

06/10/2023

Anticipez les besoins en consommation de bâtiments

Armand FAUGERE [LinkedIn](#)

armand-faugere@live.fr

Sommaire

- I) Cadrage du projet et données d'entrée
- II) Traitement et analyse des données
- III) Modélisations et Prédictions
- IV) Choix de l'algorithme
- V) Interprétation
- VI) Conclusion



I) Cadrage du projet et données d'entrée

I) Cadrage du projet et données d'entrée



- ❑ **Contexte** : Projet de prédiction des émissions de CO2 et de consommation totale d'énergie des bâtiments non destinés à l'habitation de la ville de Seattle.

Relevés minutieux par des agents en 2016, très coûteux et fastidieux à réaliser.

ENERGY STAR Score fastidieux à calculer.



❑ **But :**

- Créer un système de prédiction **des émissions de CO2 et de consommation totale d'énergie** des bâtiments non destinés à l'habitation de la ville de Seattle en se basant sur les données structurelles des bâtiments.

❑ **Objectifs :**

- Traiter et explorer le jeu de donnée
- Réaliser des modélisations prédictives
- Evaluer les modèles
- Se passer des relevés de consommation annuels et futurs
- Eviter le Data leakage

- ❑ **Le jeu de données** <https://data.seattle.gov/dataset/2016-Building-Energy-Benchmarking/2bpz-gwpy>

- Relevé détaillé des consommations des bâtiments pour l'année 2016

3376 bâtiments, 46 colonnes descriptives

- Descriptif des champs du jeu de donnée

❑ **ENERGY STAR Score**

https://www.energystar.gov/buildings/benchmark/analyze_building_benchmarking_results

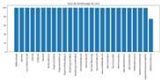
II) Traitement et analyse des données

2) Traitement et analyse des données

Jupyter Notebook, Python, *Pandas, Numpy, Matplotlib, Seaborn, sklearn*

1 Prétraitement du jeu de données

46 colonnes
→
33 colonnes

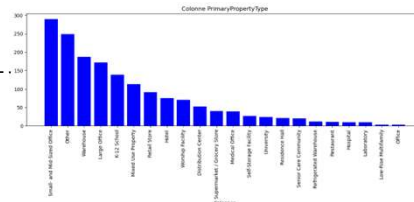
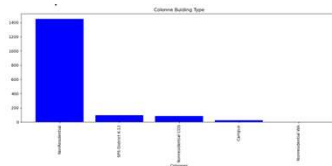


2 Analyse univariée

Analyse des 7 variables catégorielles



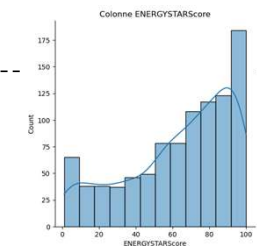
- ☐ Analyse barplot et `describe(max, min, moy)`
- ☐ Traitement des valeurs
 - *Filtre sur valeurs compliant*
 - *Suppression colonnes defaultData, OSEBuildingID, Latitude, Longitude*
 - *Suppression des doublons de colonne Neighborhood*
 - *Suppression des lignes NumberofBulding à 0*
 - *Suppression des lignes Numberoffloors à 0*



Analyse des 26 variables numériques



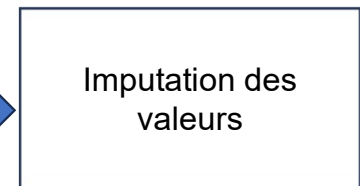
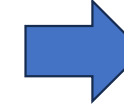
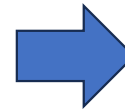
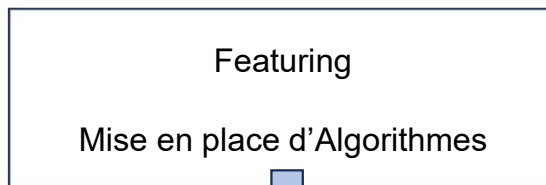
- ☐ Analyse boxplot, `describe(max, min, moy)`, courbe densité
- ☐ Traitement des valeurs
 - *Suppression 1 valeur extrêmes SiteEnergyUse(Kbtu)*
 - *Suppression 1 valeur Electricity(Kbtu) négative*



2) Traitement et analyse des données



Jupyter Notebook, Python, *Pandas, Numpy, Matplotlib, Seaborn, sklearn*

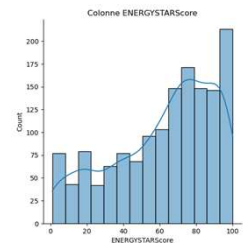
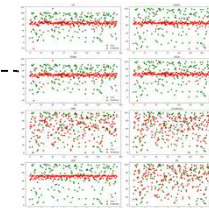
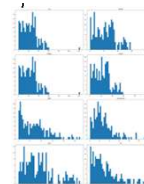


3 Traitement des NAN ENERGY STAR SCORE



- ❑ *Label Encoder : LargestPropertyUseType*
- ❑ *Normalisation : LargestPropertyUseTypeGFA, SourceEUIWN(kBtu)*
- ❑ *Création de modèles sur la base des infos du site de l'ENERGYSTARSCORE*
→ *Conservation du meilleur modèle*
- ❑ *Training et test de plusieurs algorithmes*
→ *LinearRegression*
→ *ElasticNet*
→ *Ridge*
→ *Lasso*
→ *KNeighborsRegressor*
→ *SVR*
→ *RandomForestRegressor*

- ❑ *Mise en place des métriques de mesure (MAE, MDE, MSE, RMSE, R2, VARSCORE)*
- ❑ *Choix du KNN (RMSE :18, R2 : 0,57)*
- ❑ *Test comparatif modèle KNN avec Médiane et Moyenne*
- ❑ *Validation du modèle KNN ==> "LargestPropertyUseType" (LabelEncoder), 'LargestPropertyUseTypeGFA', 'SourceEUIWN(kBtu)'*



2) Traitement et analyse des données

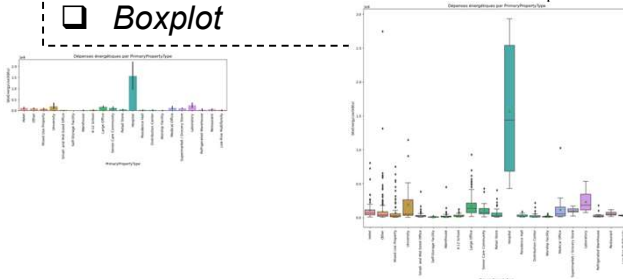


Jupyter Notebook, Python, Pandas, Numpy, Matplotlib, Seaborn, sklearn

4 Analyse Multivariée

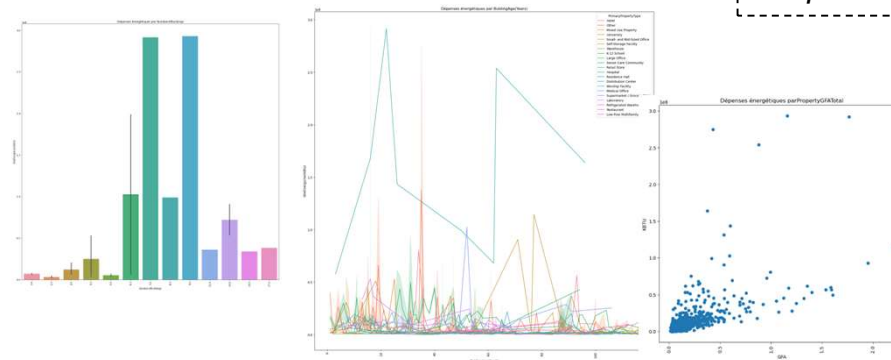
Analyse consommations
énergétiques et variables
catégorielles

- ☐ Barplot
- ☐ Boxplot



Analyse consommations
énergétiques et variables
numériques

- ☐ Création d'une colonne Age bâtiments
- ☐ Barplot
- ☐ Lineplot
- ☐ scatterplot



Pré-featuring

**Dataframe pour Machine
Learning**

- ☐ Création de la colonne CO2
- ☐ Suppression d'une valeur CO2 négative
- ☐ Suppression des colonnes pas utiles

III) Modélisations et Prédiction

3) Modélisations et Prédictions

Jupyter Notebook, Python, Pandas, Numpy, Matplotlib, Seaborn, Sklearn

Fonctions pour Machine Learning,
Métriques et graphiques

Importation Dataset
Check des colonnes
cibles

Modélisations

Training et test

Feat2

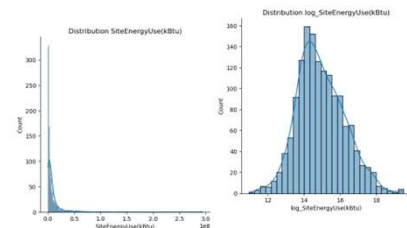
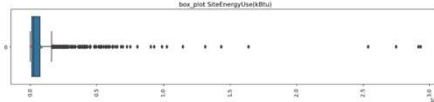
Feat3

1

Modélisations
et Tests

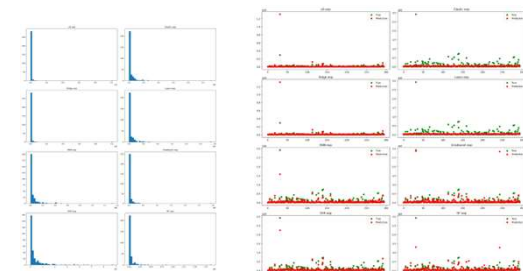
SiteEnergyUse
(Kbtu)

CO2(kg)



- ☐ Création de nouveaux indicateurs
- ☐ Encodage variables catégorielles (labelencoder et onehotencoder)
- ☐ Normalisation variables numériques
- ☐ Matrice de corrélation avec target
- ☐ Matrice de corrélation pour multicollinéarité
- ☐ Affinage du modèle avec matrice de corrélation + métriques favorables + indicateurs représentatifs

- ☐ Entrainement des modèles avec set de train et set de test (80 – 20)
- ☐ Sans transformation log target et avec transformation log



3) Modélisations et Prédictions



Nouvelles variables

Structurels :

- ☐ $(NB_FLOOR)/(NB_BUILDING)$
- ☐ $(GFA_P_USE)/(GFA_TOT)$
- ☐ $(GFA_PARKING)/(GFA_BUILDING)$
- ☐ $(GFA_BUILDING)/(GFA_TOT)$
- ☐ $(GFA_PARKING)/(GFA_TOT)$

proportion d'énergie :

- ☐ $ELEC/Energie_tot$
- ☐ $GAZ/Energie_tot$
- ☐ $STEAM/Energie_tot$

proportion CO2 :

- ☐ $ELEC/CO2_tot$
- ☐ $GAZ/CO2_tot$
- ☐ $STEAM/CO2_tot$

proportion Energie Building :

- ☐ $ELEC/Energie_tot ==> Bulding$
- ☐ $GAZ/Energie_tot ==> Bulding$
- ☐ $STEAM/Energie_tot ==> Bulding$

proportion Energie Parking :

- ☐ $ELEC/Energie_tot ==> Parking$
- ☐ $GAZ/Energie_tot ==> Parking$
- ☐ $STEAM/Energie_tot ==> Parking$

Dataleakage :

- ☐ $(GAZ)/(GFA_P_USE)$
- ☐ $(STEAM)/(GFA_P_USE)$
- ☐ $(ELEC)/(GFA_P_USE)$
- ☐ $(ELEC)/(GFA_TOT)$
- ☐ $(GAZ)/(GFA_TOT)$
- ☐ $(STEAM)/(GFA_TOT)$
- ☐ $(ELEC)/(GFA_BUILDING)$
- ☐ $(ELEC)/(GFA_PARKING)$
- ☐ $(GAZ)/(GFA_BUILDING)$
- ☐ $(GAZ)/(GFA_PARKING)$
- ☐ $(STEAM)/(GFA_BUILDING)$
- ☐ $(STEAM)/(GFA_PARKING)$

- Essais réalisés
- ➔ Corrélations très fortes avec cible
- ➔ R2 très élevé
- ➔ RMSE pas meilleur

3) Modélisations et Prédictions



SiteEnergyUSE(Kbtu)

CO2(Kg)

Feat 2 → Avec LabelEncoder

- ❑ 'LargestPropertyUseTypeGFA',
- ❑ 'NumberofFloors',
- ❑ '(GFA_P_USE)/(GFA_TOT)',
- ❑ 'NumberofBuildings',
- ❑ 'PropertyGFAParking',
- ❑ 'STEAM/Energie_tot ==> Bulding',
- ❑ 'BuldingAge(Years)',
- ❑ 'LargestPropertyUseType',
- ❑ 'GAZ/Energie_tot ==> Bulding',
- ❑ 'Neighborhood',
- ❑ 'STEAM/Energie_tot ==> Parking'

Feat 3 → Avec dummies variables

- ❑ 'LargestPropertyUseTypeGFA',
- ❑ 'LargestPropertyUseType_Hospital (General Medical & Surgical)',
- ❑ 'NumberofFloors',
- ❑ '(GFA_P_USE)/(GFA_TOT)',
- ❑ 'NumberofBuildings',
- ❑ 'LargestPropertyUseType_Data Center',
- ❑ 'PropertyGFAParking',
- ❑ 'PrimaryPropertyType_Large Office',
- ❑ 'STEAM/Energie_tot ==> Bulding',
- ❑ 'BuldingAge(Years)cat_(0.885, 29.75]',
- ❑ 'PrimaryPropertyType_Small- and Mid-Sized Office',
- ❑ 'LargestPropertyUseType_Non-Refrigerated Warehouse',
- ❑ 'CouncilDistrictCode_3',
- ❑ 'Neighborhood_DOWNTOWN'

Modèles
retenus pour
optimisation et
validation

Feat 2 → Avec LabelEncoder

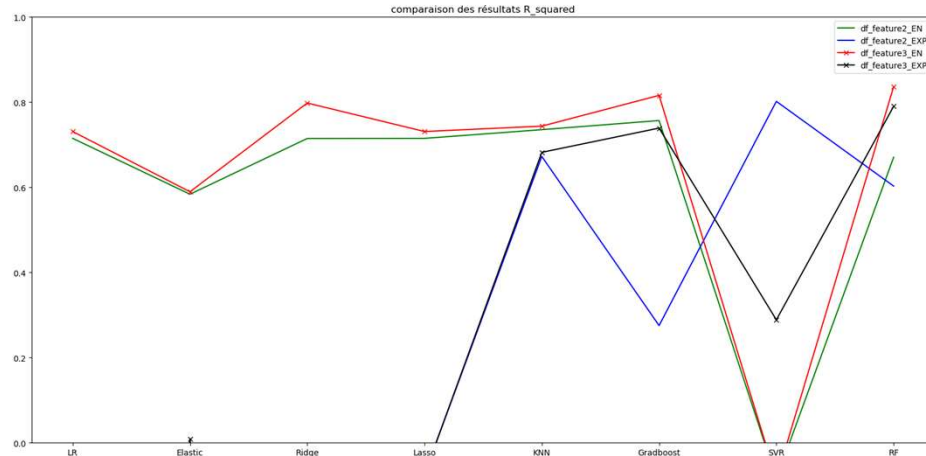
- ❑ 'LargestPropertyUseTypeGFA',
- ❑ 'NumberofBuildings',
- ❑ 'STEAM/Energie_tot ==> Bulding',
- ❑ '(GFA_P_USE)/(GFA_TOT)',
- ❑ 'NumberofFloors',
- ❑ 'ELEC/CO2_tot',
- ❑ 'LargestPropertyUseType',
- ❑ 'PropertyGFAParking',
- ❑ 'BuldingAge(Years)',
- ❑ 'STEAM/Energie_tot ==> Parking',
- ❑ 'GAZ/Energie_tot ==> Parking',
- ❑ 'Neighborhood'

Feat 3 → Avec dummies variables

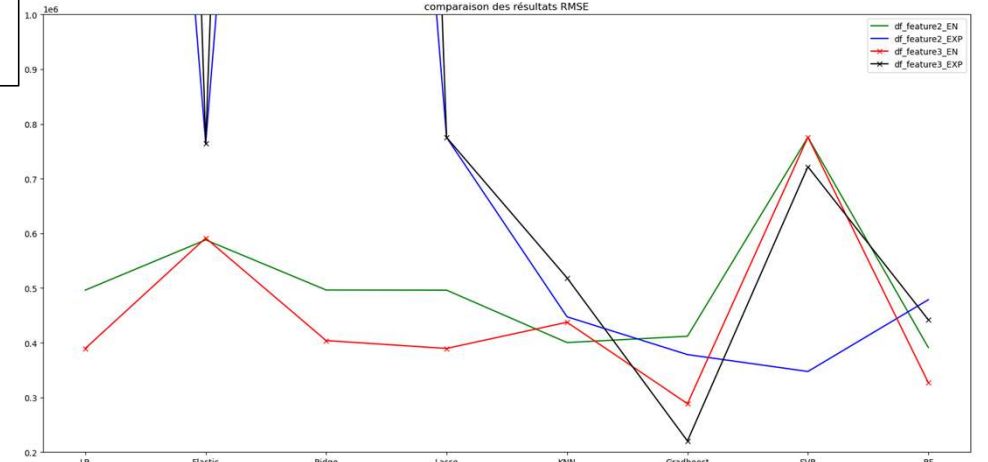
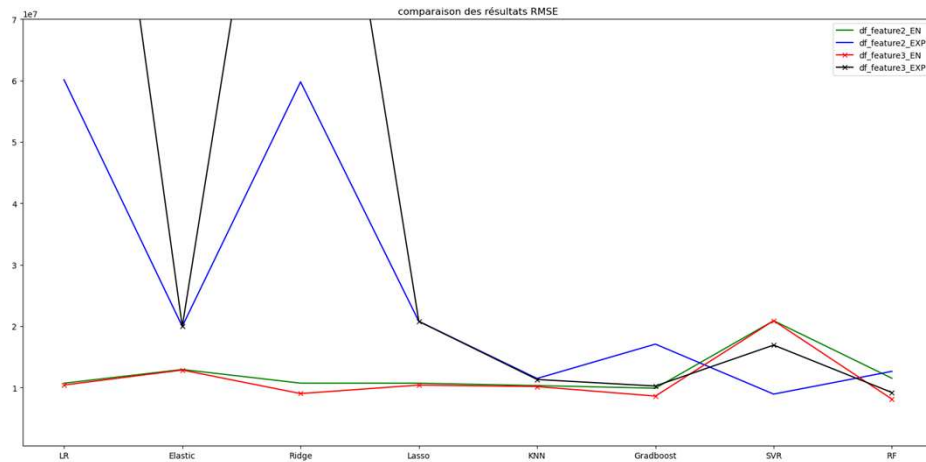
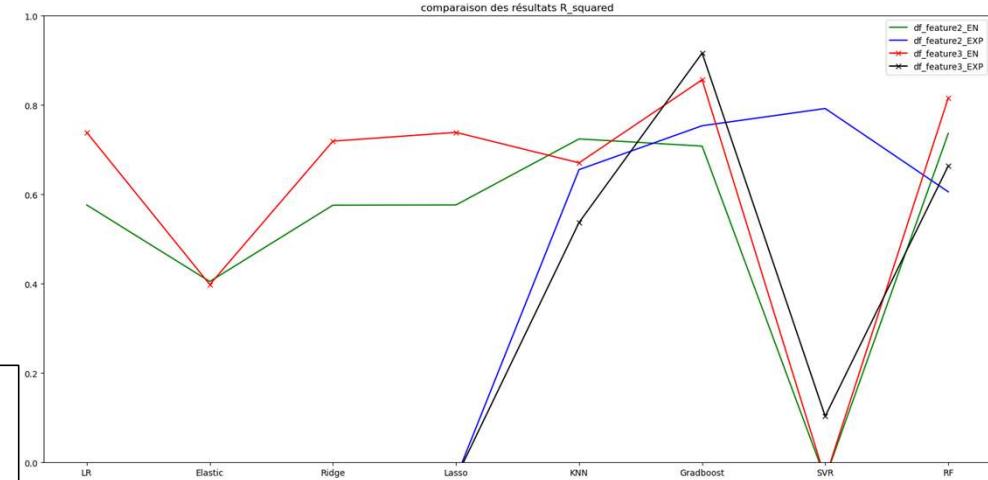
- ❑ 'PrimaryPropertyType_Hospital',
- ❑ 'LargestPropertyUseTypeGFA',
- ❑ 'NumberofBuildings',
- ❑ 'STEAM/Energie_tot ==> Bulding',
- ❑ '(GFA_P_USE)/(GFA_TOT)',
- ❑ 'NumberofFloors',
- ❑ 'Neighborhood_EAST',
- ❑ 'ELEC/Energie_tot',
- ❑ 'PrimaryPropertyType_Small- and Mid-Sized Office',
- ❑ 'CouncilDistrictCode_2',
- ❑ 'BuldingAge(Years)cat_(0.885, 29.75]'

3) Modélisations et Prédictions

SiteEnergyUSE(Kbtu)



CO2(Kg)



- ❑ Les modèles sans transformation log sont plus stables
- ❑ Le modèle feat 3 est plus performant

3) Modélisations et Prédictions

Jupyter Notebook, Python, Pandas, Numpy, Matplotlib, Seaborn, sklearn

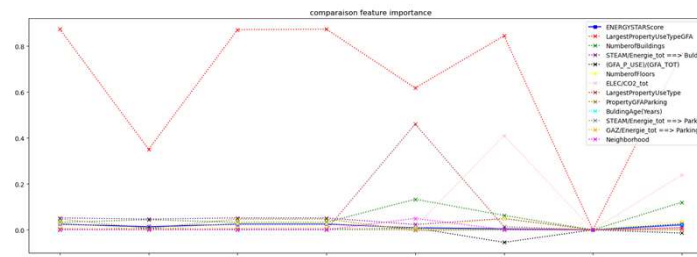
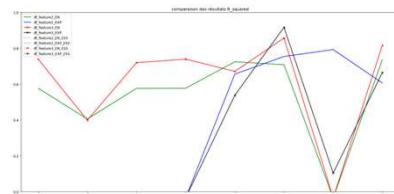
Rajout de
l'ENERGYSTARSORE dans le
modèle

Features Importance

**Pas d'impact
significatifs sur
les modèles**

- ☐ Rajout dans les modèles feat 2 et feat 3
- ☐ Training et test dans les mêmes conditions
- ☐ Comparatif des variations des métriques R2 et RMSE

- ☐ Evaluation de l'impact des variables sur la prédiction
- ☐ Comparaison sur graphiques



2
Pertinence
ENERGY
STARSORE

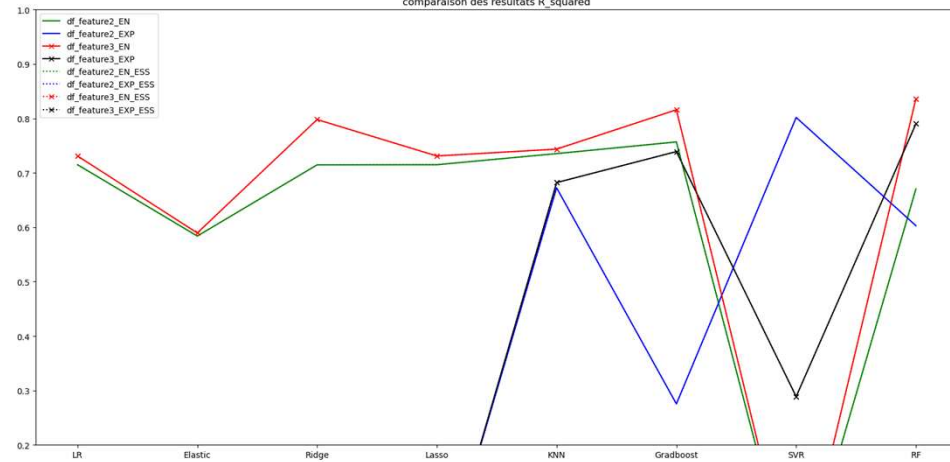
SiteEnergyUse
(Kbtu)

CO2(kg)

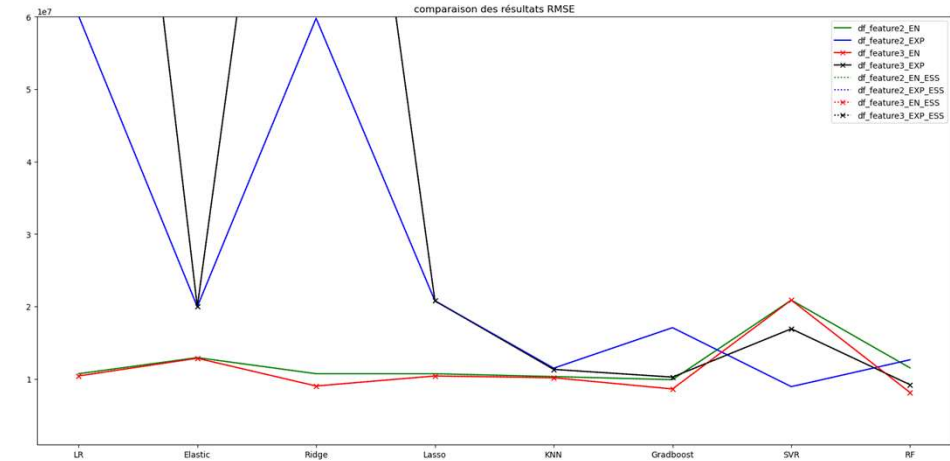
3) Modélisations et Prédictions

SiteEnergyUSE(Kbtu)

comparaison des résultats R_squared

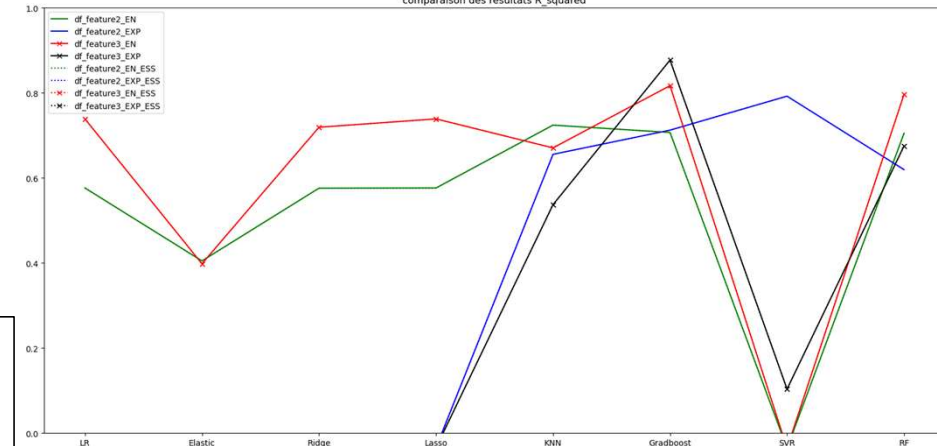


comparaison des résultats RMSE

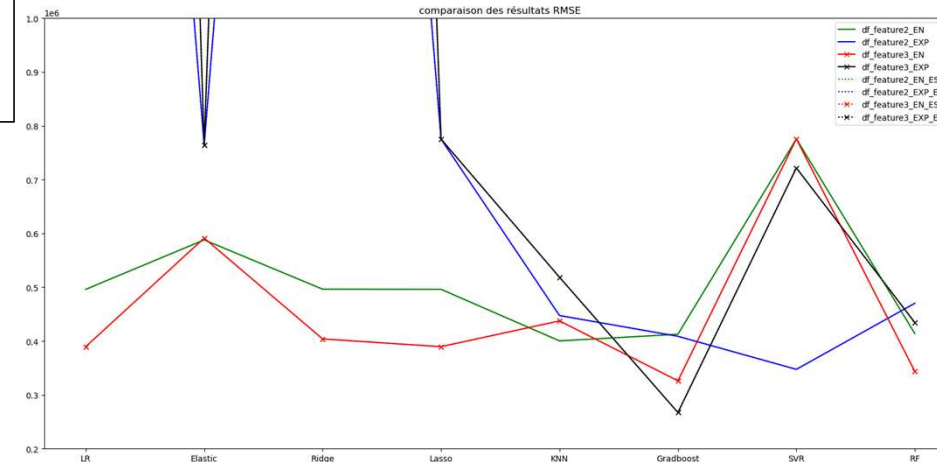


CO2(Kg)

comparaison des résultats R_squared



comparaison des résultats RMSE



❑ Le R2 n'est pas impacté par le rajout de l'ENERGYSTARSCORE

❑ Le RMSE n'est pas impacté par le rajout de l'ENERGYSTARSCORE

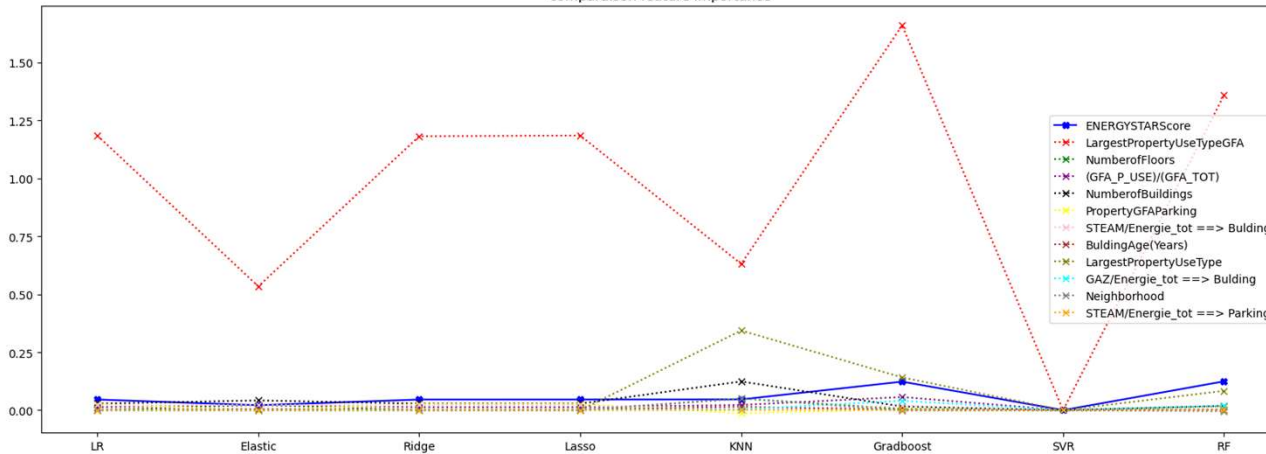
3) Modélisations et Prédictions

Feat 2

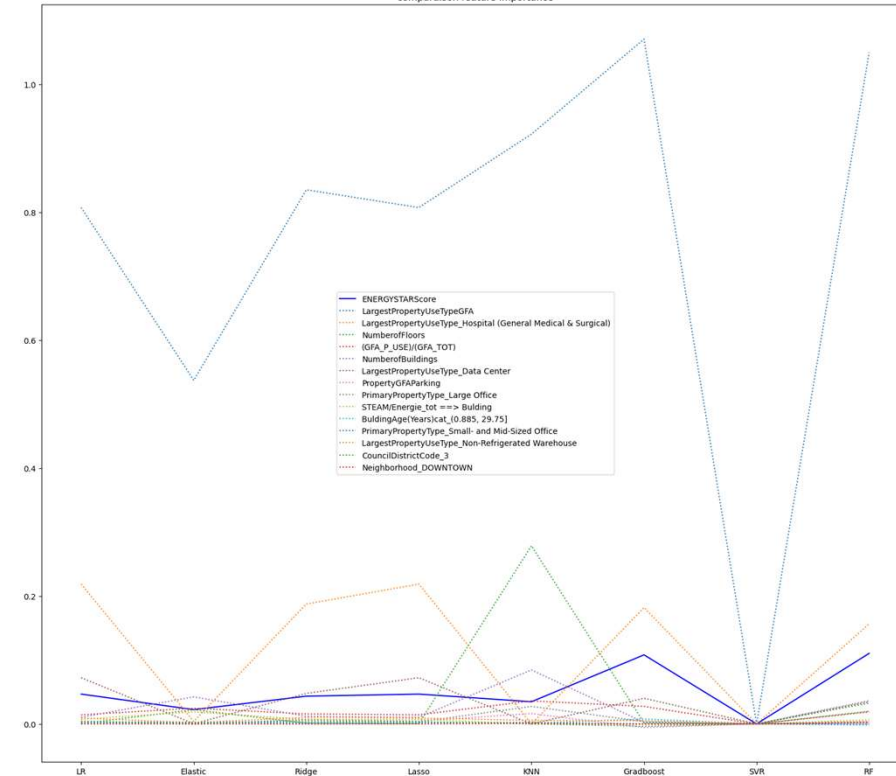
SiteEnergyUSE(Kbtu)

Feat 3

comparaison feature importance



comparaison feature importance



- ❑ La variable « LargestPropertyUsetypeGFA » est très importante
- ❑ D'autres variables se démarquent (type, nb de buildings...)
- ➔ L'ENERGYSTARSCORE a un impact limité sur la prédiction

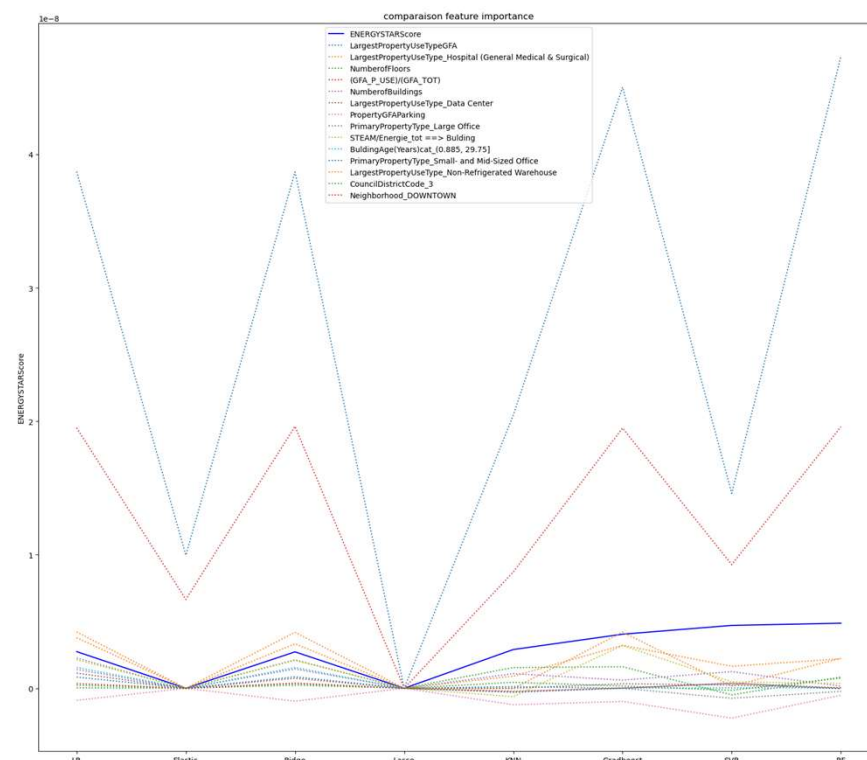
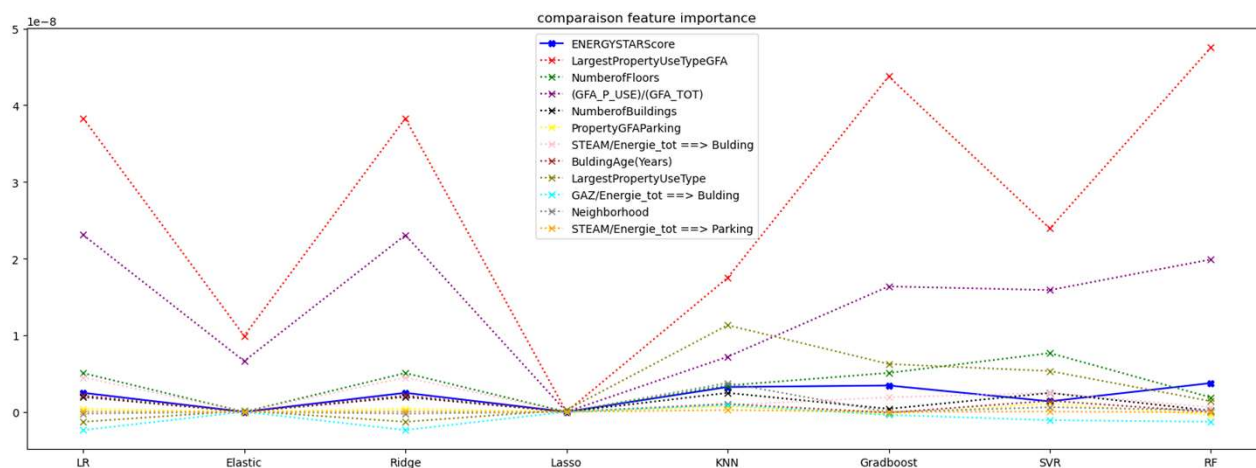
3) Modélisations et Prédictions



Feat 2
Transformation log

SiteEnergyUSE(Kbtu)

Feat 3
Transformation log



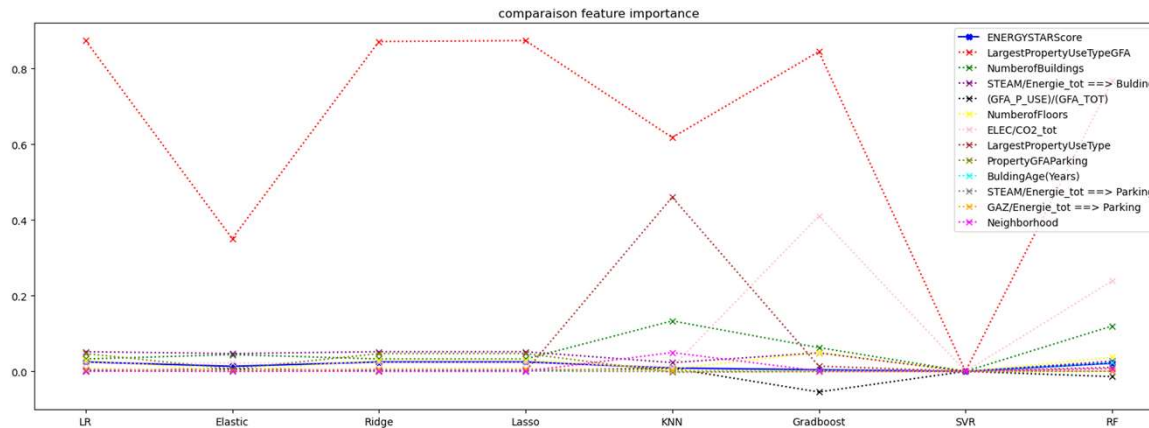
- ❑ La variable « LargestPropertyUsetypeGFA » est très importante
- ❑ D'autres variables se démarquent (type, nb de buildings...)
- ➔ L'ENERGYSTARSCORE a un impact limité sur la prédiction

3) Modélisations et Prédictions

Feat 2

CO2(Kg)

Feat 3



- ❑ La variable « LargestPropertyUsetypeGFA » est très importante
- ❑ D'autres variables se démarquent (type, nb de buildings...)
- ➔ L'ENERGYSTARSCORE a un impact limité sur la prédiction

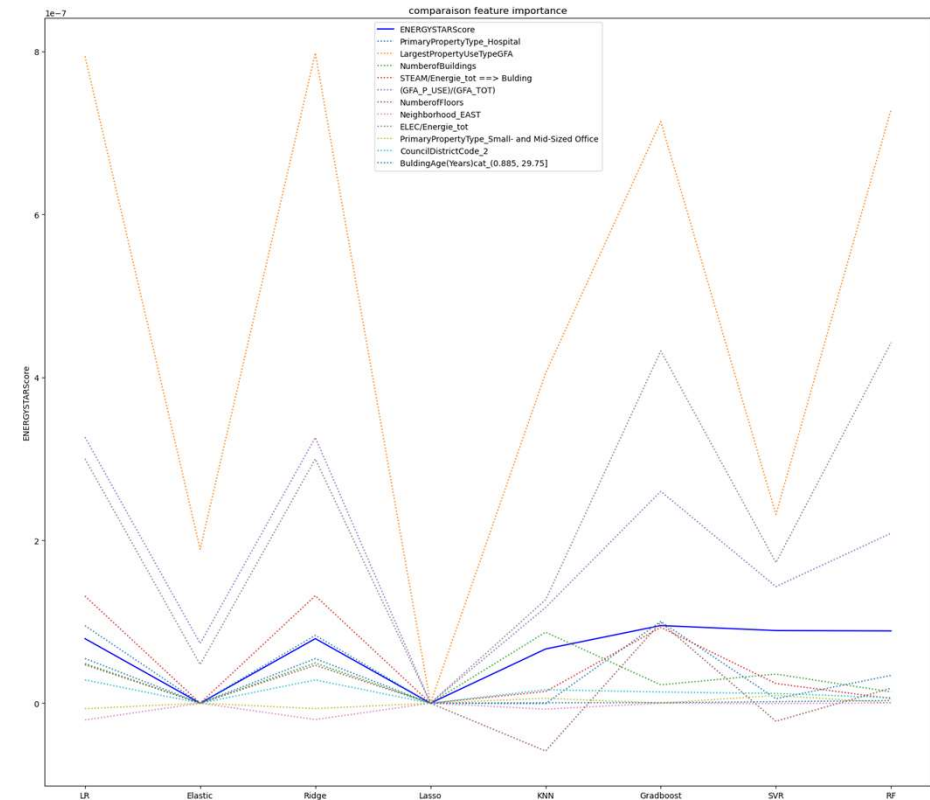
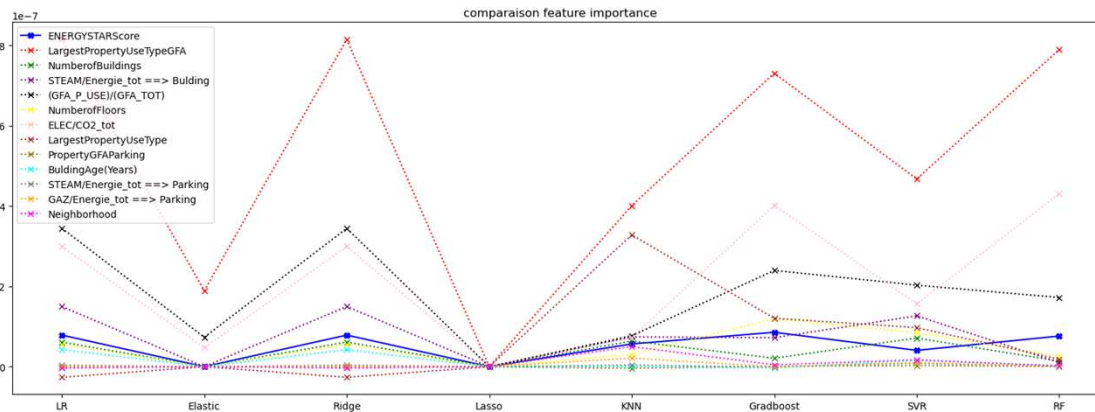


3) Modélisations et Prédictions

Feat 2
Transformation log

CO2(Kg)

Feat 3
Transformation log



- ☐ La variable « LargestPropertyUsetypeGFA » est très importante
- ☐ D'autres variables se démarquent (type, ratios d'energie...)
- ➔ L'ENERGYSTARSCORE a un impact limité sur la prédiction

3) Modélisations et Prédictions

Jupyter Notebook, Python, Pandas, Numpy, Matplotlib, Seaborn, sklearn

Fonction pour optimisation paramétrique

Validation croisée et optimisations
hyperparamétriques

Bilan des performances

**Grilles des
paramètres
optimisés**

- ☐ *Paramétrage Gridsearch (Holdout, ["neg_mean_squared_error", "r2"])*
- ☐ *Choix des paramètres à tester*
- ☐ *Entraînement et test*

- ☐ *Bilan des métriques*
- ☐ *Bilan des meilleurs paramètres*

3
**Optimisation
hyper
paramétrique**

SiteEnergyUse
(Kbtu)

CO2(kg)

3) Modélisations et Prédictions

Sans Transformation log

SiteEnergyUSE(Kbtu)

Avec Transformation log (unité de log)

	Feat2 R2	Feat3 R2	Feat2 RMSE	Feat3 RMSE
LR	0.550181	0.594310	1.330168e+07	1.149177e+07
ELN	0.551873	0.579688	1.336251e+07	1.306041e+07
RGE	0.550372	0.636346	1.330137e+07	1.169834e+07
LAS	0.550181	0.594310	1.330168e+07	1.149178e+07
KNN	0.247022	0.484180	1.630441e+07	1.435157e+07
GDB	0.554043	0.593493	1.344209e+07	1.281695e+07
SVR	-0.081985	-0.081722	2.033896e+07	2.033570e+07
RF	0.567130	0.604258	1.392568e+07	1.337679e+07

	Feat2 R2	Feat3 R2	Feat2 RMSE	Feat3 RMSE
LR	0.464010	0.504464	0.877750	0.839362
ELN	0.436899	0.455320	0.899525	0.883469
RGE	0.464225	0.504496	0.877566	0.839333
LAS	0.405671	0.409341	0.924628	0.921236
KNN	0.195660	0.453995	1.080240	0.884367
GDB	0.641248	0.549178	0.716220	0.798940
SVR	0.435966	0.492259	0.903117	0.846967
RF	0.571281	0.49630	0.779775	0.843626

	feat2	feat3
LR	{'fit_intercept': True}	{'fit_intercept': True}
ELN	{'alpha': 0.1}	{'alpha': 0.1}
RGE	{'alpha': 1.0}	{'alpha': 1.0}
LAS	{'alpha': 1.0}	{'alpha': 1.0}
KNN	{'leaf_size': 10, 'n_neighbors': 4}	{'leaf_size': 10, 'n_neighbors': 3}
GDB	{'learning_rate': 0.09999999999999999, 'n_estimators': 30}	{'learning_rate': 0.19, 'n_estimators': 30}
SVR	{'C': 1.4000000000000001, 'kernel': 'linear'}	{'C': 1.4000000000000001, 'kernel': 'poly'}
RF	{'min_samples_split': 2, 'n_estimators': 70}	{'min_samples_split': 2, 'n_estimators': 110}

	Feat 2	feat3
LR	{'fit_intercept': True}	{'fit_intercept': True}
ELN	{'alpha': 0.1}	{'alpha': 0.1}
RGE	{'alpha': 1.0}	{'alpha': 0.1}
LAS	{'alpha': 0.1}	{'alpha': 0.1}
KNN	{'leaf_size': 10, 'n_neighbors': 4}	{'leaf_size': 10, 'n_neighbors': 7}
GDB	{'learning_rate': 0.18000000000000002, 'n_estimators': 30}	{'learning_rate': 0.09, 'n_estimators': 50}
SVR	{'C': 0.1, 'kernel': 'linear'}	{'C': 0.1, 'kernel': 'sigmoid'}
RF	{'min_samples_split': 2, 'n_estimators': 110}	{'min_samples_split': 3, 'n_estimators': 110}

- ☐ Le modèle feat 3 semble être le plus performant
- ☐ Le Rigde et le Gradboost semblent être plus performants
- ☐ Obtention des meilleurs hypers paramètres

3) Modélisations et Prédictions



Sans Transformation log

CO2(Kg)

Avec Transformation log (unité de log)

	Feat2 R2	Feat3 R2	Feat2 RMSE	Feat3 RMSE
LR	0.412113	0.558361	456612.416065	383759.196905
ELN	0.446777	0.441745	456866.398562	459689.225697
RGE	0.412974	0.560478	456482.580715	389962.839408
LAS	0.412114	0.558361	456613.307885	383764.879183
KNN	0.400643	0.490832	455709.778319	437244.259198
GDB	0.483819	0.565752	453240.169923	409399.369013
SVR	-0.036735	-0.030956	613463.287058	611535.067033
RF	0.501577	0.576306	468713.940461	430099.554339

	Feat2 R2	Feat3 R2	Feat2 RMSE	Feat3 RMSE
LR	0.595523	0.595468	0.915698	0.915771
ELN	0.579350	0.579548	0.933888	0.933563
RGE	0.595738	0.596815	0.915452	0.914219
LAS	0.558495	0.561149	0.956956	0.954002
KNN	0.327284	0.571652	1.182054	0.942185
GDB	0.734114	0.677478	0.742688	0.817607
SVR	0.575218	0.620608	0.938818	0.886299
RF	0.685142	0.656817	0.808176	0.843537

	feat2	feat3
LR	{'fit_intercept': True}	{'fit_intercept': True}
ELN	{'alpha': 0.30000000000000004}	{'alpha': 0.30000000000000004}
RGE	{'alpha': 1.0}	{'alpha': 0.4}
LAS	{'alpha': 1.0}	{'alpha': 1.0}
KNN	{'leaf_size': 10, 'n_neighbors': 3}	{'leaf_size': 10, 'n_neighbors': 4}
GDB	{'learning_rate': 0.04, 'n_estimators': 130}	{'learning_rate': 0.02, 'n_estimators': 190}
SVR	{'C': 1.4000000000000001, 'kernel': 'linear'}	{'C': 1.4000000000000001, 'kernel': 'poly'}
RF	{'min_samples_split': 3, 'n_estimators': 10}	{'min_samples_split': 3, 'n_estimators': 10}

	feat2	feat3
LR	{'fit_intercept': True}	{'fit_intercept': True}
ELN	{'alpha': 0.1}	{'alpha': 0.1}
RGE	{'alpha': 1.0}	{'alpha': 1.0}
LAS	{'alpha': 0.1}	{'alpha': 0.1}
KNN	{'leaf_size': 10, 'n_neighbors': 4}	{'leaf_size': 10, 'n_neighbors': 8}
GDB	{'learning_rate': 0.09999999999999999, 'n_estimators': 130}	{'learning_rate': 0.12, 'n_estimators': 40}
SVR	{'C': 0.1, 'kernel': 'linear'}	{'C': 1.4000000000000001, 'kernel': 'rbf'}
RF	{'min_samples_split': 3, 'n_estimators': 120}	{'min_samples_split': 3, 'n_estimators': 110}

- ☐ Le modèle feat 3 semble être le plus performant
- ☐ Le Rigde et le Gradboost semblent être plus performants
- ☐ Obtention des meilleurs hypers paramètres

3) Modélisations et Prédictions



Jupyter Notebook, Python, Pandas, Numpy, Matplotlib, Seaborn

Sélection du featurig 3

Mise en place d'une validation
croisée répétée (KFoldRepeated)

Bilan des résultats

Sélection meilleur
modèle avec et sans
transformation log

4

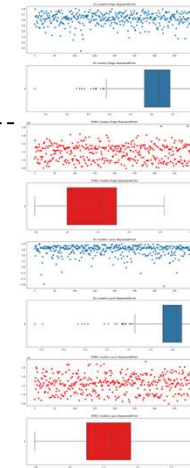
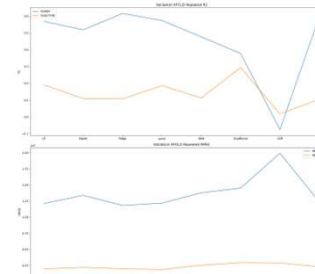
Validation des
résultats

SiteEnergyUse
(Kbtu)

CO2(kg)

- ☐ Paramétrage KFR (*split = 2, n_repeats = 200*)
- ☐ Mise en place des meilleurs hypers paramètres
- ☐ Entraînement et récupération des métriques (*R2 et RMSE*)

- ☐ Calcul des moyennes, min, max, écart type
- ☐ Lineplot
- ☐ Boxplot
- ☐ Scatterplot

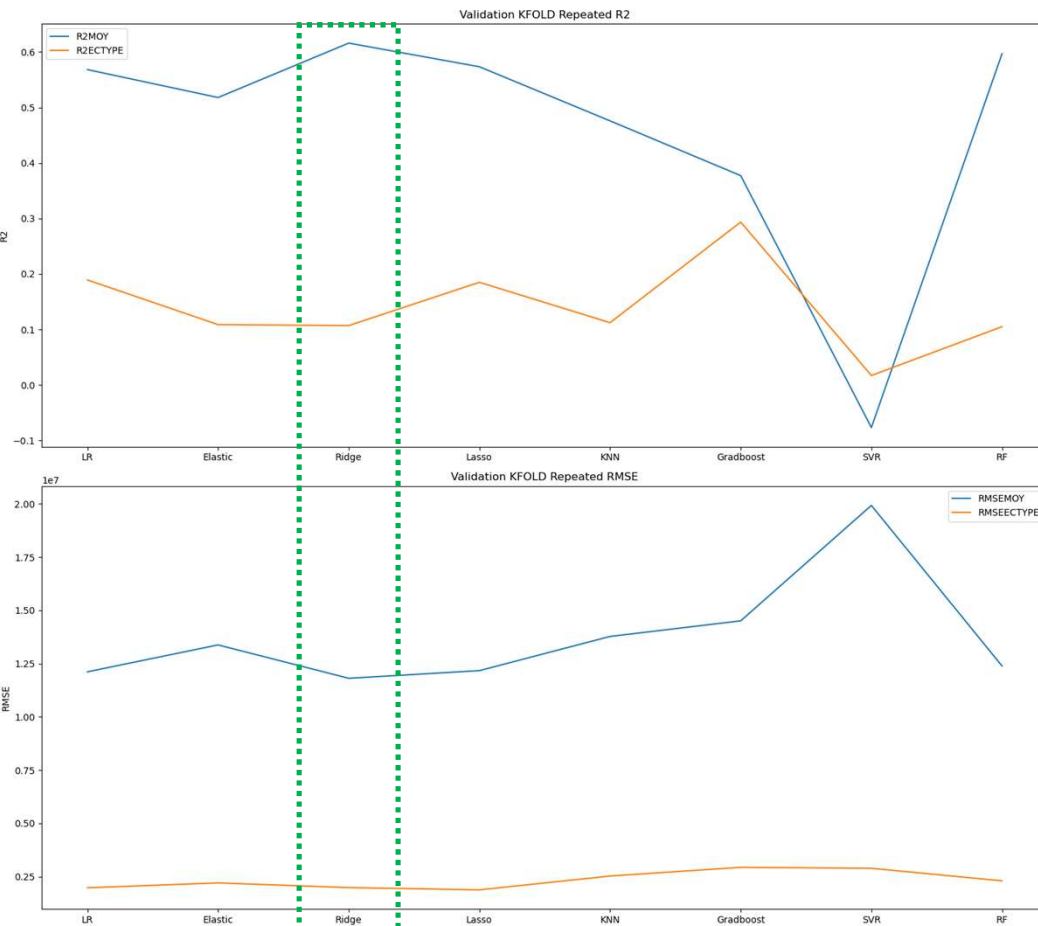


3) Modélisations et Prédictions

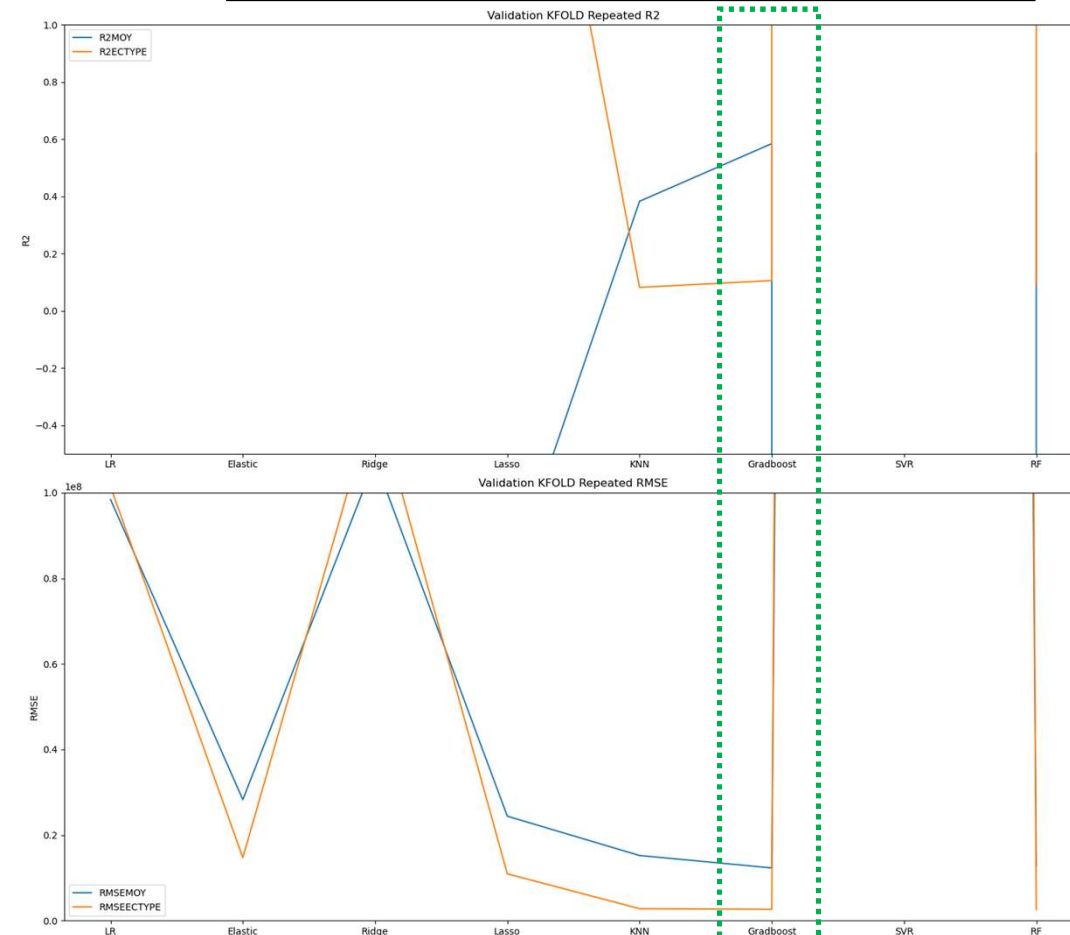
❑ Sans transformation LOG, le meilleur modèle est le RIDGE

SiteEnergyUSE(Kbtu)

❑ Avec transformation LOG, le meilleur modèle est le GRADBOOST



2023 Armand FAUGERE



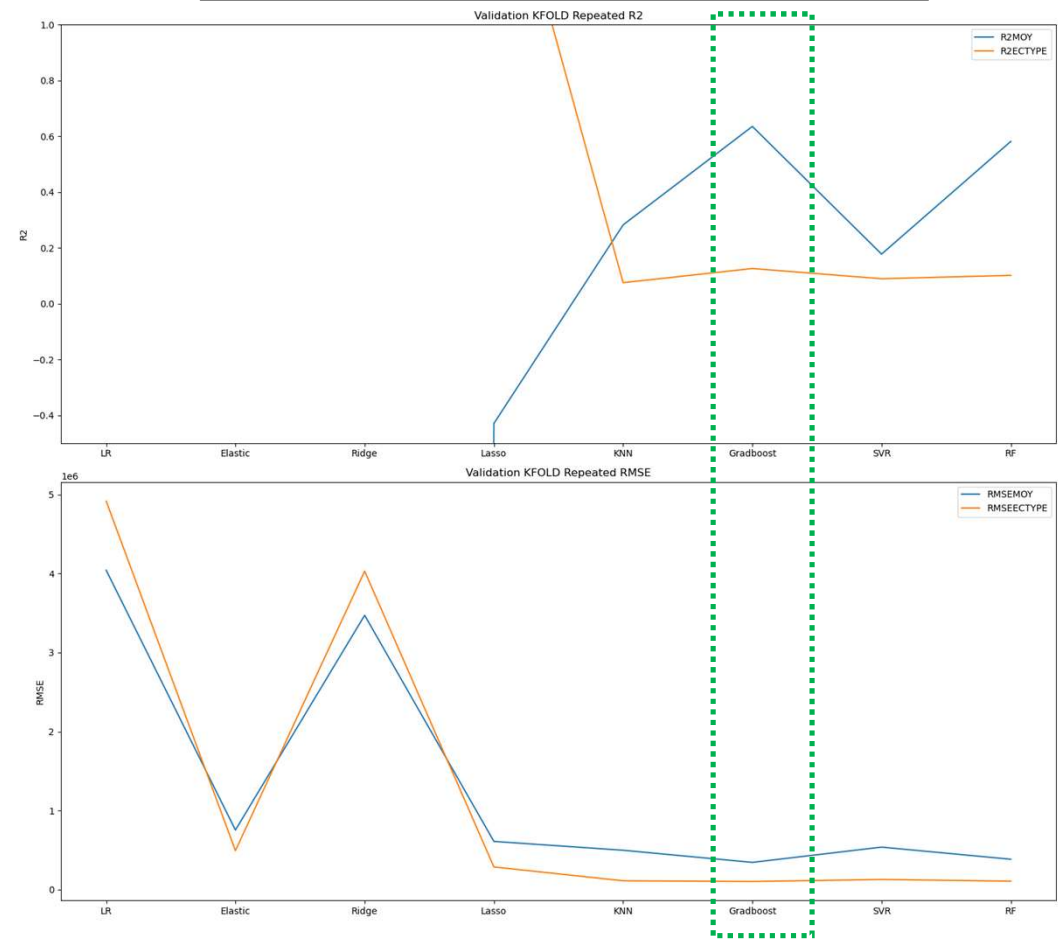
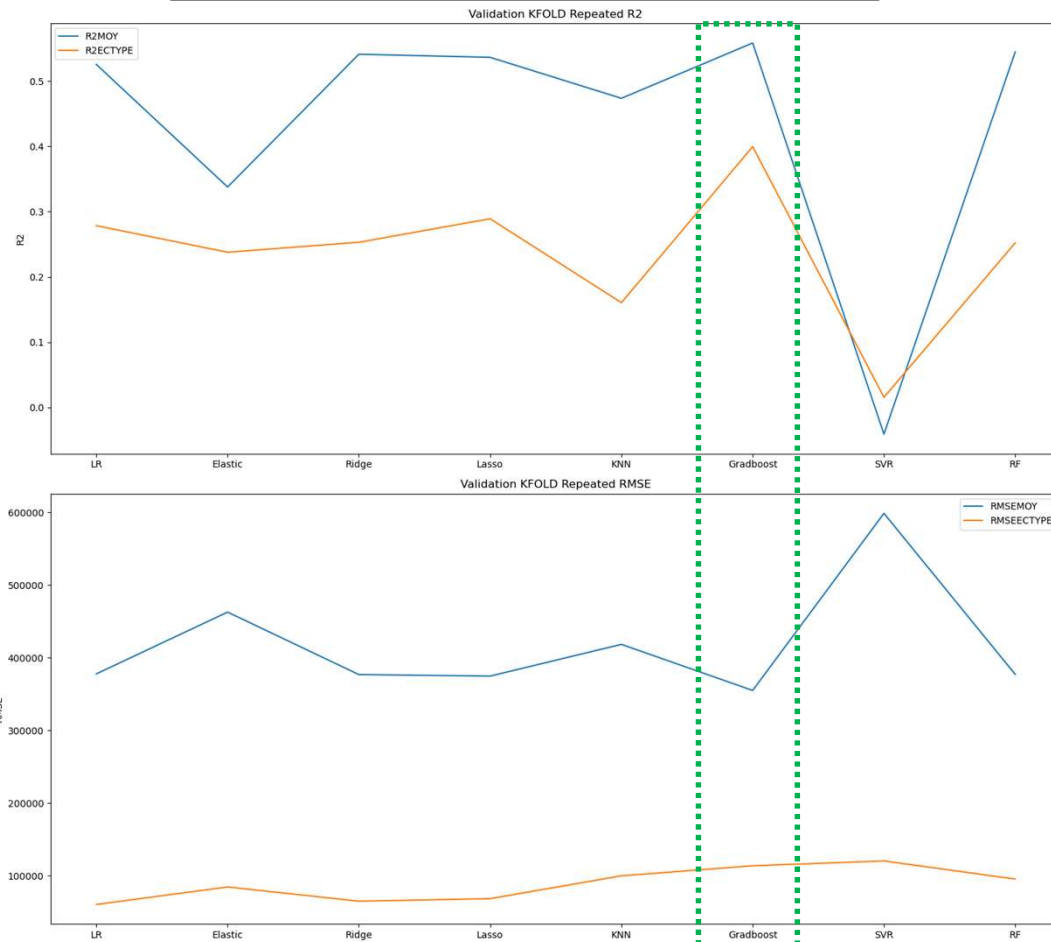
Anticipez les besoins en consommation de bâtiments

3) Modélisations et Prédictions

□ Sans transformation LOG, le meilleur modèle est le GRADBOOST

CO2(Kg)

□ Avec transformation LOG, le meilleur modèle est le GRADBOOST



IV) Choix de l'algorithme

4) Choix de l'algorithme

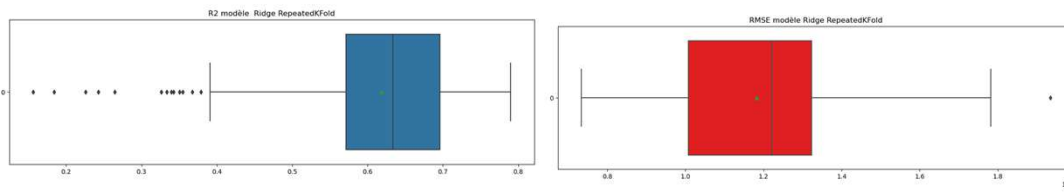
SiteEnergyUSE(Kbtu)

Modèle RIDGE (sans transformation de target)

$R^2 = 0.616$

RMSE = 1.181037×10^7

'feat3': {'alpha': 1.0}

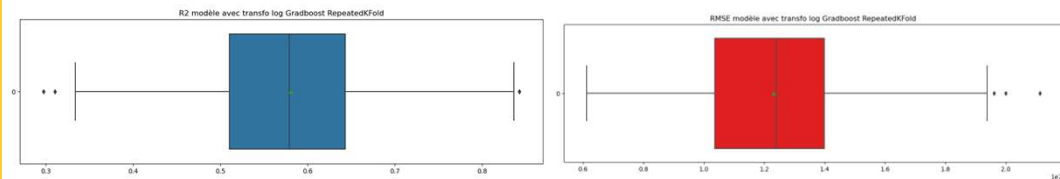


Modèle GRADBOOST (avec transformation target)

$R^2 = 0.584$

RMSE = 1.231867×10^7

'feat3': {'learning_rate': 0.09, 'n_estimators': 50}



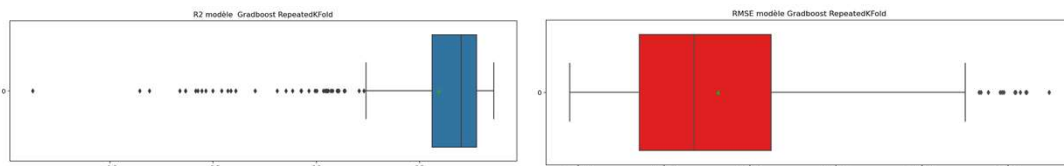
CO2(Kg)

Modèle Gradboost (sans transformation target)

$R^2 = 0.558$

RMSE = 354848.005577

'feat3': {'learning_rate': 0.02, 'n_estimators': 190}

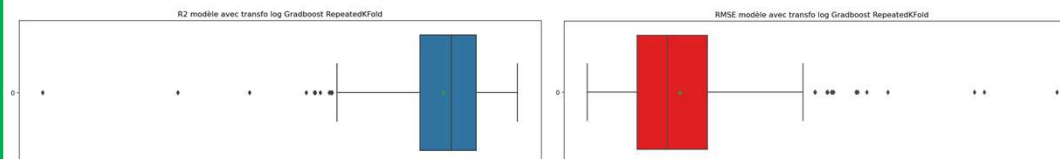


Modèle Gradboost (avec transformation target)

$R^2 = 0.635357$

RMSE = 345292.402070

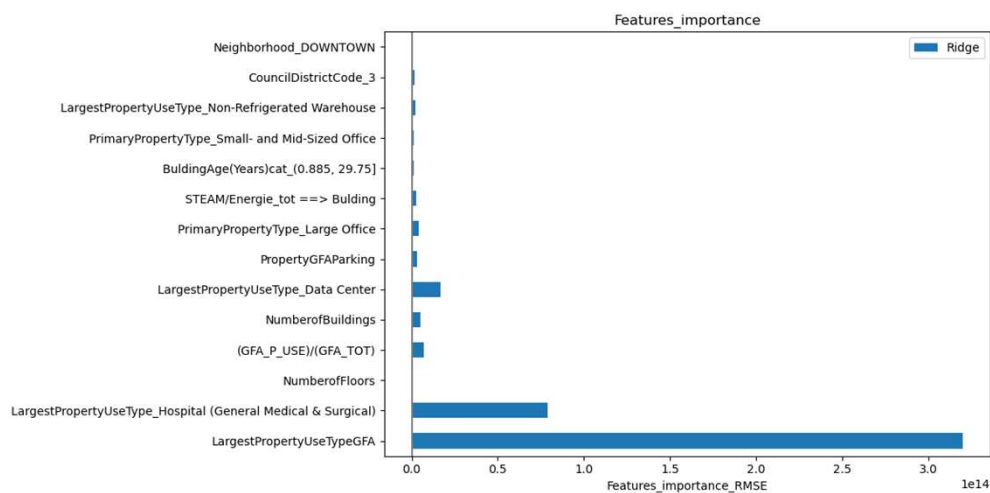
'feat3': {'learning_rate': 0.12, 'n_estimators': 40}



V) Interprétation

5) Interprétation

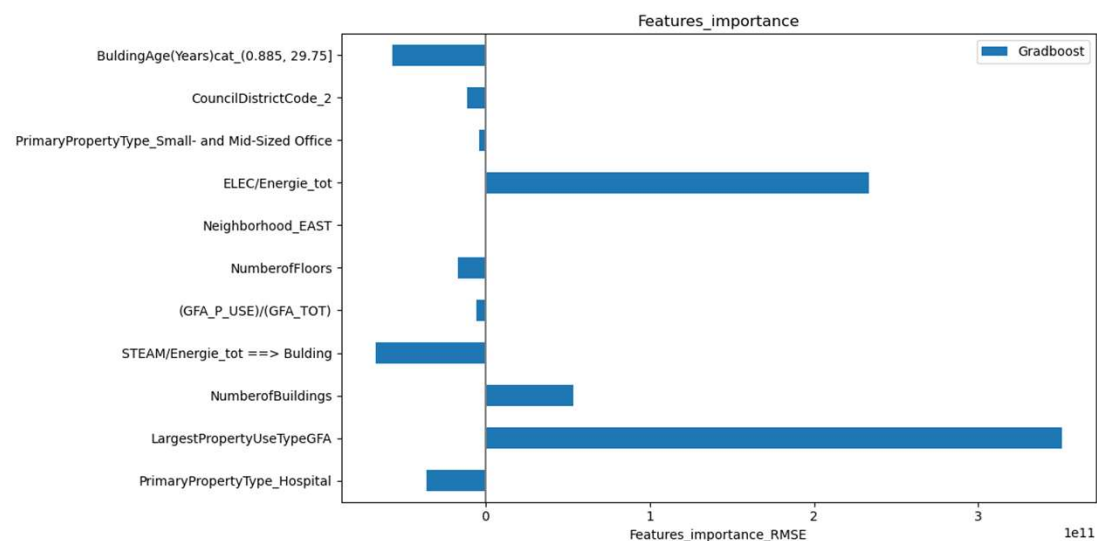
SiteEnergyUSE(Kbtu)



❑ **Les 3 variables importantes sont :**

- LargestPropertyUseTypeGFA,
- LargestPropertyUseType_Hospital (General Medical & Surgical),
- LargestPropertyUseType_Data Center

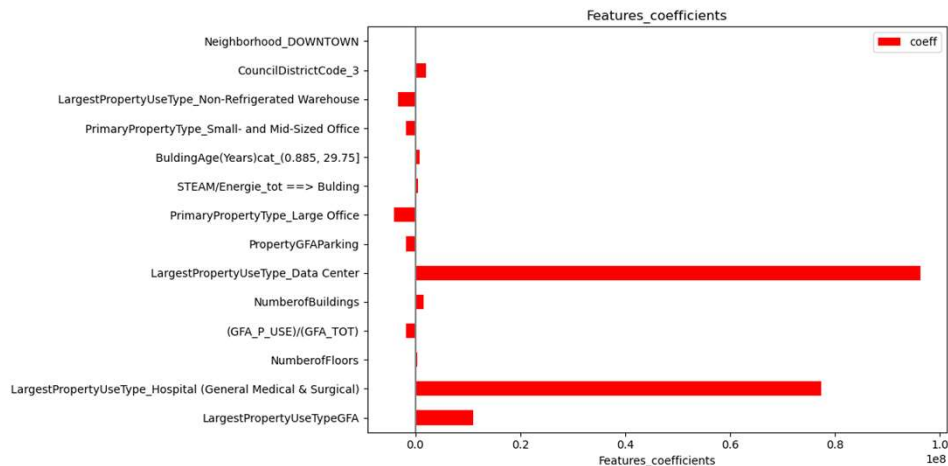
CO2(Kg)



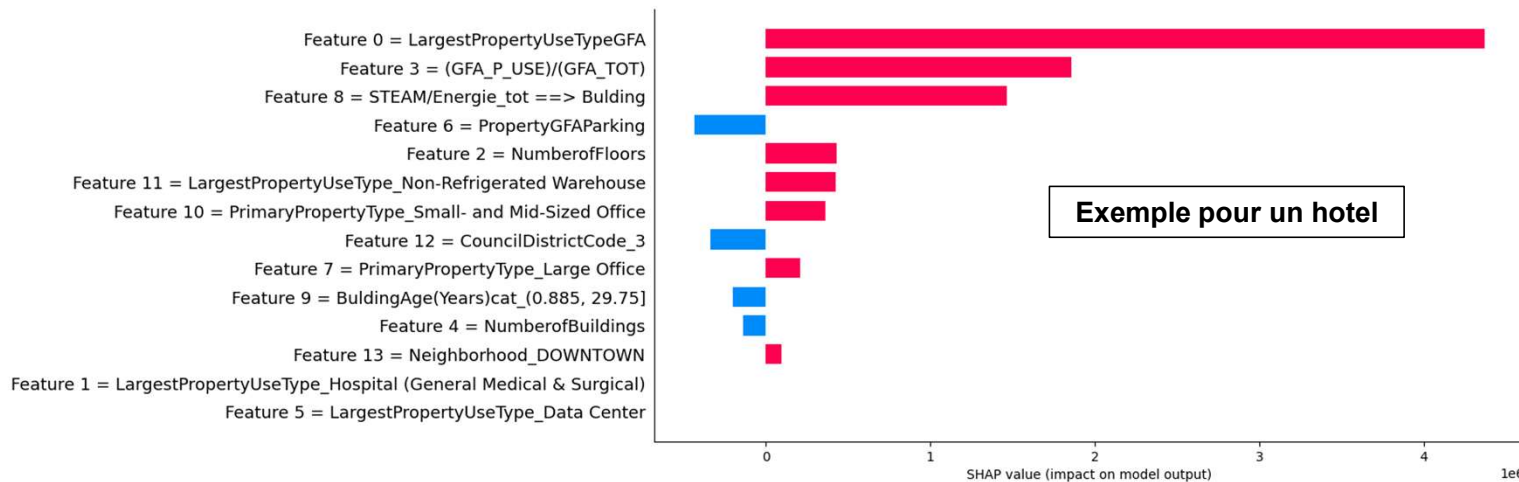
❑ **Les 3 variables importantes sont :**

- LargestPropertyUseTypeGFA,
- ELEC/Energie_tot
- NumberofBuildings

5) Interprétation



❑ 3 variables font augmenter plus fortement le niveau de consommation



Exemple pour un hotel

❑ 3 variables font augmenter plus fortement le niveau de consommation :

- 2 liées à la surface
- 1 à l'énergie utilisée

❑ D'autres variables font baisser le niveau de consommation

VI) Conclusion

6) Conclusion



But :

→ Créer un système de prédiction **des émissions de CO2 et de consommation totale d'énergie** des bâtimentss non destinés à l'habitation de la ville de Seattle en se basant sur les données structurelles des bâtiments.

Objectifs :

- Traiter et explorer le jeu de donnée
- Réaliser des modélisations prédictives
- Evaluer les modèles
- Se passer des relevés de consommation annuels et futurs
- Eviter le Data leakage



Recommandations :

- ☐ Mettre en place des plans d'optimisation de surface utiles (réduire les surfaces et conserver la valeur ajoutée)
- ☐ Mettre en place un plan d'actions pour les hôpitaux et les Datacenters (surveillance, aides pour rénovation et amélioration du matériel)
- ☐ Pour certaines activités ciblées, encourager la rénovation des bâtiments
- ☐ Encourager la performance énergétique des équipements électriques

Merci

- Armand FAUGERE
- armand-faugere@live.fr

