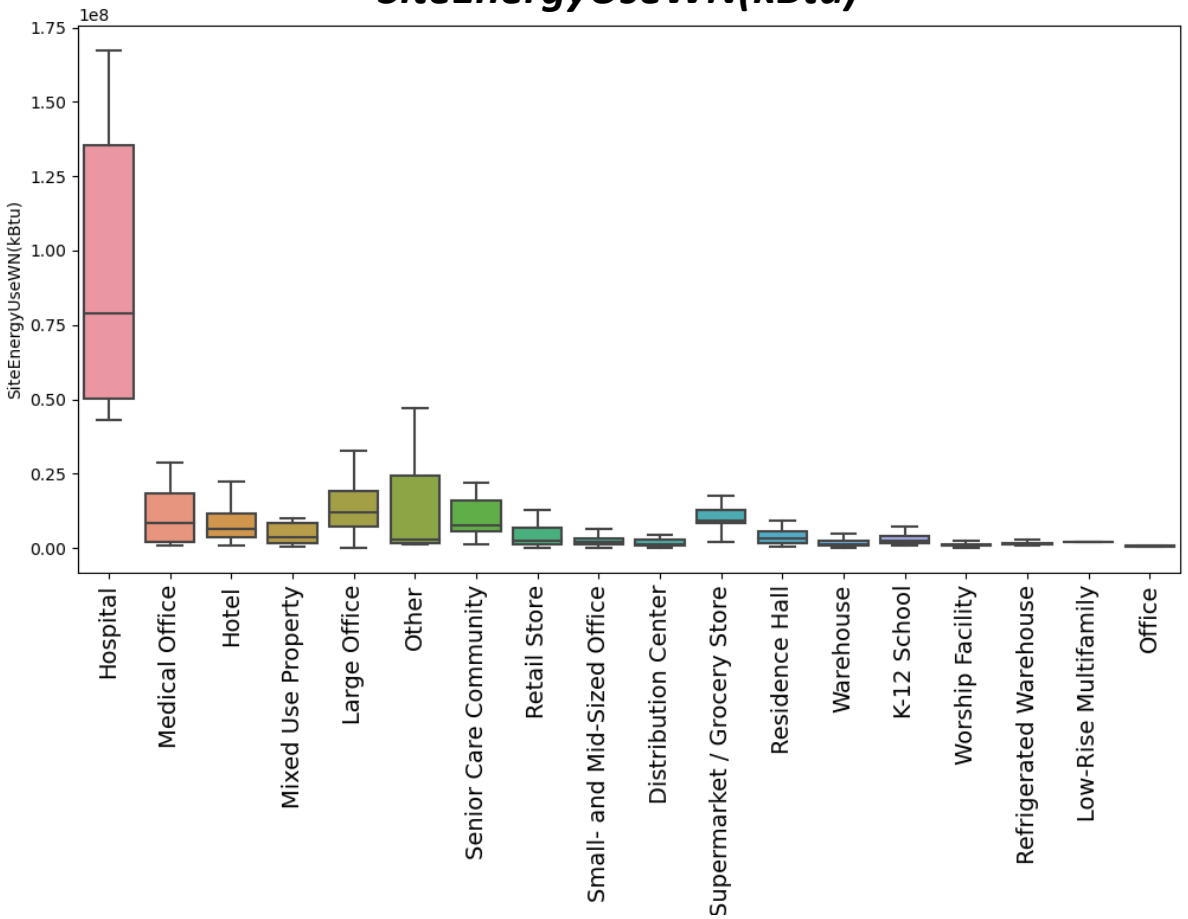
Analyses bivariées des données

❖ **Corrélation de Pearson**

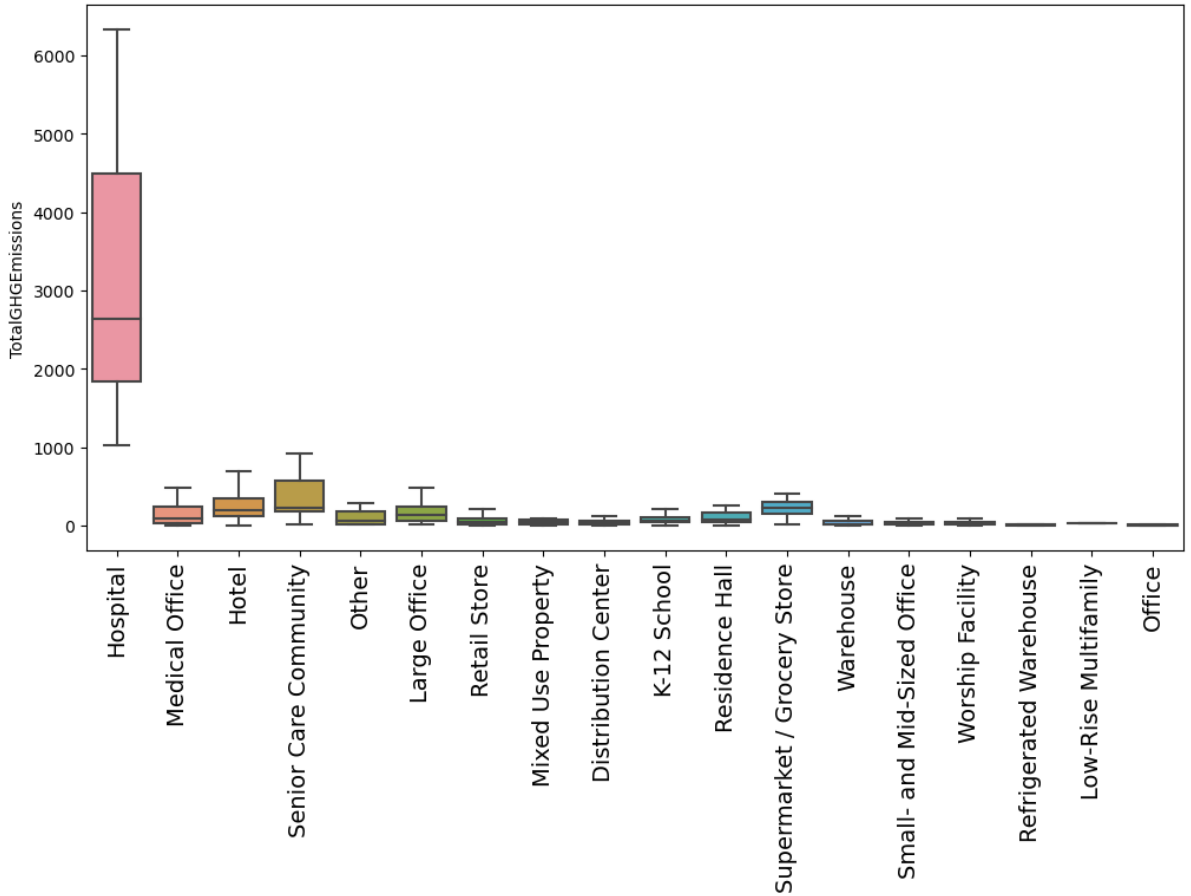


Analyses bivariées des données

SiteEnergyUseWN(kBtu)



TotalGHGEmissions



➢ **Les hôpitaux sont les bâtiments qui utilisent le plus d'énergie, et ceux qui émettent le plus de CO2**

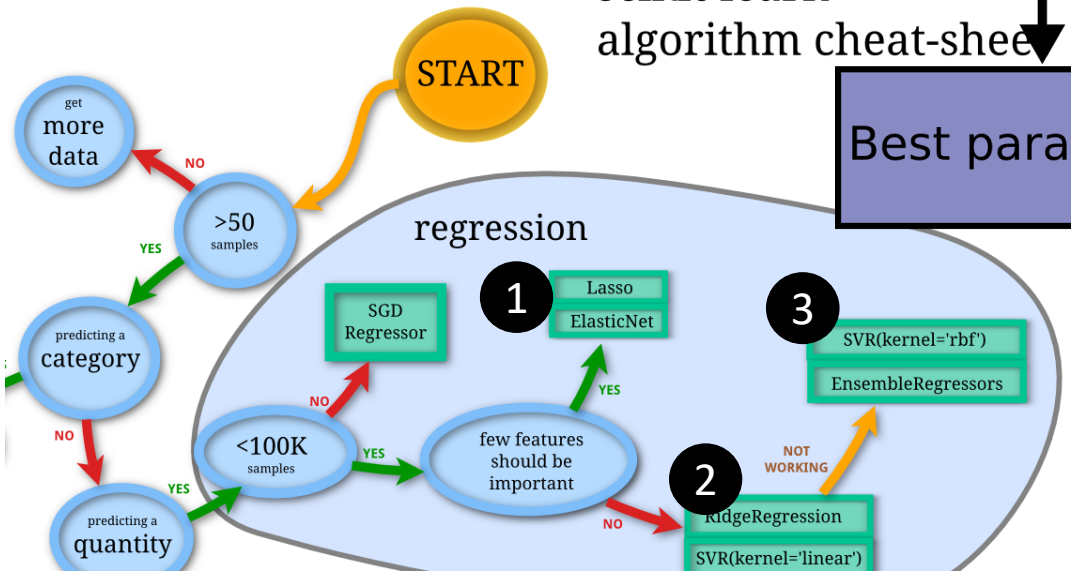
Modélisation

Problématique

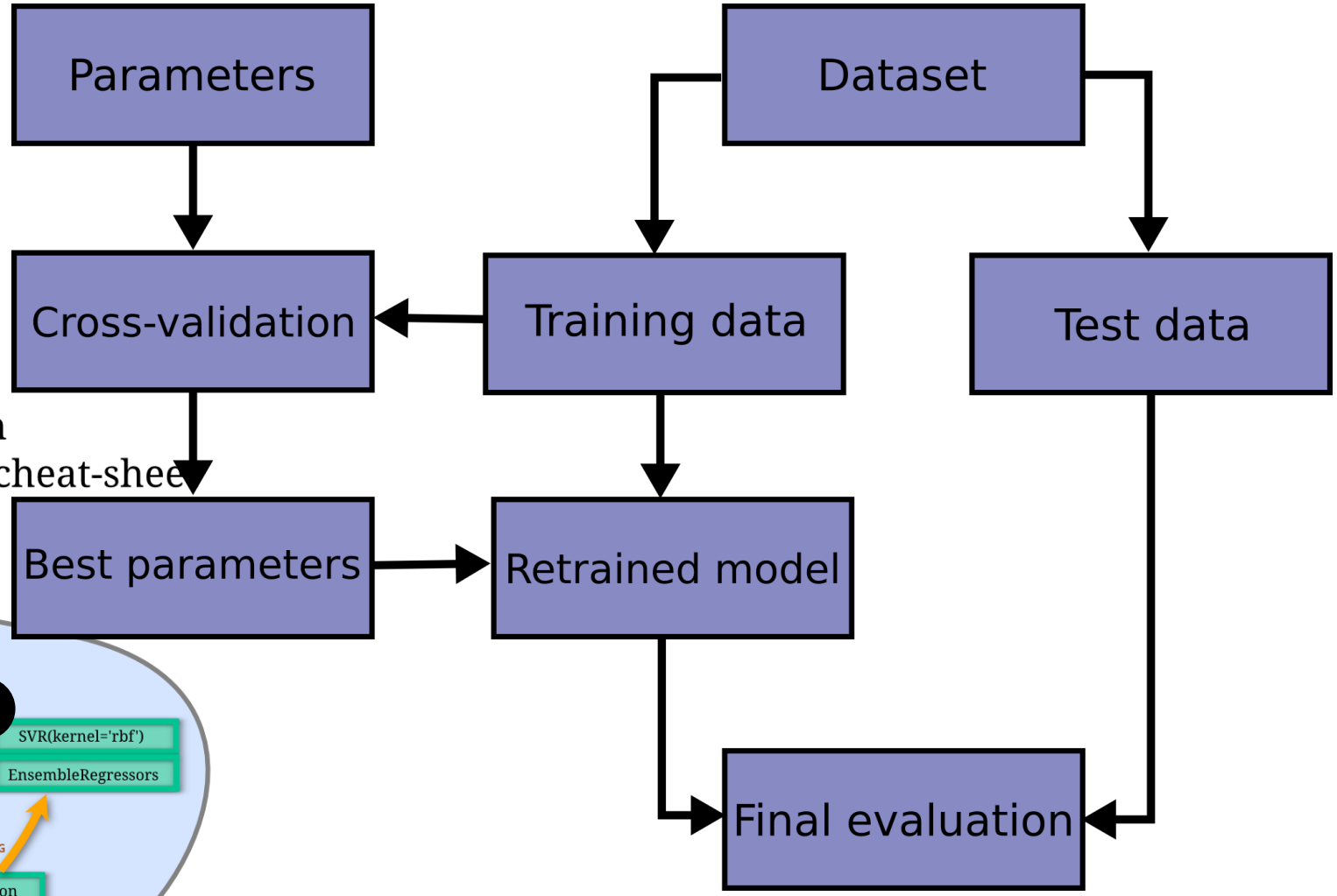
Données

Modélisation

Conclusions



scikit-learn algorithm cheat-sheet



Modélisation : Features engineering (suite) : encodage, transformation

ENCODAGE CATEGORIES

- **OneHotEncoder** si cardinalité faible (<5 modalités)
- **LabelEncoder** pour les autres

TRANSFORMATIONS FEATURES

- **MinMaxScaler**
- **Logarithme népérien (+1)**

VALIDATION CROISEE

- *GridSearchCV* **-> ré-entrainement avec hyperparamètres optimisés**

EVALUATION MODELE REGRESSION

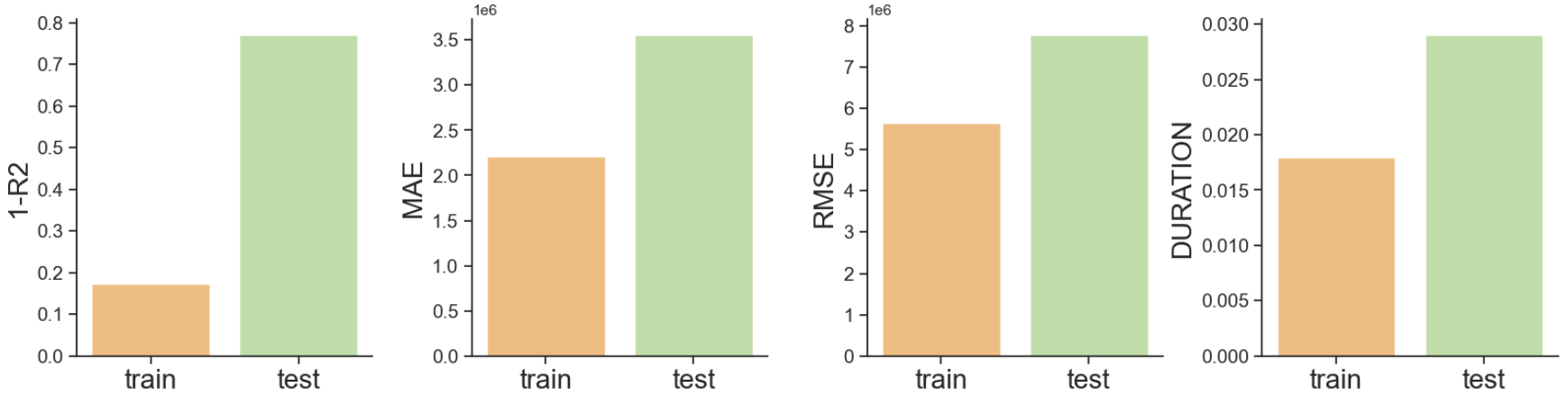
On choisit de calculer :

- **Le coefficient de détermination (R2)** pour comparer les modèles entre eux.
- **L'erreur absolue moyenne (MAE)** pour sa pertinence business et son intuitivité.
- **L'écart quadratique moyen (RMSE)** pour la pénalisation des erreurs opérée.

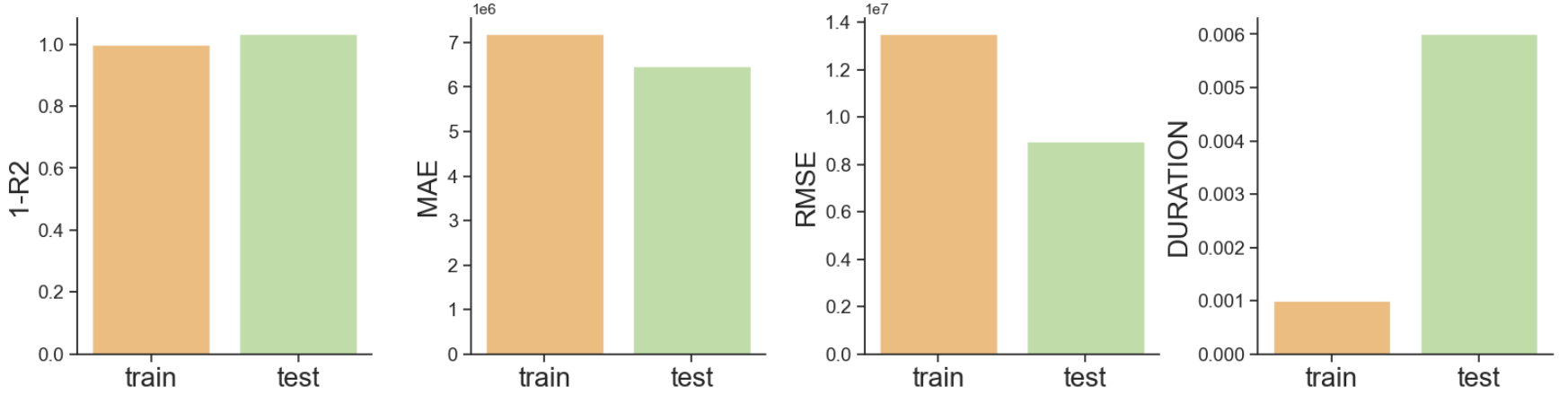
➢ **Entrainement des modèles sur X_train (19 features + 5 variables post-encodage, 24 features totales) et y = 1 cible.**

Modélisation 0 : KNN avant validation croisée

KNeighborsRegressor (n_neighbors=2)



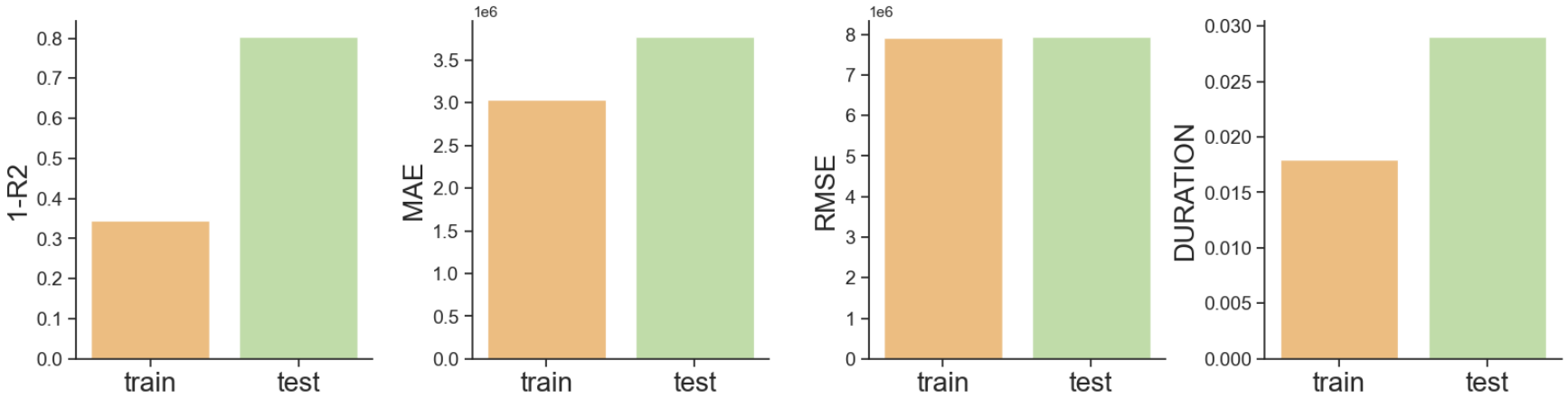
dummy.DummyRegressor (strategy='mean')



- **L'algorithme des k plus proches voisins est meilleur qu'une prédiction aléatoire, et qu'un DummyRegressor (mean)**

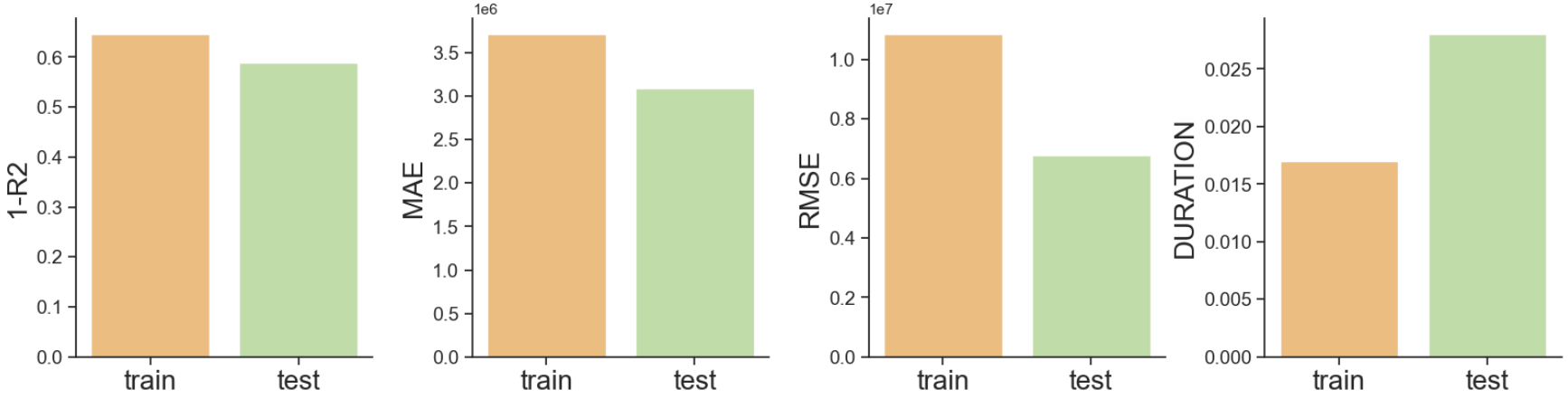
Modélisation 0 : KNN après validation croisée

KNeighborsRegressor (n_neighbors=4)



Avec transformation log

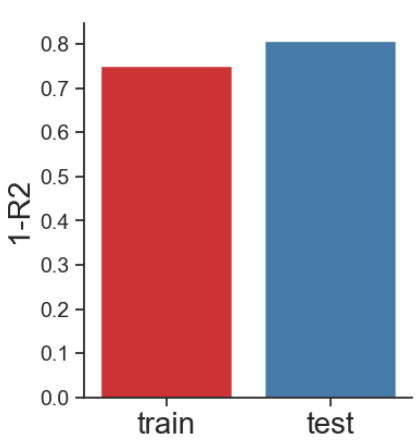
KNeighborsRegressor (n_neighbors=10)

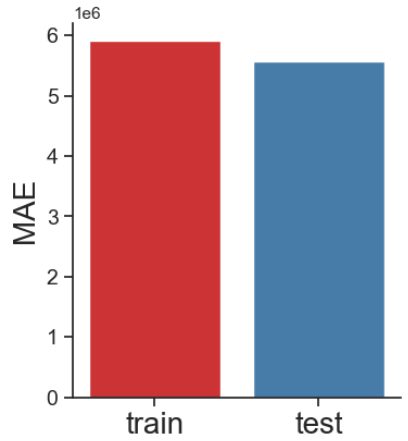


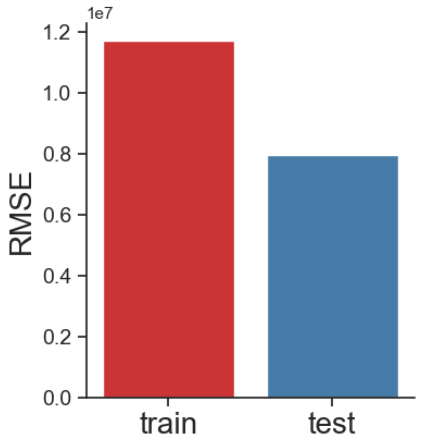
- **Après validation croisée Gridsearch, les hyperparamètres recommandés sont n_neighbors = 4 et 10 pour log**

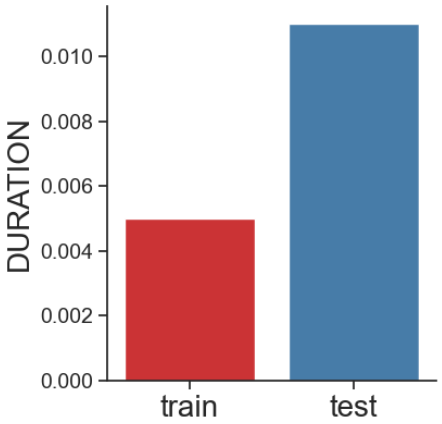
Modélisation 1 : Modèle linéaire | Elastic après CV

ElasticNet(alpha=0.01)



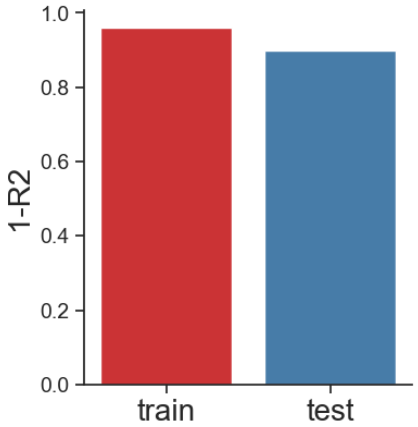


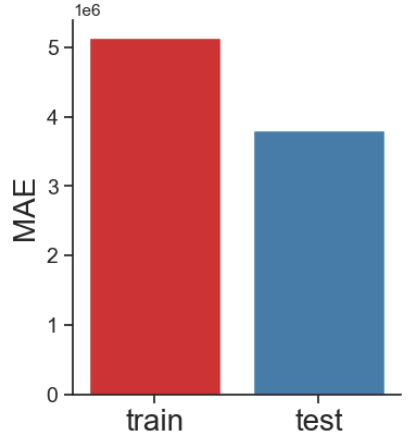


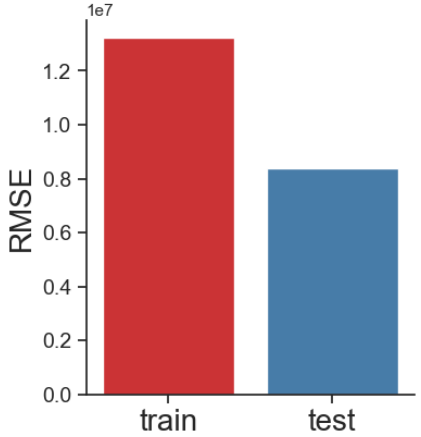


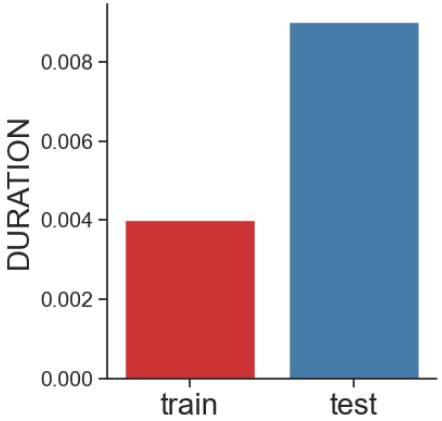
Avec transformation log

ElasticNet(alpha=1)



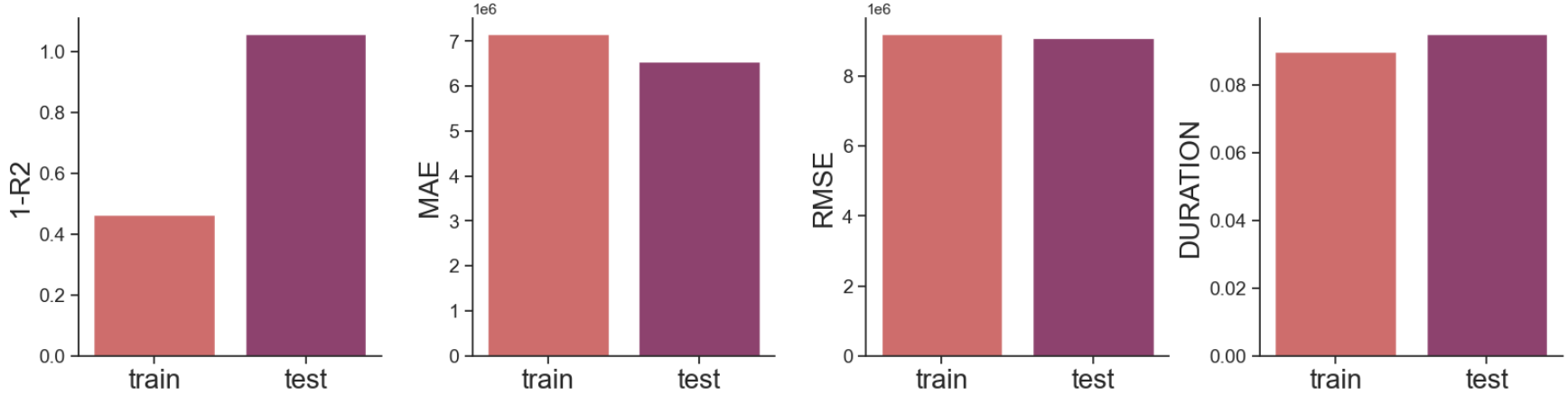






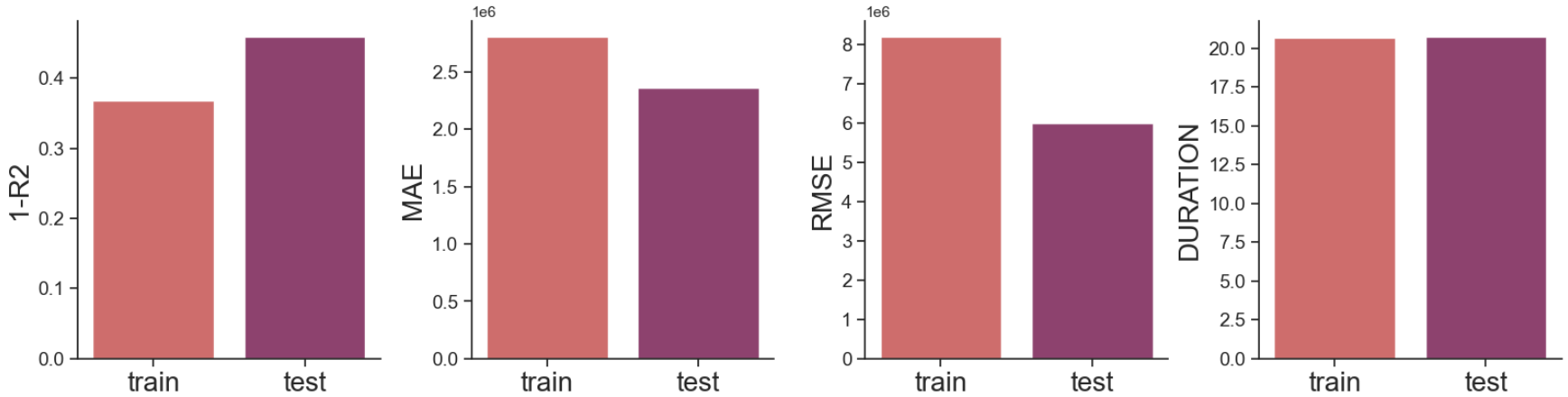
Modélisation 2 : Modèle linéaire | SVR linéaire après CV

SVR(C=100)



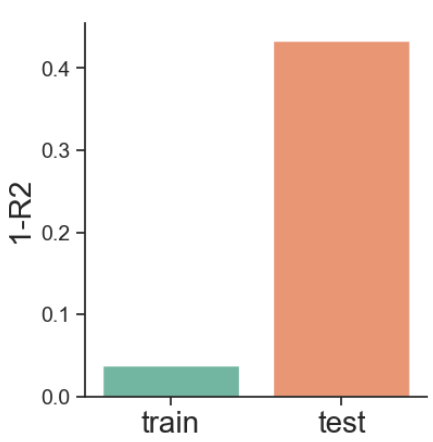
Avec transformation log

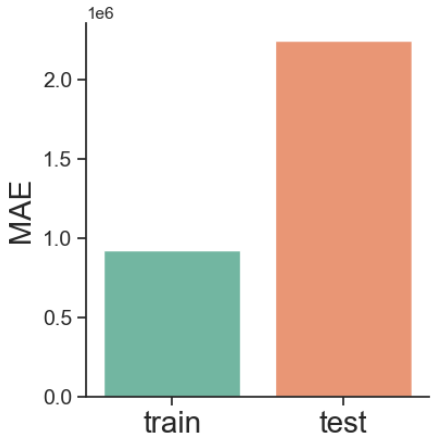
SVR(C=500)

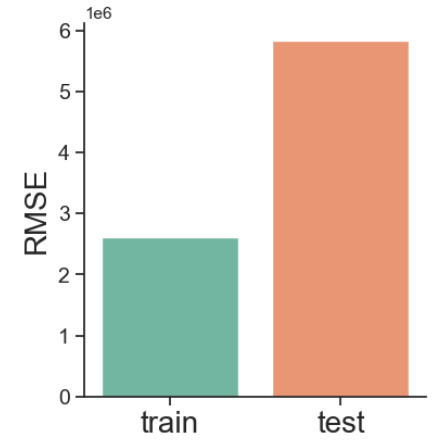


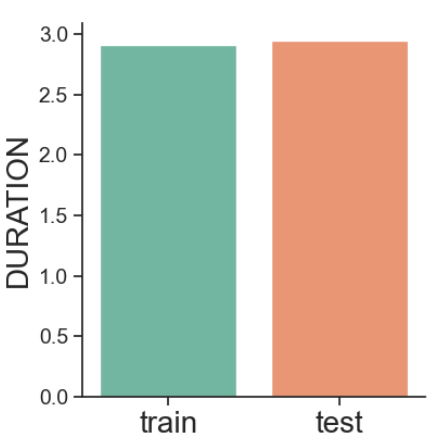
Modélisation 3 : Bagging | Random Forest après CV

RandomForestRegressor (n_estimators=400)



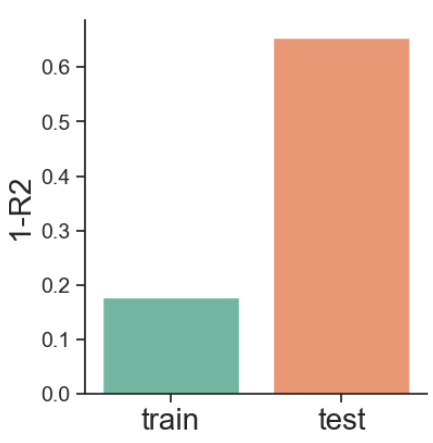


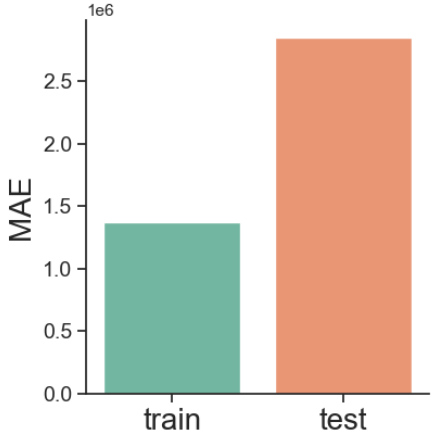


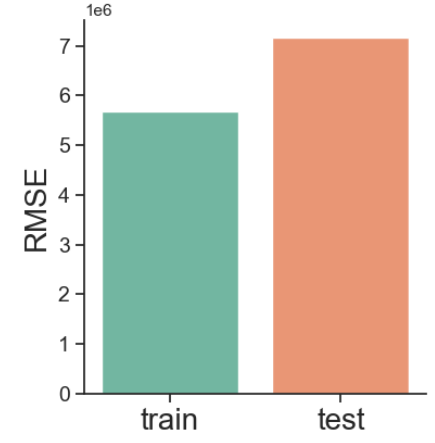


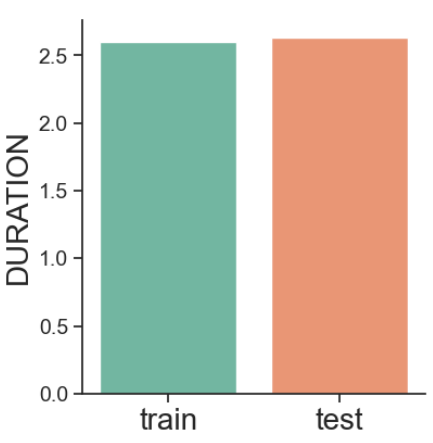
Avec transformation log

RandomForestRegressor (n_estimators=400)



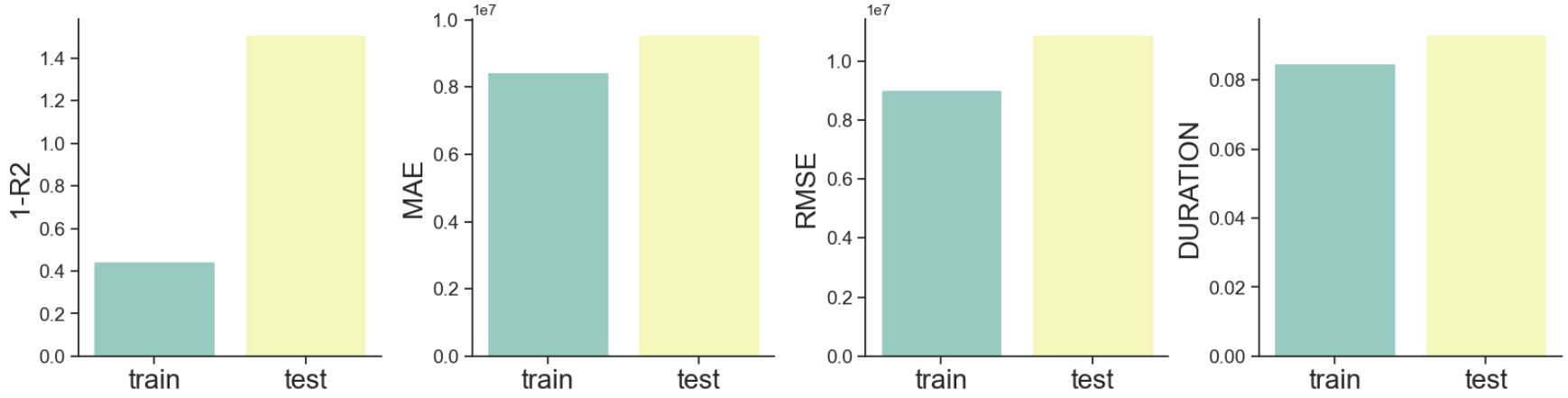






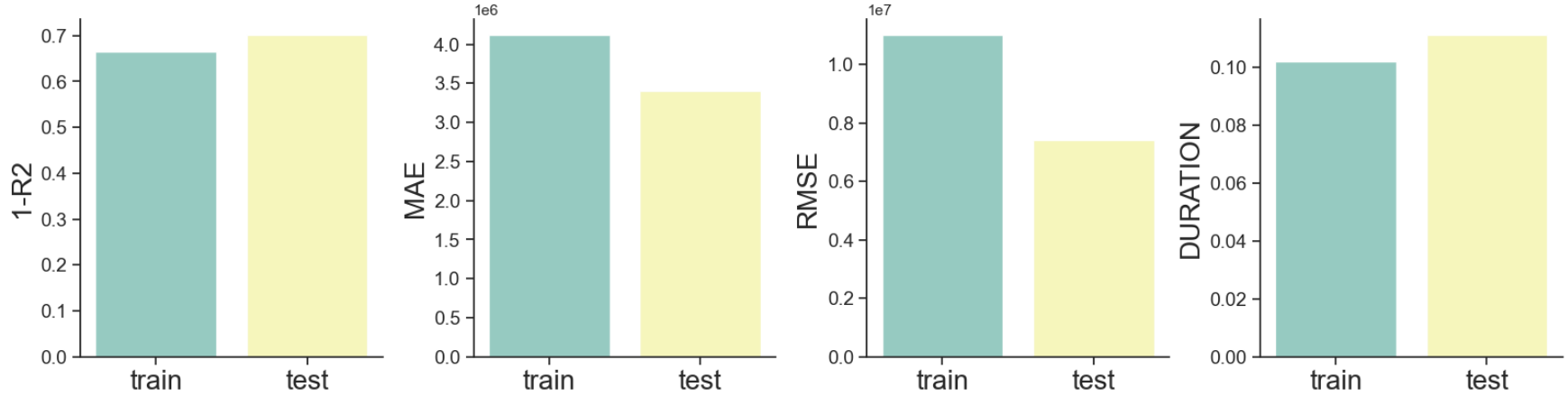
Modélisation 3 : Boosting| GBoost, après cv

AdaBoostRegressor
(n_estimators=100)



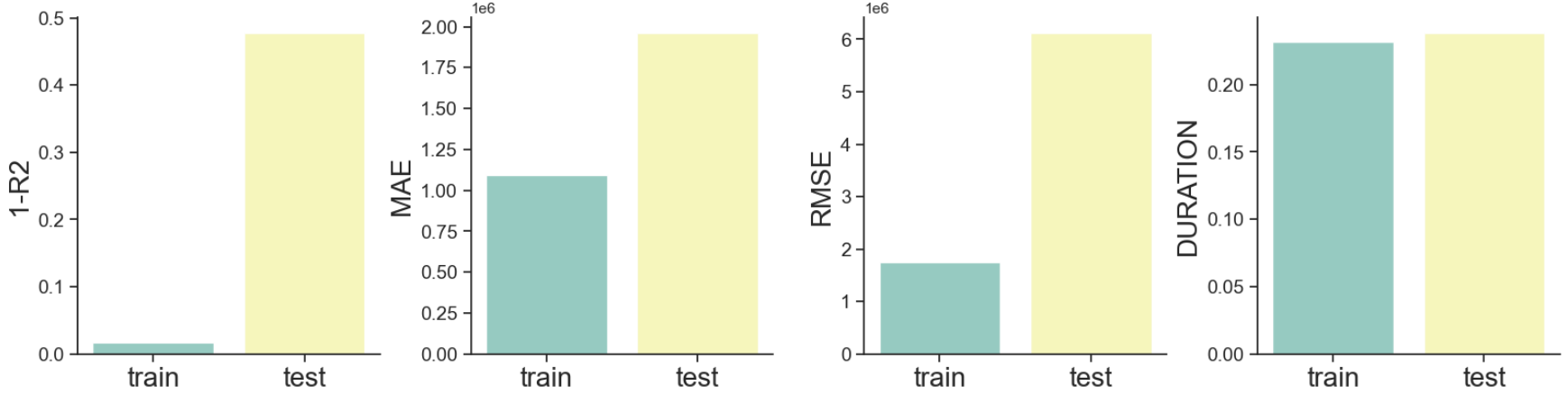
Avec transformation log

AdaBoostRegressor
(n_estimators=400)



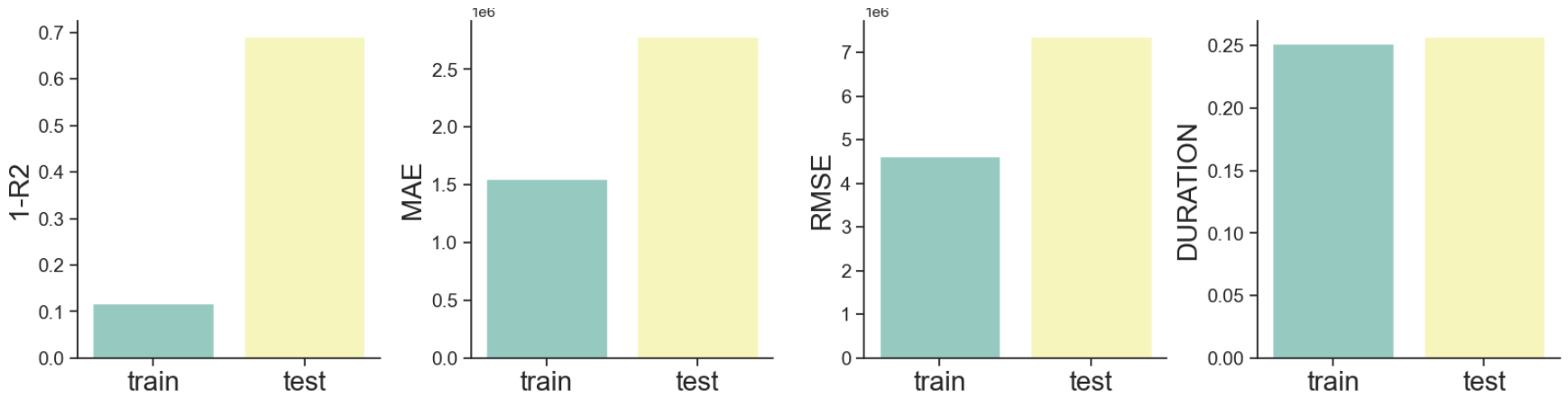
Modélisation 3 : Boosting| GBoost, après cv

GradientBoostingr()
{'n_estimators': 100}



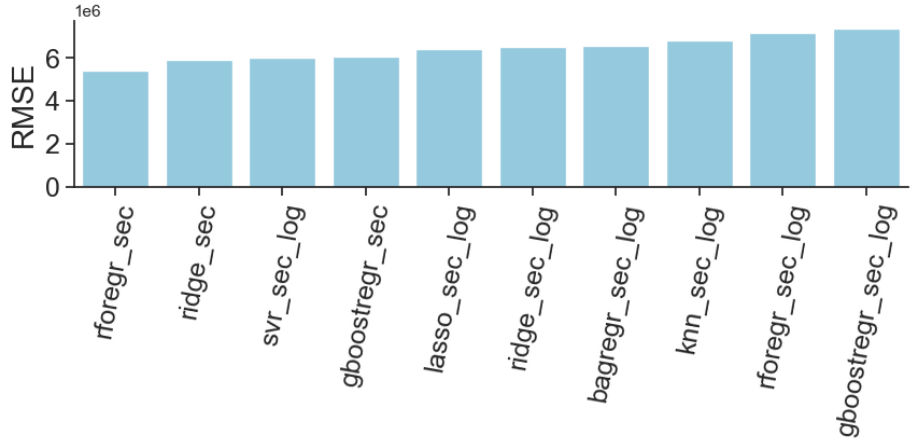
Avec transformation log

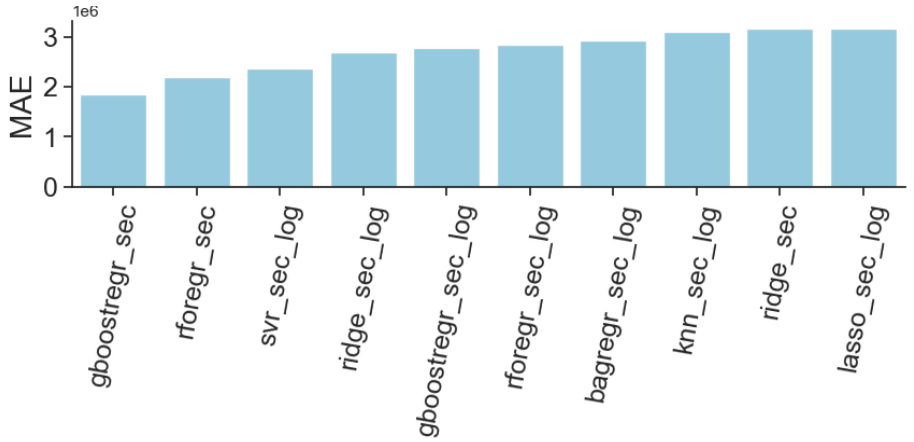
GradientBoostingRegressor()
{'n_estimators': 100}

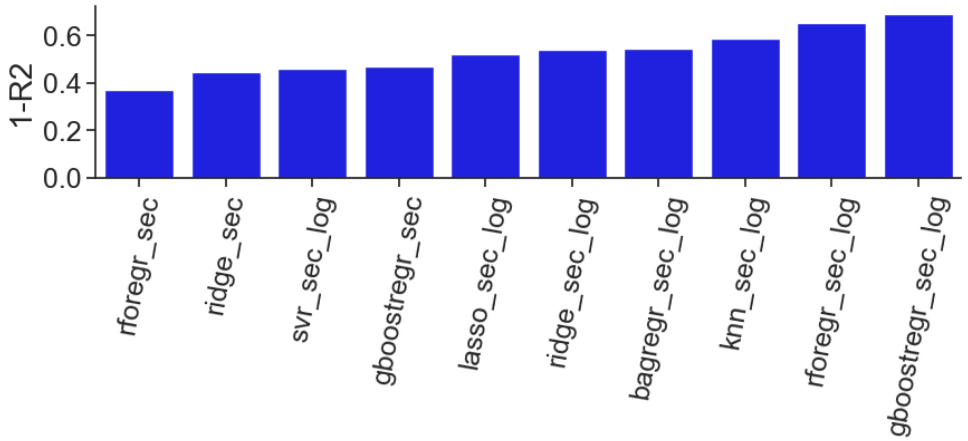


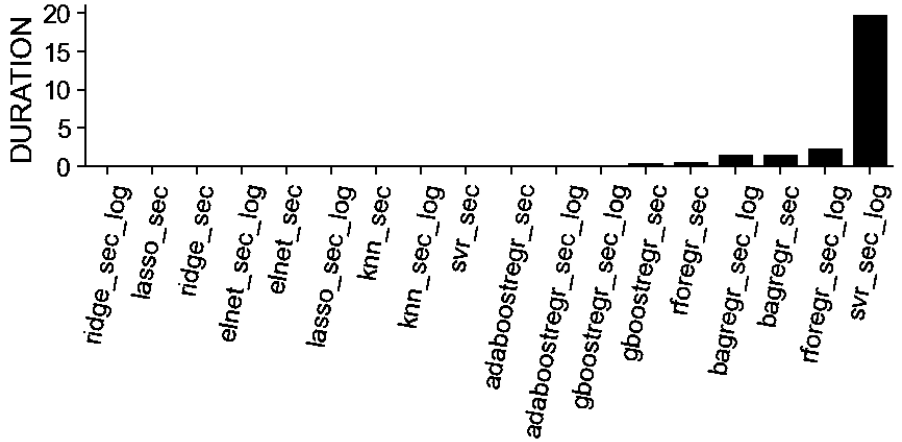
Conclusions : Comparaison modèles sur le jeu test

- *SiteEnergyUseWN(kBtu)*







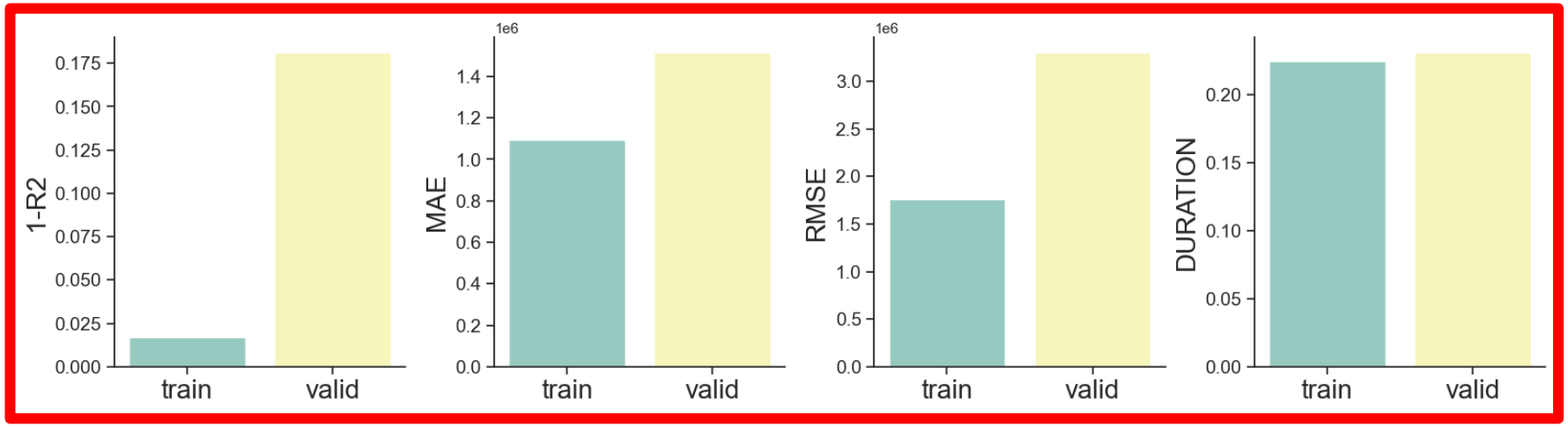


➢ **GradientBoostingRegressor ou randomforest ?**

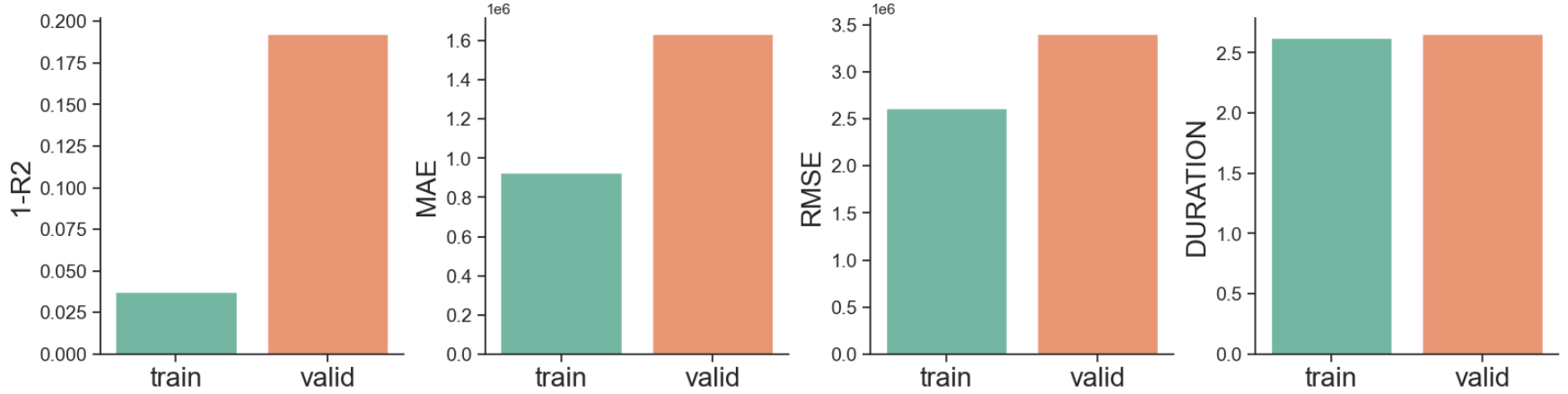
Conclusions : Comparaison modèles sur le jeu validation

- *SiteEnergyUseWN(kBtu)*

GradientBoostingRegressor()
{'n_estimators': 100}



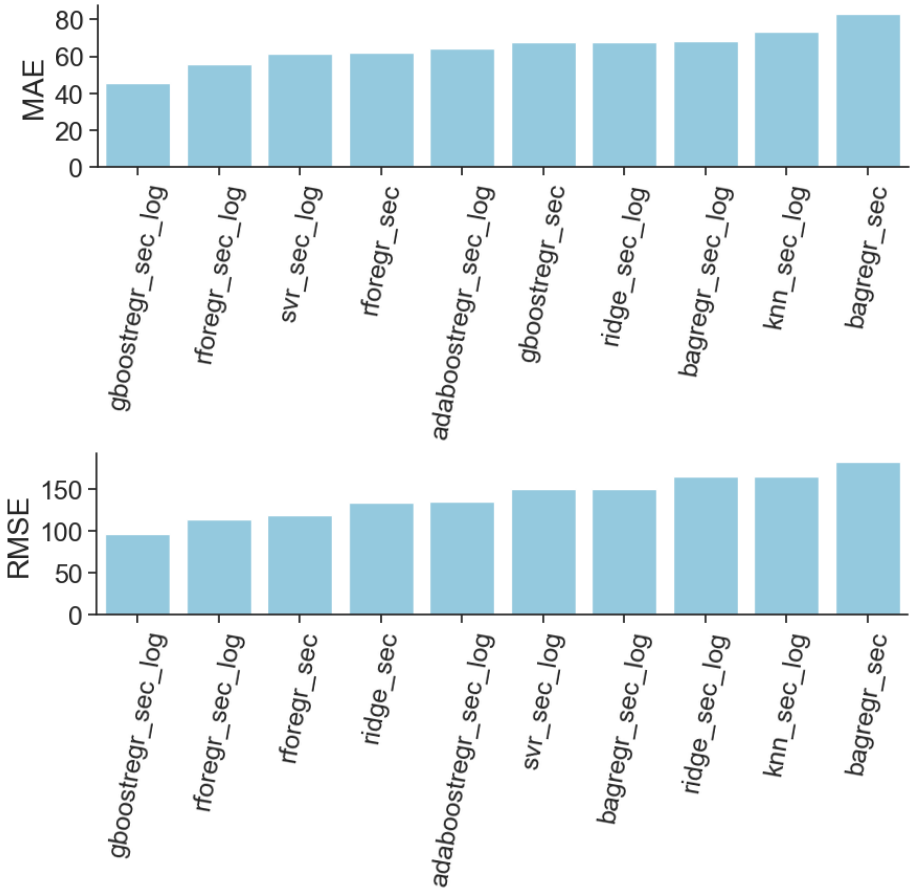
RandomForestRegressor
(n_estimators=400)

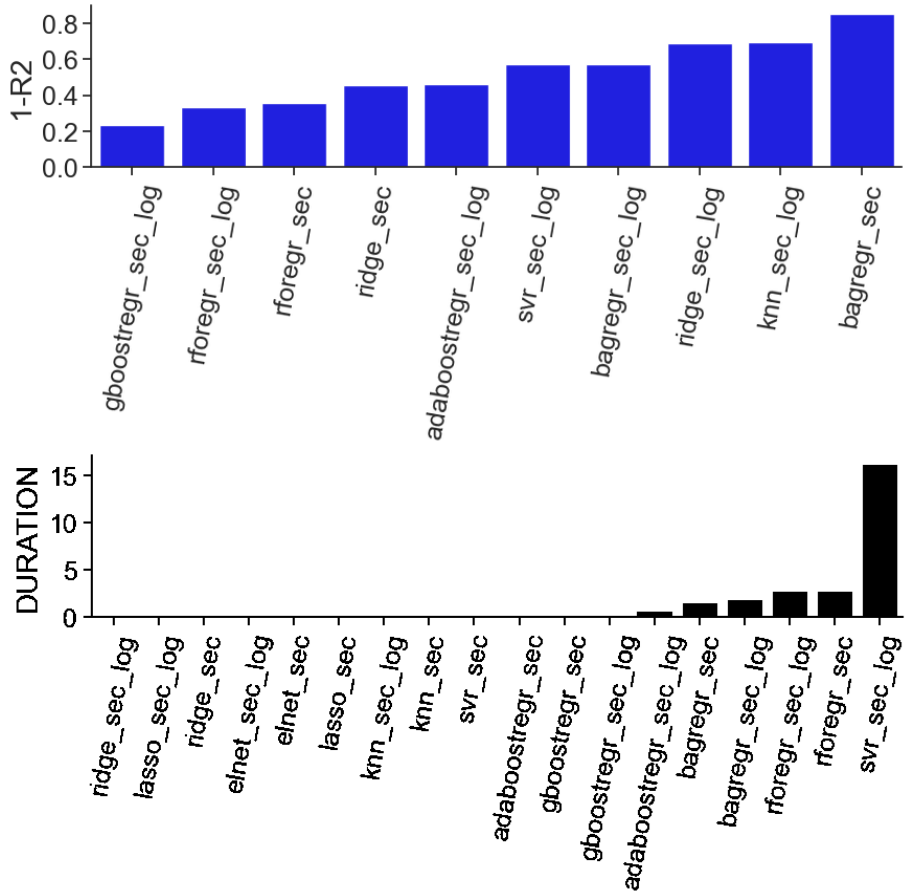


➢ **GradientBoostingRegressor est le meilleur modèle pour prédire la consommation totale d'énergie des bâtiments.**

Conclusions : Comparaison modèles

- *TotalGHGEmissions*

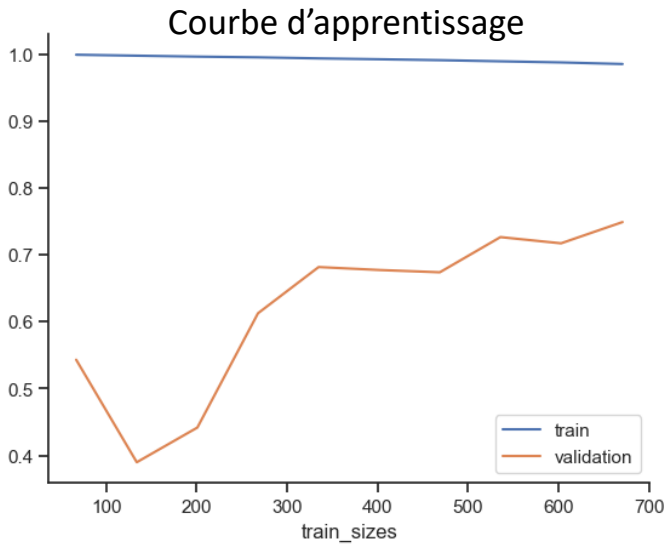


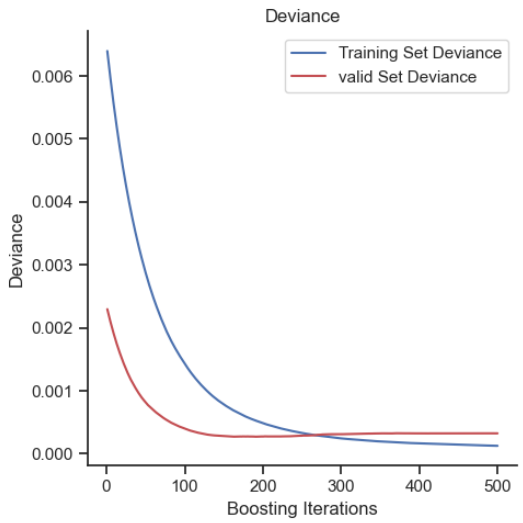


➢ **GradientBoostingRegressor est le meilleur modèle pour prédire les émissions de CO2 des bâtiments.**
(avec transformation log des features et de la cible)

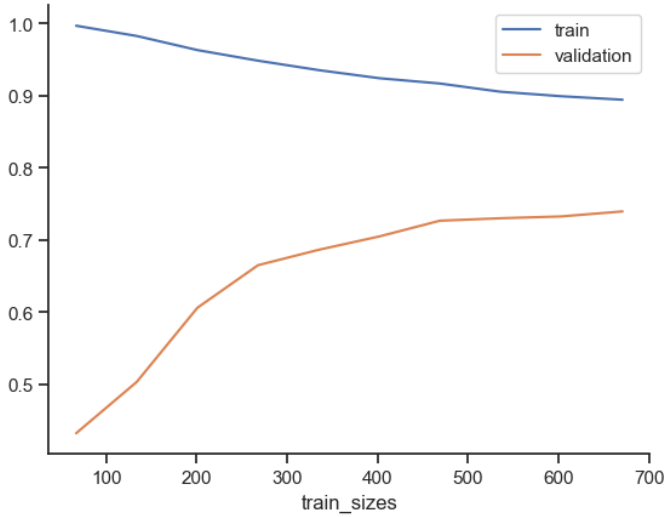
Conclusions : Le meilleur modèle, GradientBoostingRegressor

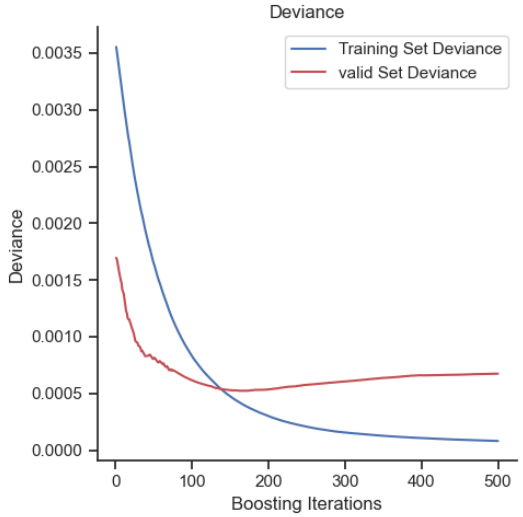
- *SiteEnergyUseWN(kBtu)*





- *TotalGHGEmissions*



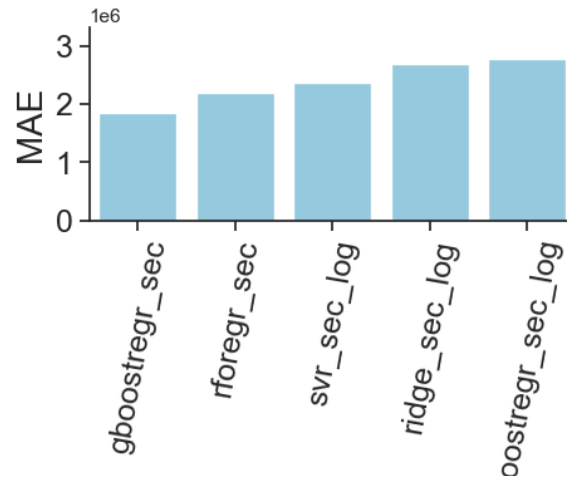


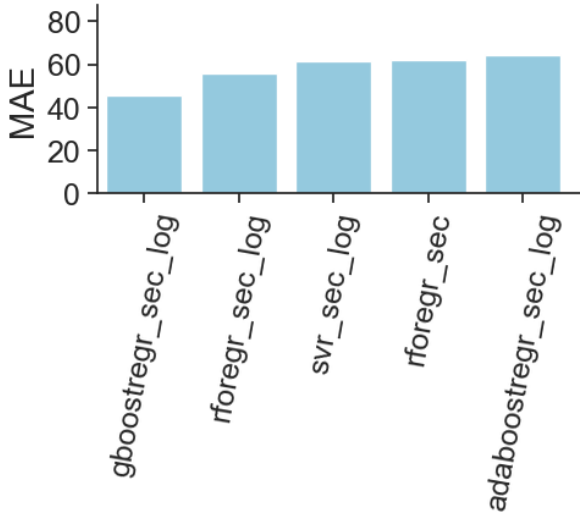
- **Les modèles pourraient être optimisés avec un nombre plus important de données.**

Conclusions : Performances des modèles sans ENERGYSTARScore

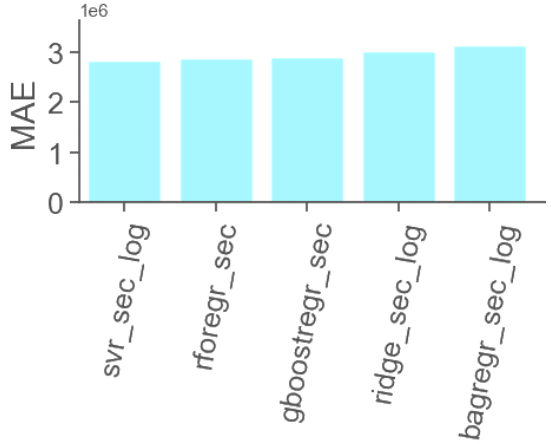
- *SiteEnergyUseWN(kBtu)*
- *TotalGHGEmissions*

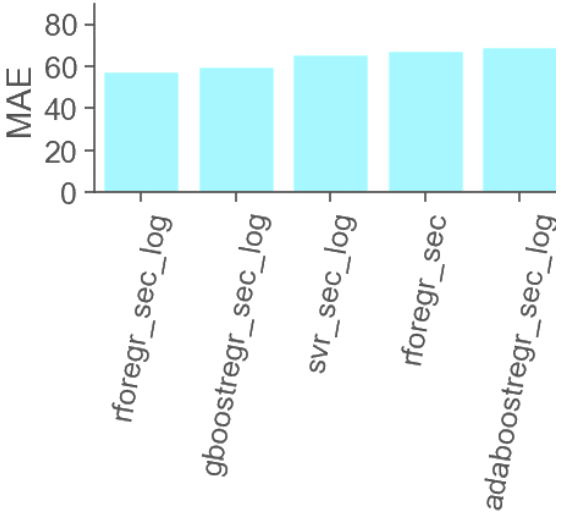
AVEC ENERGYSTARScore





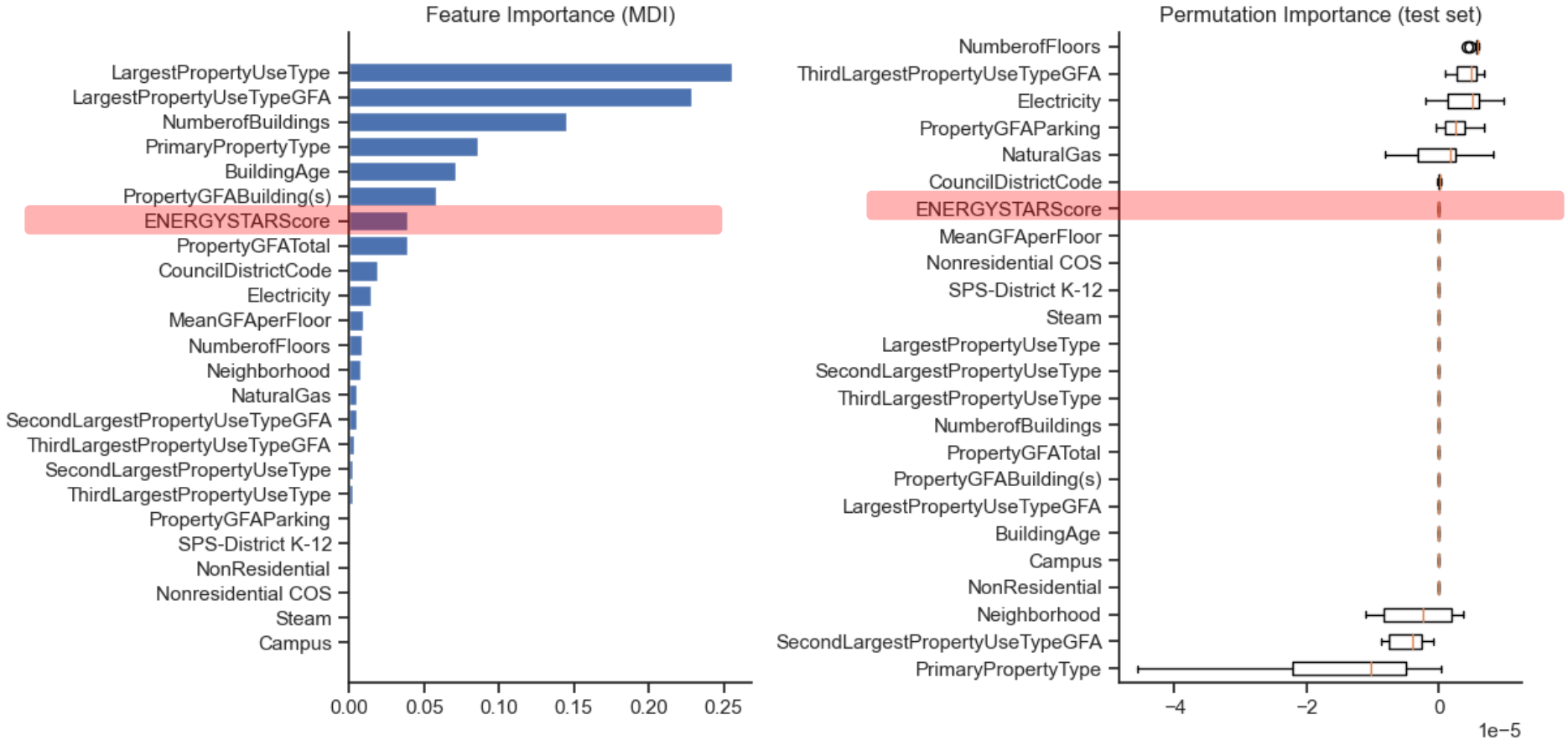
SANS ENERGYSTARScore





Conclusions : Features importances ENERGYSTARScore pour la prédiction d'émissions

- *TotalGHGEmissions*



➢ **Malgré son impact positif sur les prédictions, ENERGYSTARScore a une importance moyenne/assez relative dans le classement**

Conclusion générale

Missions confiées par la ville de Seattle:

- ✓ **Prédire la consommation totale d'énergie de ces bâtiments.**
- ✓ **Prédire les émissions de CO2 des bâtiments hors habitations.**
- ✓ **Evaluer l'intérêt de l'"ENERGYSTARScore" pour la prédiction d'émissions.**

Prédiction de la consommation énergétique

Site Energy Use (SEU) modélisé par 4 grands types de modèles différents.

Résultat optimal obtenu avec GradientBoostingRegressor

Total GreenHouse Gases (GHG) corrèle fortement avec SEU, et effectivement nous retrouvons les mêmes comportements lors de la modélisation.

Intérêt de la variable EnergySTARScore

Elle améliore systématiquement les performance des modèles.

Importance intermédiaire/relative toutefois dans le classement global des variables.

Evaluation des performances du modèle

Les modèles pourraient être optimisés avec un nombre plus important de données.

Merci pour votre attention

ANNEXES

AVEC

- *SiteEnergyUseWN(kBtu)*

	Modele	Data	1-R2	MAE	RMSE	DURATION
1	gboostregr_sec	test	0.48	1958828.48	6112619.57	0.24
1	rforegr_sec	test	0.43	2246625.26	5824940.84	2.89
1	svr_sec_log	test	0.46	2365070.96	5992877.76	20.75
1	ridge_sec_log	test	0.54	2688168.74	6510954.81	0.01
1	gboostregr_sec_log	test	0.69	2780256.01	7355205.74	0.23
1	rforegr_sec_log	test	0.65	2844430.88	7157408.64	2.58
1	bagregr_sec_log	test	0.55	2936045.60	6536997.77	1.62
1	knn_sec_log	test	0.59	3094794.78	6788227.24	0.03
1	ridge_sec	test	0.45	3157783.33	5918164.37	0.01
1	lasso_sec_log	test	0.52	3160320.33	6405679.17	0.01
1	adaboostregr_sec_log	test	0.70	3398035.41	7409787.95	0.09
1	bagregr_sec	test	0.74	3542222.21	7606234.80	1.61
1	knn_sec	test	0.80	3769719.98	7931221.63	0.04
1	elnet_sec_log	test	0.90	3811849.21	8392983.94	0.01
1	lasso_sec	test	0.97	5417444.56	8737292.26	0.01
1	elnet_sec	test	0.81	5565716.66	7948774.92	0.01
1	svr_sec	test	1.06	6542260.44	9093828.79	0.11
1	adaboostregr_sec	test	1.51	9541547.65	10871831.35	0.11

- *TotalGHGEmissions*

	Modele	Data	1-R2	MAE	RMSE	DURATION
1	gboostregr_sec_log	test	0.24	45.60	95.89	0.23
1	rforegr_sec_log	test	0.33	55.77	113.70	2.73
1	svr_sec_log	test	0.57	61.54	149.75	18.00
1	rforegr_sec	test	0.36	62.31	118.29	3.00
1	adaboostregr_sec_log	test	0.46	64.36	134.32	0.65
1	gboostregr_sec	test	1.13	67.80	209.77	0.21
1	ridge_sec_log	test	0.69	67.91	164.18	0.02
1	bagregr_sec_log	test	0.57	68.08	149.94	1.88
1	knn_sec_log	test	0.69	73.33	164.51	0.03
1	bagregr_sec	test	0.85	83.36	182.84	1.60
1	lasso_sec_log	test	1.01	85.27	198.42	0.01
1	elnet_sec_log	test	1.01	86.52	198.80	0.01
1	knn_sec	test	0.97	89.05	195.24	0.03
1	ridge_sec	test	0.46	93.12	133.74	0.01
1	lasso_sec	test	1.04	104.27	201.84	0.01
1	elnet_sec	test	1.03	131.13	200.63	0.01
1	svr_sec	test	1.39	184.90	233.47	0.09
1	adaboostregr_sec	test	1.89	237.64	271.71	0.14

SANS

- *SiteEnergyUseWN(kBtu)*

	Modele	Data	1-R2	MAE	RMSE	DURATION
1	svr_sec_log	test	0.51	2824688.68	6303415.78	20.31
1	rforegr_sec	test	0.57	2878932.44	6687929.52	2.88
1	gboostregr_sec	test	0.54	2905201.07	6514011.07	0.24
1	ridge_sec_log	test	0.61	3011237.62	6889252.54	0.01
1	bagregr_sec_log	test	0.60	3144668.22	6856734.77	1.69
1	lasso_sec_log	test	0.52	3160320.33	6405679.17	0.01
1	adaboostregr_sec_log	test	0.62	3178355.79	6952007.65	0.04
1	rforegr_sec_log	test	0.74	3209309.30	7629158.17	2.89
1	knn_sec_log	test	0.62	3253755.96	6973296.06	0.03
1	ridge_sec	test	0.50	3455155.80	6249659.28	0.01
1	elnet_sec_log	test	0.90	3811849.21	8392983.94	0.01
1	bagregr_sec	test	0.83	3927821.13	8053100.58	1.98
1	knn_sec	test	0.87	3983850.81	8244331.84	0.03
1	gboostregr_sec_log	test	3.10	4190945.62	15578872.08	0.21
1	lasso_sec	test	0.97	5417444.56	8737292.26	0.01
1	elnet_sec	test	0.81	5565716.66	7948774.92	0.01
1	svr_sec	test	1.25	7626652.14	9898173.32	0.10
1	adaboostregr_sec	test	1.79	10386445.85	11856569.68	0.12

- *TotalGHGEmissions*

	Modele	Data	1-R2	MAE	RMSE	DURATION
1	rforegr_sec_log	test	0.39	57.67	123.50	2.32
1	gboostregr_sec_log	test	0.54	59.82	145.22	0.21
1	svr_sec_log	test	0.61	65.63	154.66	11.26
1	rforegr_sec	test	0.40	67.56	124.64	2.65
1	adaboostregr_sec_log	test	0.53	69.08	144.13	0.10
1	gboostregr_sec	test	0.72	69.44	167.68	0.21
1	bagregr_sec_log	test	0.60	70.20	153.52	1.70
1	knn_sec_log	test	0.67	73.37	161.87	0.03
1	ridge_sec_log	test	0.77	73.90	173.28	0.01
1	lasso_sec_log	test	1.01	85.27	198.42	0.01
1	elnet_sec_log	test	1.01	86.52	198.80	0.01
1	bagregr_sec	test	0.93	90.05	190.66	1.77
1	knn_sec	test	1.01	91.81	199.15	0.03
1	ridge_sec	test	0.55	98.52	147.26	0.01
1	lasso_sec	test	1.04	104.27	201.84	0.01
1	elnet_sec	test	1.03	131.13	200.63	0.01
1	adaboostregr_sec	test	1.71	216.03	258.67	0.15
1	svr_sec	test	2.15	232.60	289.62	0.09