Seattle

ANTICIPATION DES BESOINS EN CONSOMMATION D'UN BATIMENT

Soutenance Projet 3

Etudiant: SEKPONA Kokou Sitsope

INTRODUCTION

Nous travaillons pour la ville de Seattle. Elle nous demande de l'aider pour atteindre son objectif de ville neutre en émissions de carbone en 2050. Nous allons donc étudier de près la consommation et les émissions des bâtiments non destinés à l'habitation.
Puisque les relevés sont très couteux, nous allons a partir des relevés déjà réalisées, tenter de prédire les émissions de CO2 et la consommation totale d'énergie de bâtiments non destinés à l'habitation pour lesquels elles n'ont pas encore été mesurées.
Pour ce faire, nous allons:

- Réaliser une courte analyse exploratoire.
- Tester différents modèles de prédiction afin de répondre au mieux à la problématique et tenter d'améliorer leur performances.

Chronologie du Projet

Nettoyage

Notre dataset a 3376 ligne et 46 colonnes.

Analyse exploratoire

Prediction de CO2

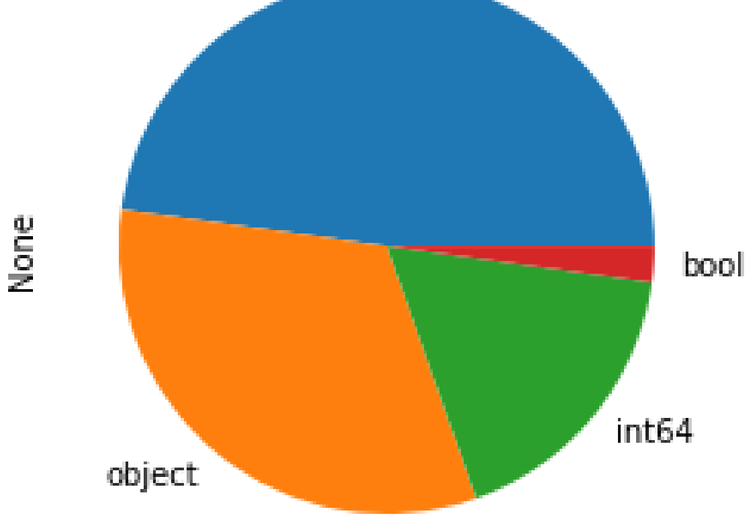
Prediction Consommation energie

I. Nettoyage

1. Les différents types de données

la pluspart des valeurs sont du type float, ensuite du type et un peu de type int aussi, et puis quelques valeurs booléannes

Aussi, notre dataset contient 3375 lignes et 46 colonnes.
Mais puisque nous n'allons prédire au final que de bâtiments destinés à la non habitation, nous n'allons garder que lignes qui sont des bâtiments qui sont des "Nonresidential"




```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1546 entries, 0 to 3375
Data columns (total 46 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   OSEBuildingID                    1546 non-null   int64
 1   DataYear                         1546 non-null   int64
 2   BuildingType                     1546 non-null   object
 3   PrimaryPropertyType              1546 non-null   object
 4   PropertyName                     1546 non-null   object
 5   Address                          1546 non-null   object
 6   City                             1546 non-null   object
 7   State                            1546 non-null   object
 8   ZipCode                          1530 non-null   float64
 9   TaxParcelIdentificationNumber    1546 non-null   object
```

2. Valeurs manquantes

Pres de 8 colonnes sur les 46 contiennent des valeurs manquantes dont les premieres sont Comment, Outlier, YearsEnergyStarCertified, ThirdLargestPropertyUseType, ThirdLargestPropertyUseTypeGFA (qui ont plus de 50% de valeurs manquantes)

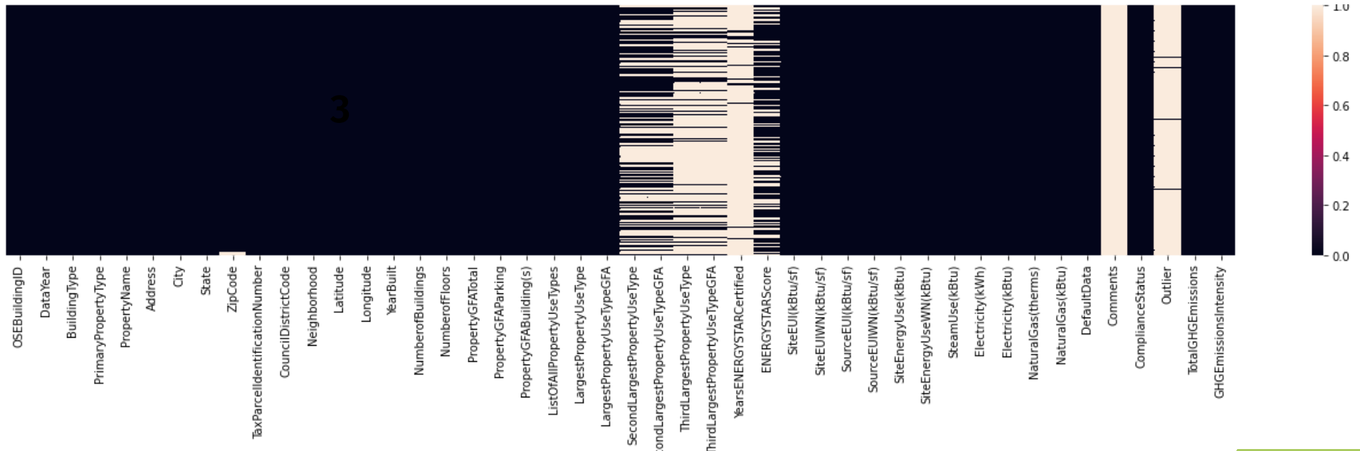
	Nan_percent	colonnes
41	100.000000	Comments
43	98.965071	Outlier
27	94.113842	YearsENERGYSTARCertified
25	77.684347	ThirdLargestPropertyUseType
26	77.684347	ThirdLargestPropertyUseTypeGFA
23	45.536869	SecondLargestPropertyUseType
24	45.536869	SecondLargestPropertyUseTypeGFA
28	34.928849	ENERGYSTARScore

Comme EnergyStarScore est une variable importante:

- Nous creons un sous dataset qui n'a que les lignes qui ont des valeurs energyStarScore
- Ensuite nous faisons l'imputation par la mediane d'energyStarScore dans la dataset originale.

-Méthodes de traitement des valeurs manquantes:

- Suppression des colonnes à plus de 50% de NaN (Les colonnes dans le cas ne sont pas trop importantes aussi à la prédiction)
- Imputation par la médiane de celles restants de type réels
- Imputation par le mode des celles restantes de type Objet



Suppression des colonnes inutiles

3. Features Enginering

a Etude des correlations et hypothèses

Nous allons etudier la correlation entre les variables dans cette partie et emetttre des hypothèses

b Ratio entre certaines colonnes

A cette etape nous allons transformer certaines colonnes en ratio pour creer de nouvelles variables mais suprimer galement certaines qui ne serront plus utile

c

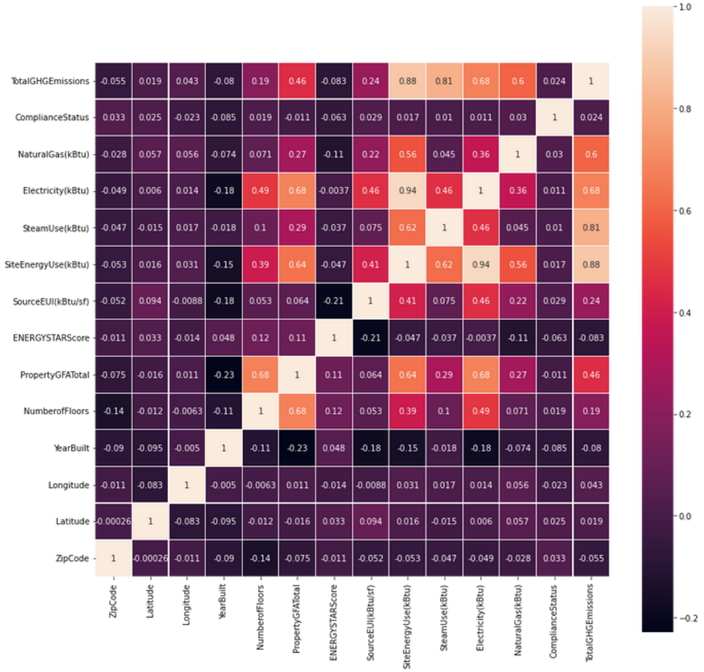
La variable Yearbuilt est l'année de cnstruction et nous l'avons transformer en difference entre l'année de construction et 2016 afin de garder le critère de récence ou d'ancienneté.

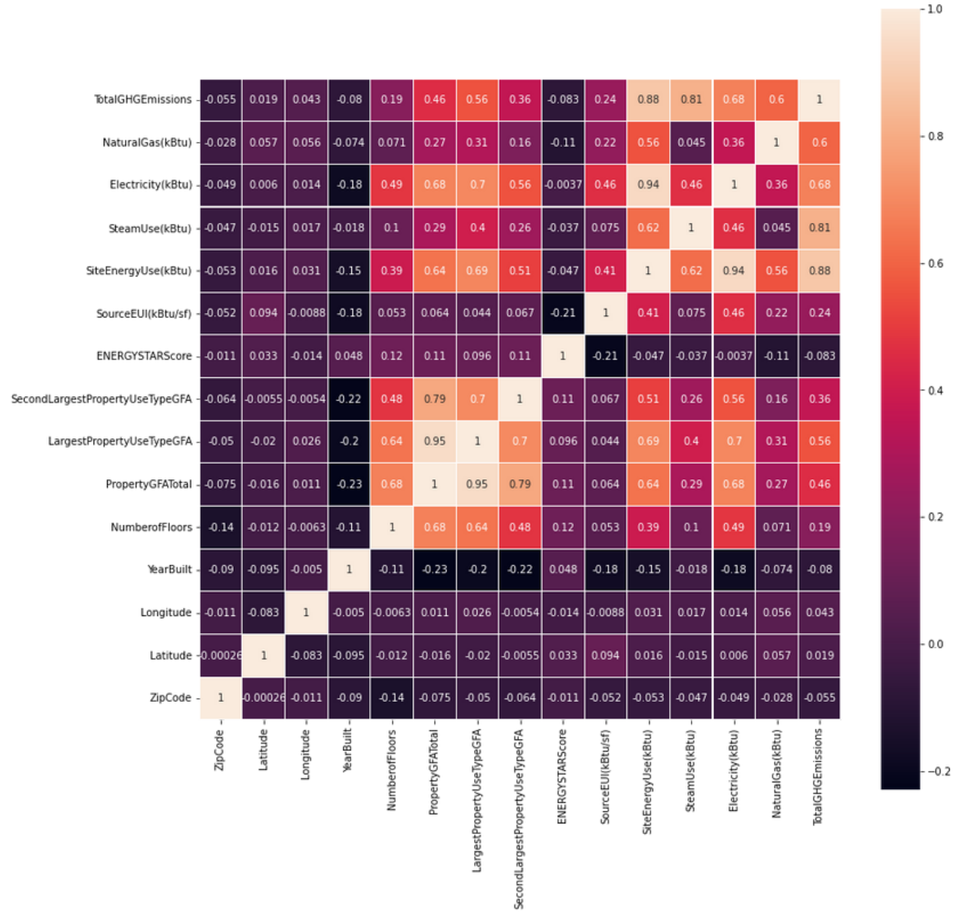
d Encodage de certaines variables categorielles

Cette variable indique si une propriété a satisfait aux exigences d'analyse comparative énergétique pour l'année de déclaration en cours. On va donc l'encoder en donnant plus d'importance aux observations satisfaisant à la condition

Correlation entre les variables (Details)

Nous observons ici:
*'SecondLargestPropertyUseTypeGFA et le LargestPropertyUseTypeGFA sont tres correllés à 'PropertyGFATotal', Nous allons les supprimer





Apres Traitement des variables
nous obtenons ce heatmap et la taille de notre dataset est maintenant de 1006 observations et 23 colonnes

4. Doublons

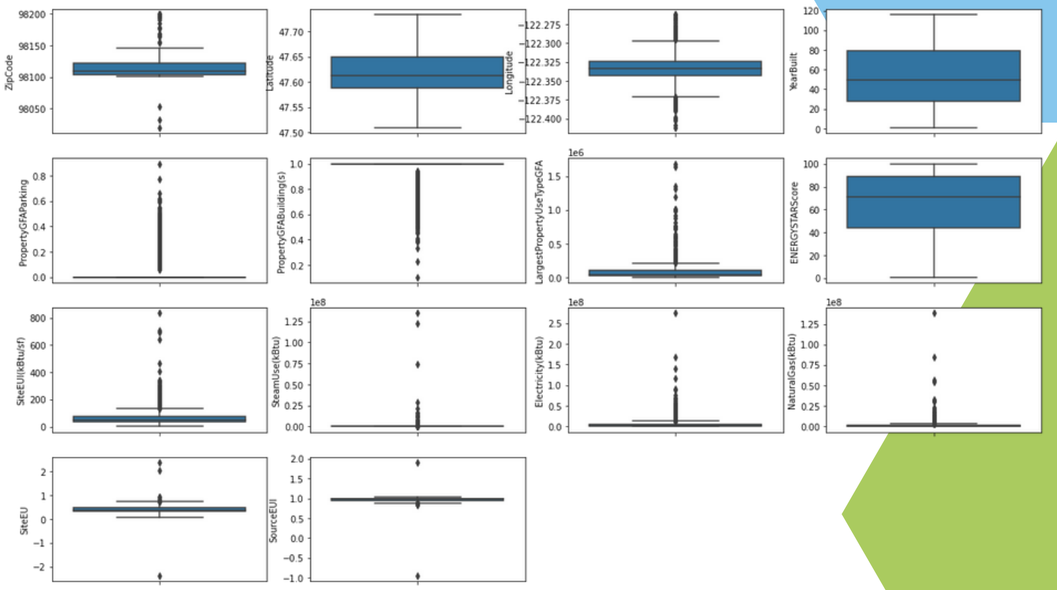
Nous avons 5 qui ont la même adresse, même longitude, même latitude, même année de construction, même numéros de construction.
nous allons donc les supprimer

```
doublons=df[['Latitude','Longitude', 'YearBuilt', 'ZipCode','Address']].duplicated().sum()
print(f'Nous avons {doublons} qui ont la meme adresse, meme longitude, meme latitude, meme année de construction, meme numéros d

Nous avons 5 qui ont la meme adresse, meme longitude, meme latitude, meme année de construction, meme numéros de construction.
 nousallons donc les supprimer
```

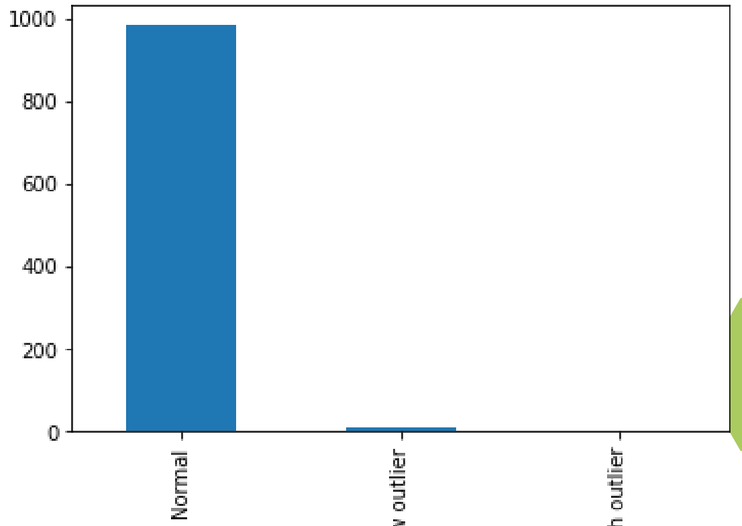
5. Valeurs aberrantes

Les variables comme 'SourceEUI' et SiteEU comportent des valeurs négatives. Bien que ce ne sont pas des valeurs anormales (des batiments qui fournissent de l'enrgie), ces valeurs vont reduire la précision de notre modele. Nous allons donc supprimer ces valeurs



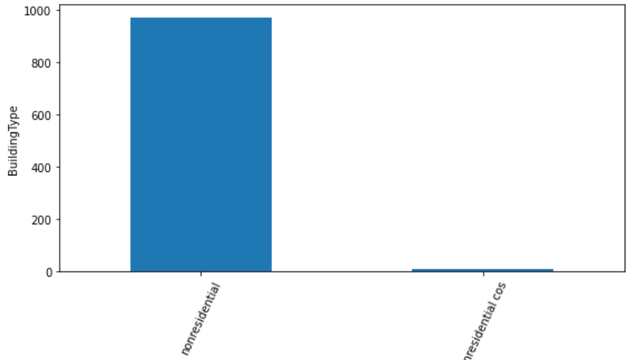
Toutes les autres valeurs apparement aberrantes sont possibles, mais cela entrainera un mauvais fonctionnement de notre modele. Nous allons corriger cela plus tard par le passage au log et le standard scaling

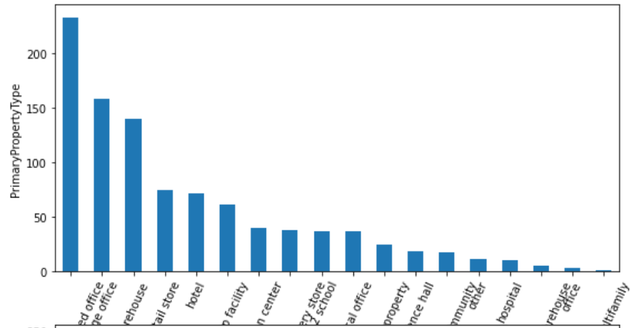
Ce diagramme est la distribution de la variable outliers. Nous n'allons conserver que la partie normale supprimer les autres observations

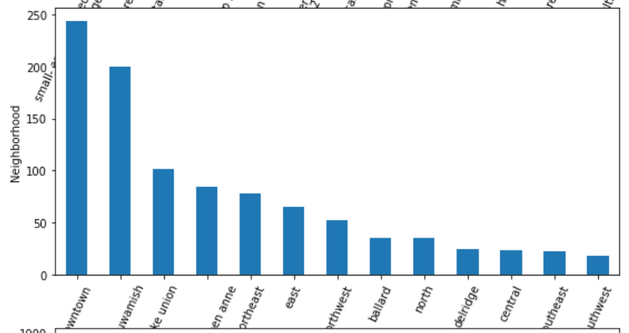


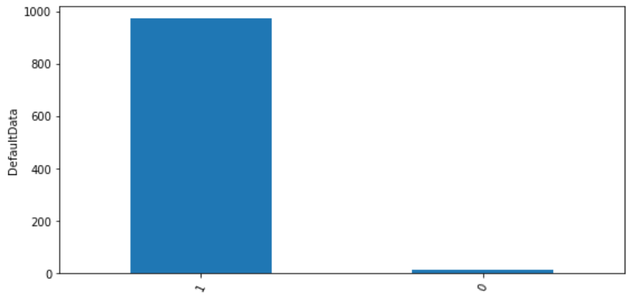
Analyse Univarié

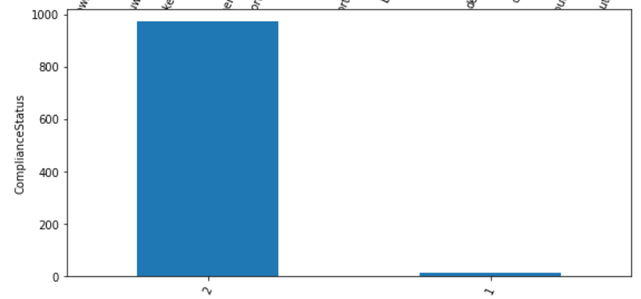
1. Variables qualitatives









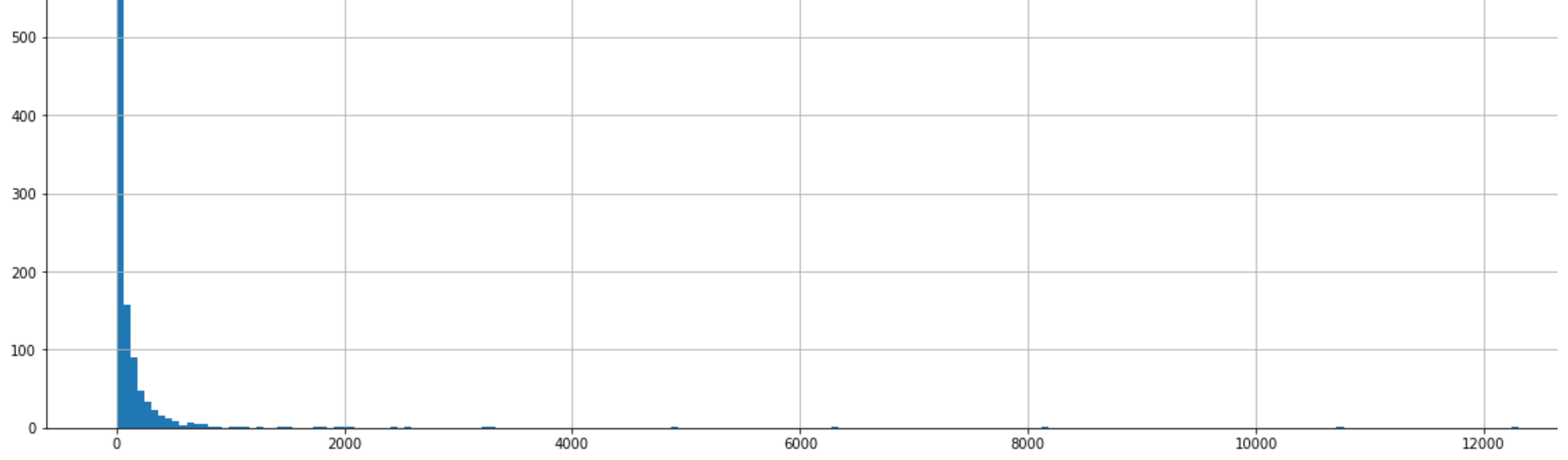


Les batiments appartiennent en majorité à la ville downtown et sont en majorité des bureau.

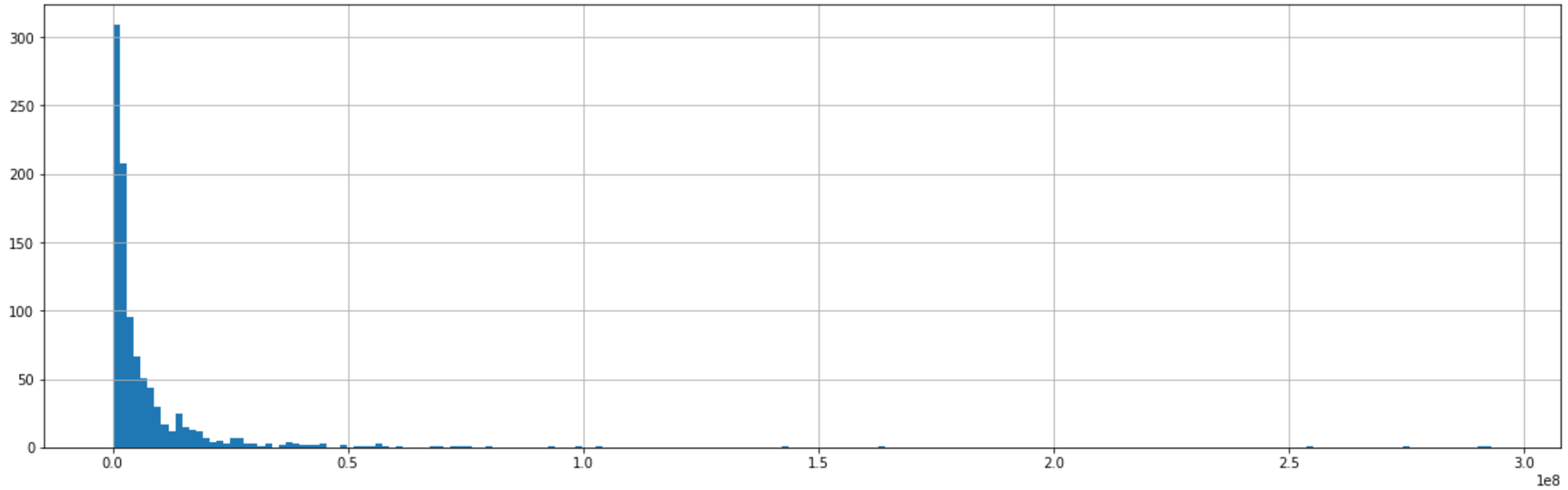
2. Analyse des Targets

La distribution est la même pour les 2 target, une distribution asymétrique étalé vers la droite.

Cela montre près de la moitié de bâtiments ont une consommation et une émission nulle en CO_2 et en consommation d'Energie, c'est une hypothèse qui semble être vérifié par le fait que plus la plupart des bâtiments sont des bureaux, les bureaux consommant la plus faible Energie possible.

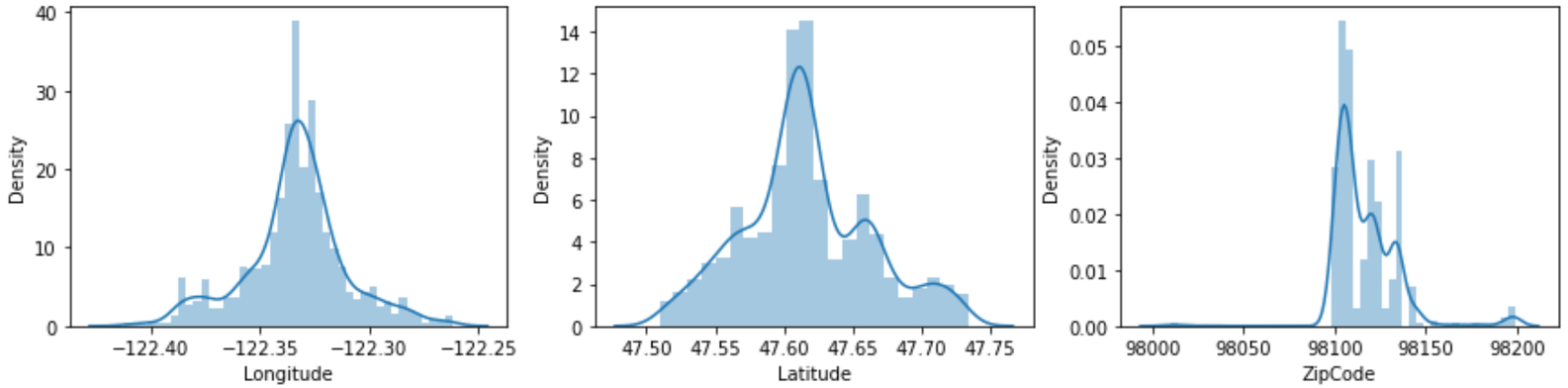


TotalGHGEmissions

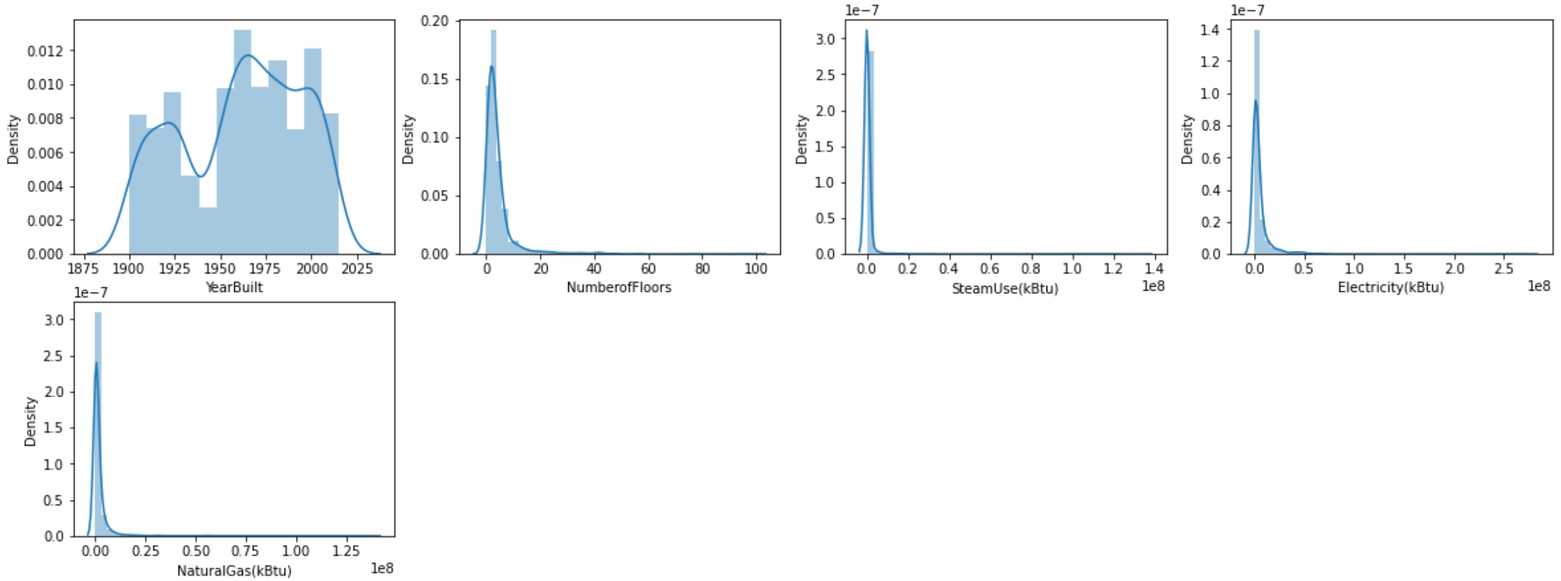


SiteEnergyUse(kBtu)

3. Variables quantitatives



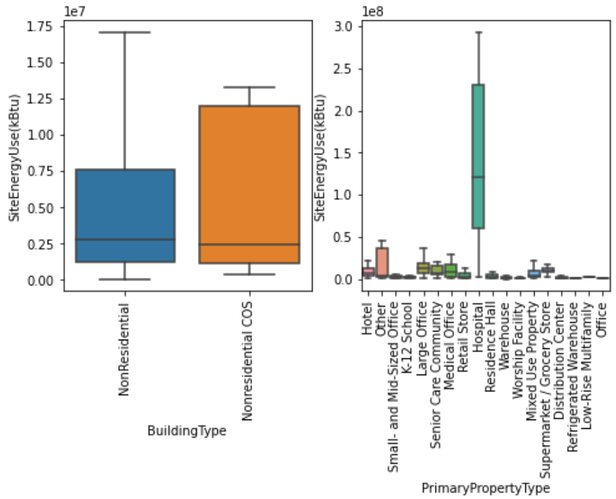
Les variables Longitudes, latitudes, Zipcode ont une distribution normalisée

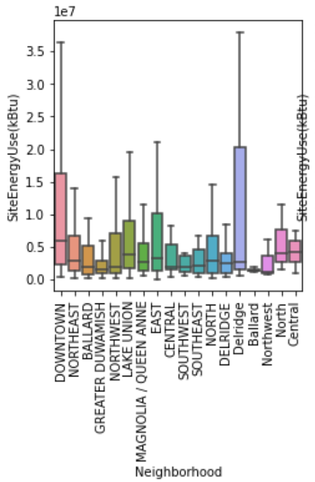


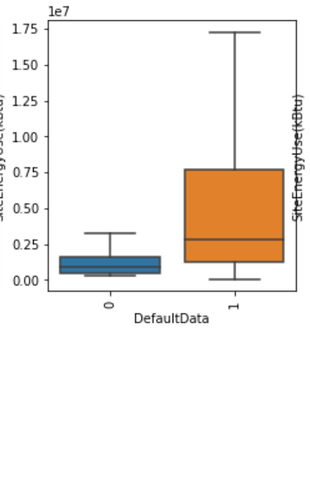
YearBuilt',
'NumberofFloors',
'SteamUse(kBtu)',
'Electricity(kBtu)',
'NaturalGas(kBtu)', n'ont pas de distribution gaussienne, on va plus tard les normaliser.

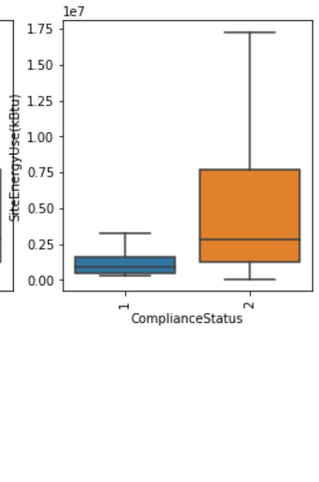
Analyse Bivarié

Rapport entre certaines variables et les targets

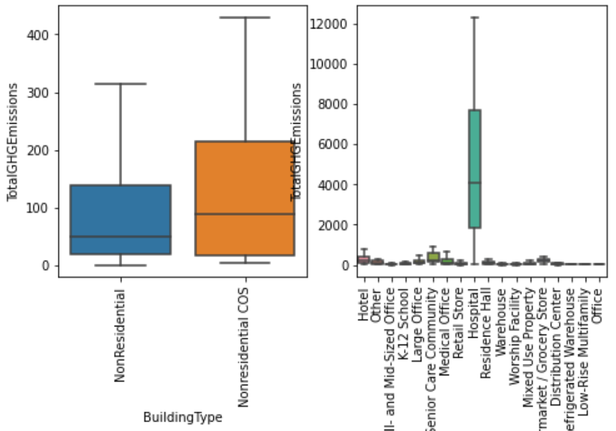


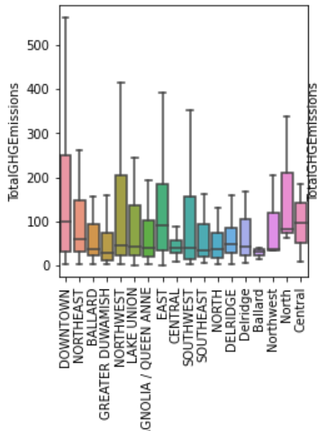


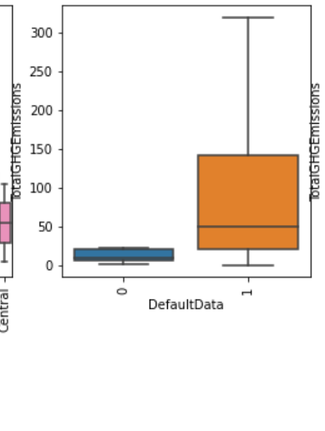


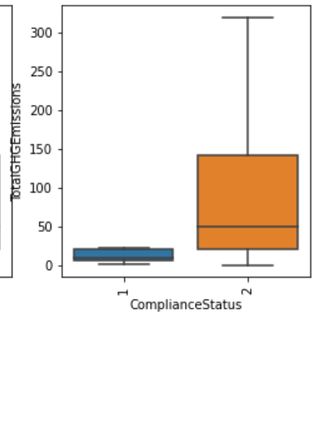


- Nous pouvons en deduire que certaines villes utilisent plus d'energie et émettent plus de C02 que d'autres: C'est le cas des capitales ou des villes dévéloppés(DOWNTOWN)





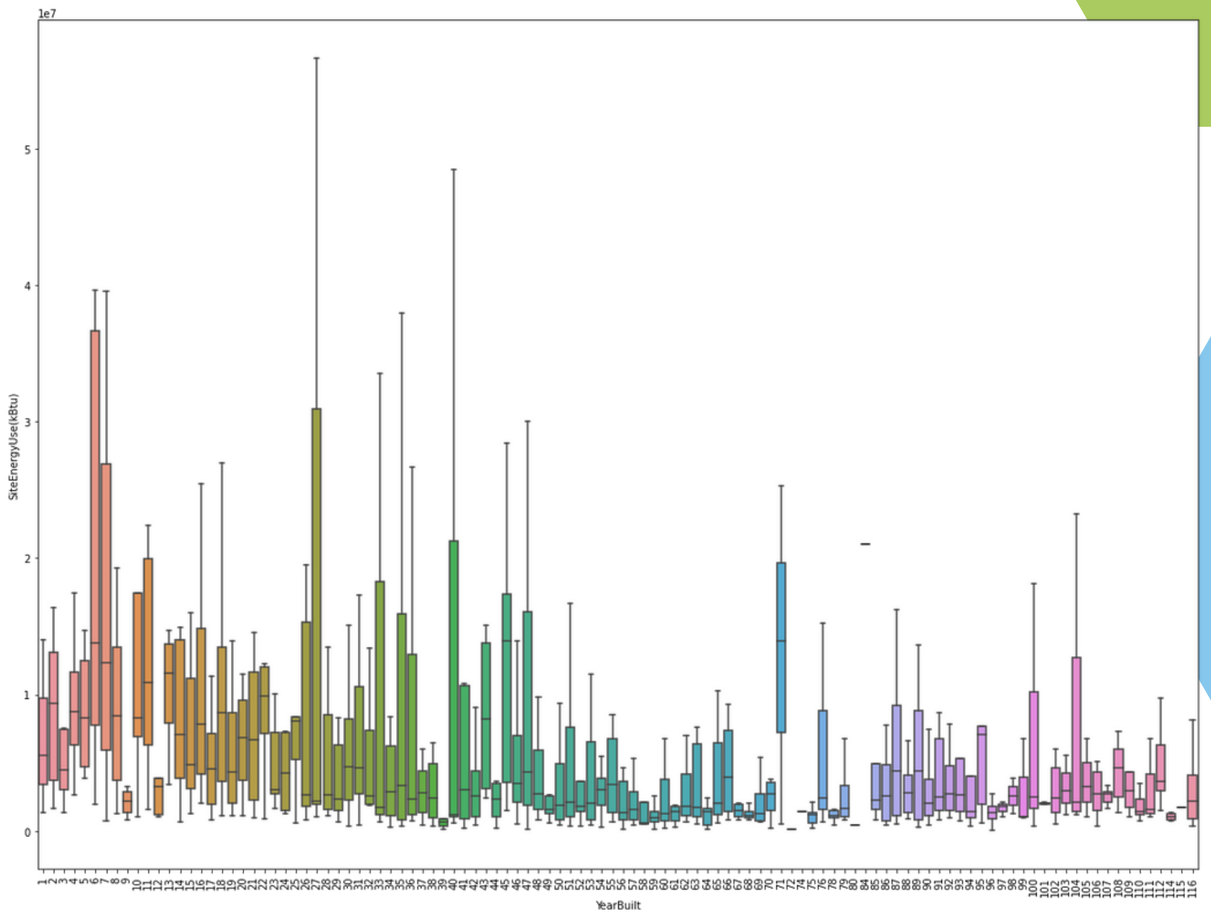




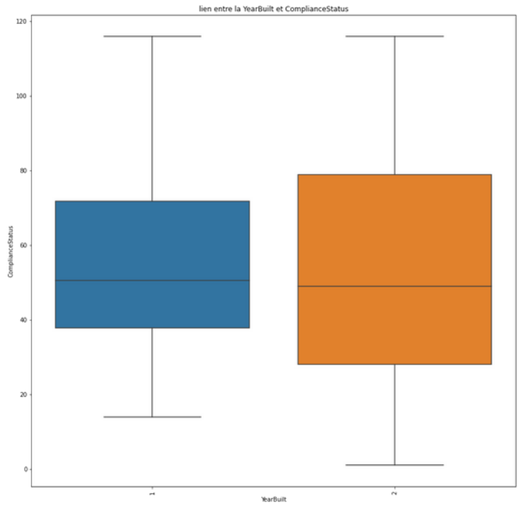
- Les hopitaux utilisent enormement d'energie et dégagent plus de CO2 que les autres imeubles

Rapport entre le Yearbuilt et le Site Energy Use

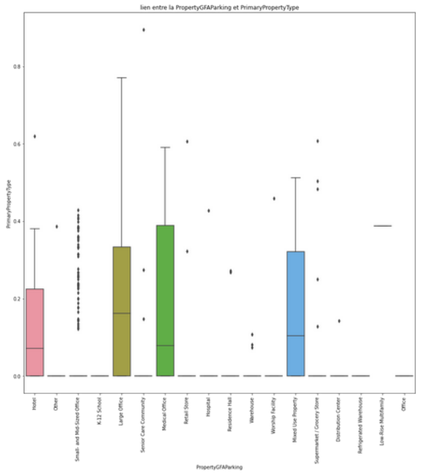
les nouveaux batiments ont tendance à consommer plus d'energie que les anciennes



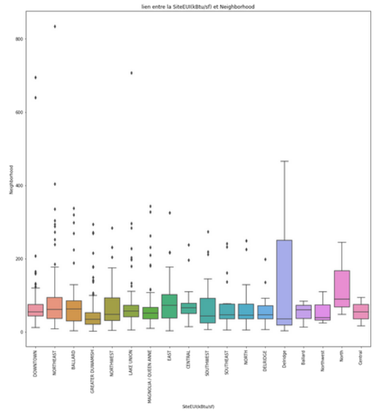
Relations entre certaines variables qualitatives quantitatives



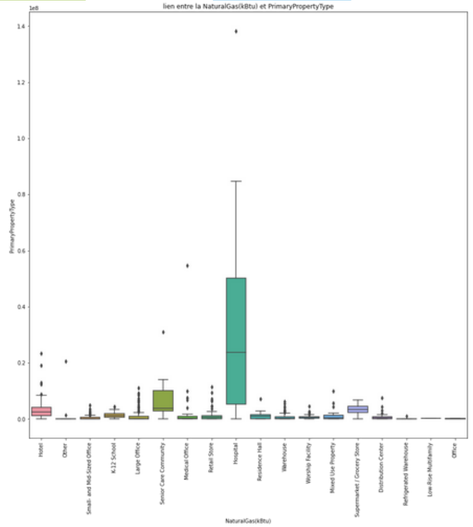
Relation entre YearBuit et Compliance Status



Relation entre PropertyGFAParking et Compliance Status et Primary propertyType



Entre SiteEnergyUse et NeighborsHood



Relation Entre PrimaryPropertyTpe et NaturalGaz

Prediction d'emission C02

Nous allons dans cette section tester les différents modelés et hyperparamètres pour choisir le meilleur modèle et hyperparamètre qui donne la meilleure précision

Preprocessing

Standardisation avec le StandardScaler

Encodage des variables catégorielles avec le OneHotEncoder

Transformation des variables catégorielles en une matrice compressée de 0 et de 1 afin qu'il soit utilisable par le modèle lors de l'entrainement.

Passage au log des variables quantitatives

Nous remplaçons les valeurs négatives par 0 et nous passons au log toutes les valeurs de ces colonnes

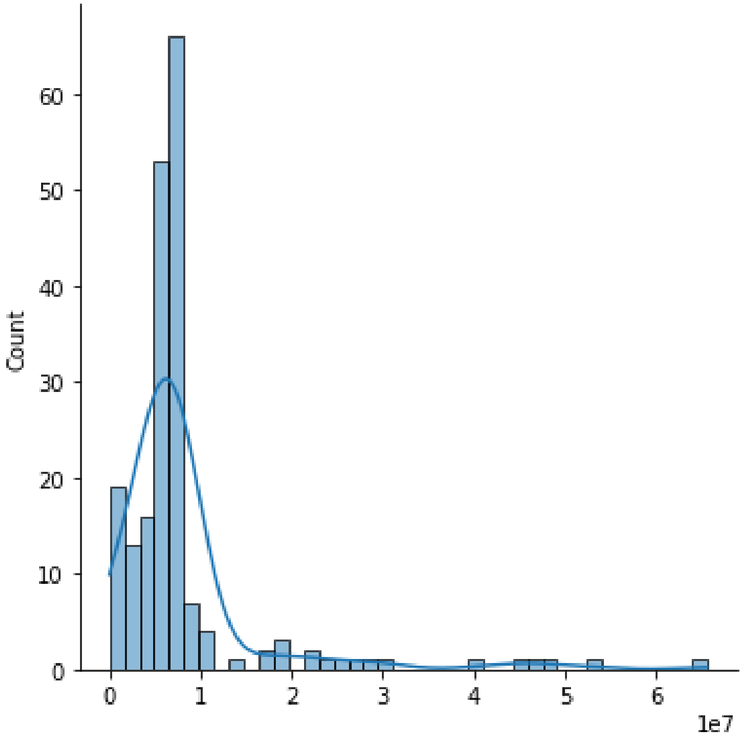
Separation de la dataset en target et features
Puis séparation en train et test avec le train_test_split de sklearn

```
colonnes=[i for i in df.columns if i not in targets]
X=df[colonnes]#.apply(lambda x: str(x)).values
y=df["TotalGHGEmissions"].values
```

Modelisation

1. Model DummyRegressor

Avec le GridSearchCV, nous avons
tester ce model avec les differentes stratégies suivantes:
'mean', 'median', 'constant'



Nous avons calculé
les différents types d'erreur.
Score:-0.024
erreur totale moyenne: 30.907131113589923

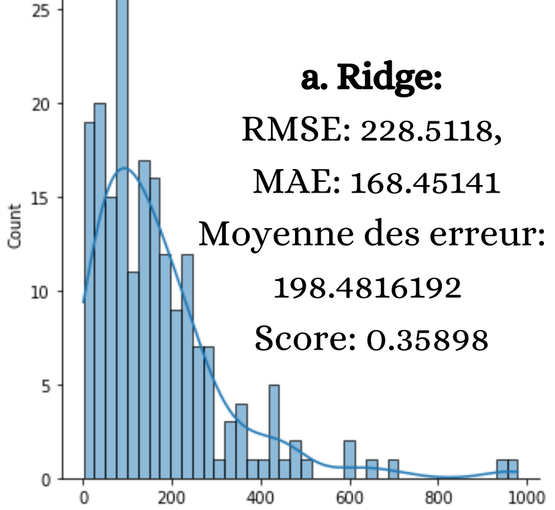
Une précision très insignifiante, ce model n'arrive pas à décrire les phénomènes à l'origine.

2. Modeles linéaires

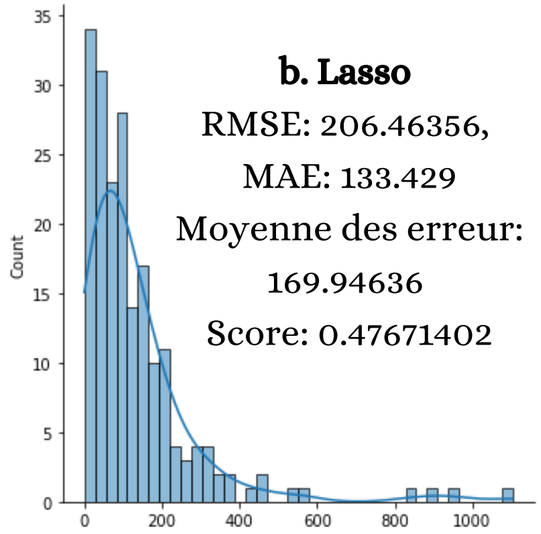
Nous utilisons les modeles linéaires tels que Ridge, Lasso, ElasticNet, tous avec du GridSearchCV pour trouver les meilleurs paramètres.

Nous allons dessiner la courbe de la différence entre les valeurs prédites et les valeurs réels y_test

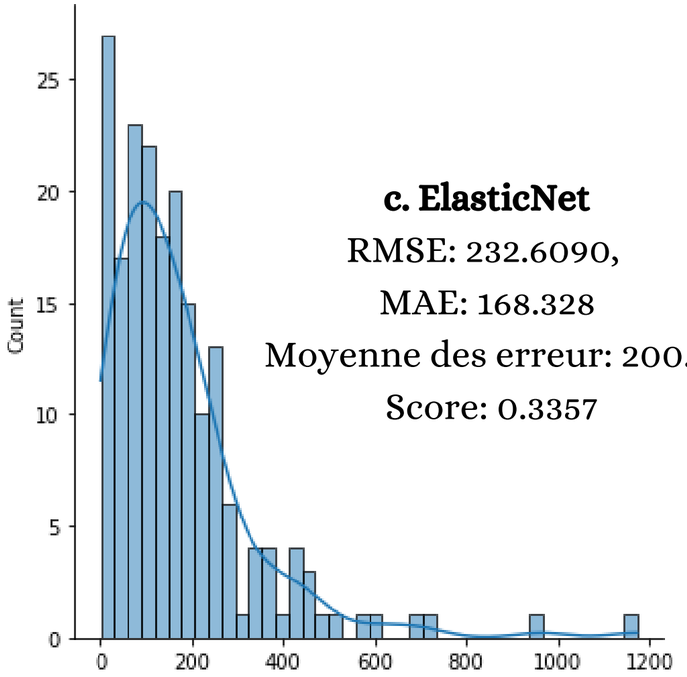
Ici, les modeles sont toues les memes sauf que le model lasso commet mois d'erreur que les autres modeles



a. Ridge:
RMSE: 228.5118,
MAE: 168.45141
Moyenne des erreur: 198.4816192
Score: 0.35898



b. Lasso
RMSE: 206.46356,
MAE: 133.429
Moyenne des erreur: 169.94636
Score: 0.47671402



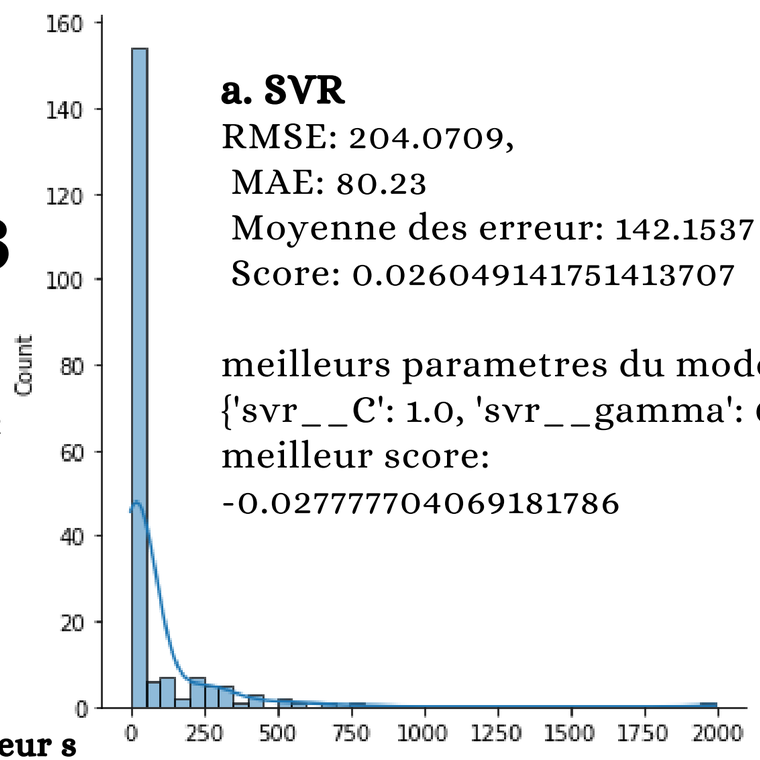
c. ElasticNet
RMSE: 232.6090,
MAE: 168.328
Moyenne des erreur: 200.4
Score: 0.3357

3. Modeles non linéaires

Nous utilisons ici le SVR et le RandomForestRegressor avec GreadSearchCV

Conclusion

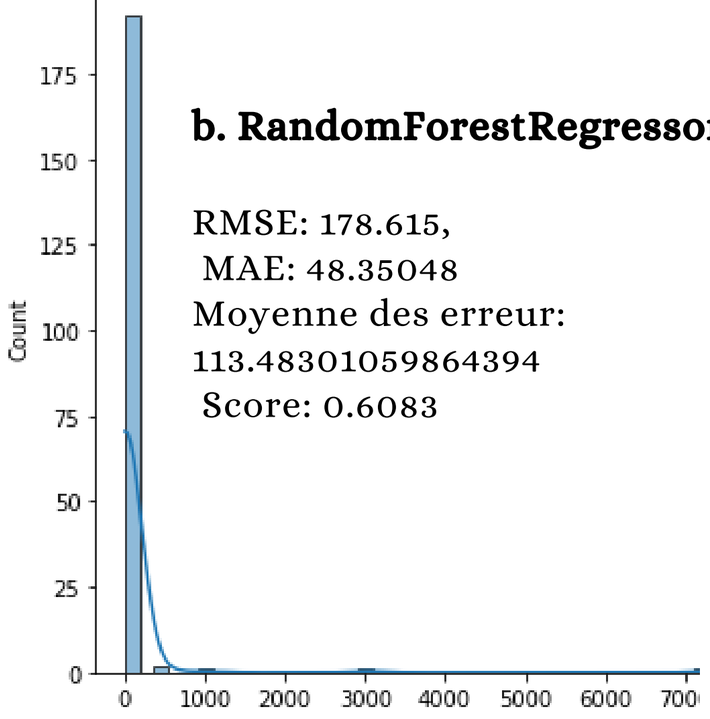
Le RandomForestRegressor a le meilleur s résultat sur notre jeu de donnée. ce serra le model retenu pour la prédiction de l'émission du CO2.



a. SVR

RMSE: 204.0709,
MAE: 80.23
Moyenne des erreur: 142.1537
Score: 0.026049141751413707

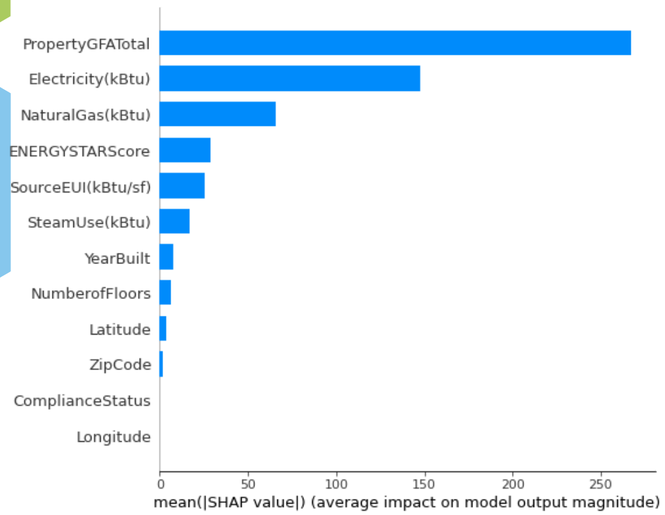
meilleurs parametres du model: {'svr__C': 1.0, 'svr__gamma': 0.1}, meilleur score: -0.027777704069181786



b. RandomForestRegressor

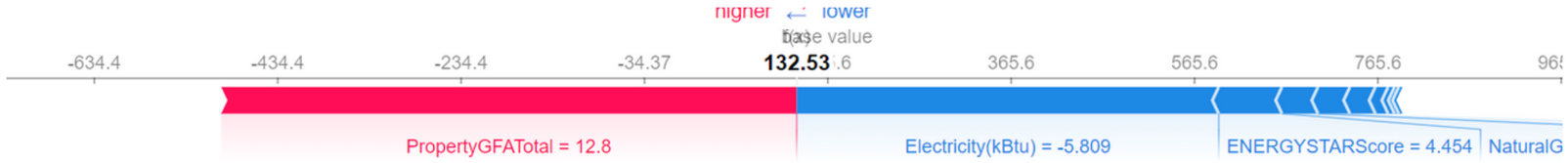
RMSE: 178.615,
MAE: 48.35048
Moyenne des erreur: 113.48301059864394
Score: 0.6083

Importance Globale



Importance globale montrant l'importance de chaque variable dans le model

Importance Locale



Importance locale montrant l'importance de chaque variable dans la prédiction d'une observation prise au hazard

Prediction de Consommation d'energie

Nous allons dans cette section tester les differents modeles et hyperparametres pour choisir le meilleur modele et hyperparametre qui donne la meilleure précision pour la prediction de la consommation d'energie avec et sans EnergyStarScore

Preprocessing

Standardisation avec le StandardScaler

Encodage des variables catégorielles avec le OneHotEncoder

Transformation des variables catégorielles en une matrice compressée de 0 et de 1 afin qu'il soit utilisable par le modèle lors de l'entrainement.

Passage au log des variables quantitatives

Nous remplaçons les valeurs négatives par 0 et nous passons au log toutes les valeurs de ces colonnes

Separation de la dataset en target et features
Puis séparation en train et test avec le train_test_split de sklearn

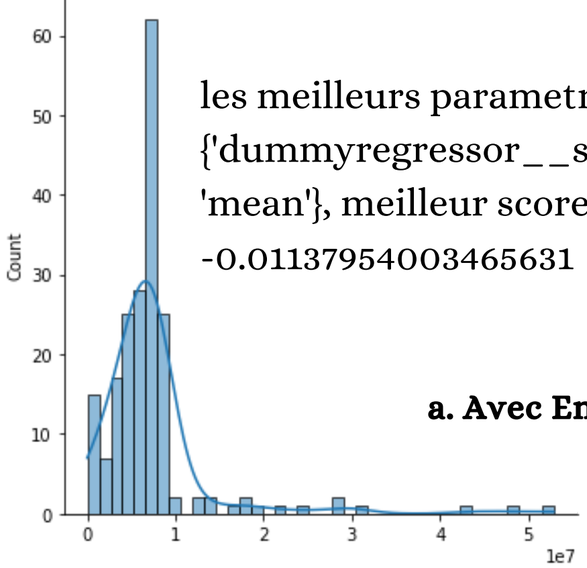
```
colonnes=[i for i in df.columns if i not in targets]
X=df[colonnes]#.apply(lambda x: str(x)).values
y=df["TotalGHGEmissions"].values
```

Modelisation

1. Model Naif DummyRegressor

Avec le GridSearchCV, nous avons
tester ce model avec les differentes stratégies suivantes:
'mean', 'median', 'max'

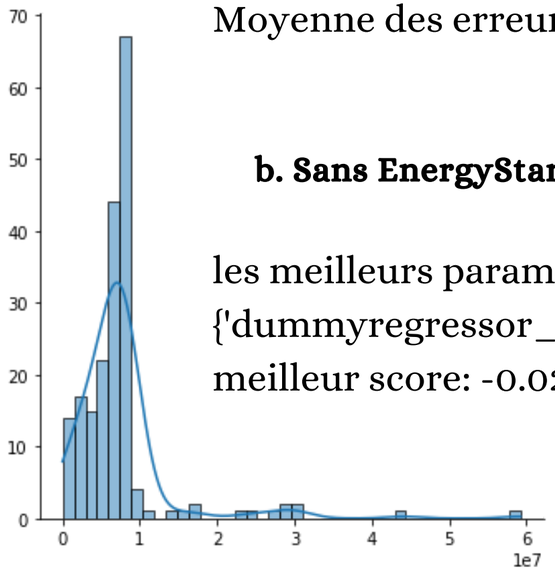
Une précision très insignifiante, ce model n'arrive pas à décrire les phénomènes à l'origine.



les meilleurs parametres du model: {'dummyregressor__strategy': 'mean'}, meilleur score: -0.01137954003465631

a. Avec EnergyStarScore

RMSE: 10026812.074345825, MAE: 7477219.285001384
Moyenne des erreur: 8752015.679673605



b. Sans EnergyStarScore

les meilleurs parametres du model: {'dummyregressor__strategy': 'mean'}, meilleur score: -0.024142262489261414

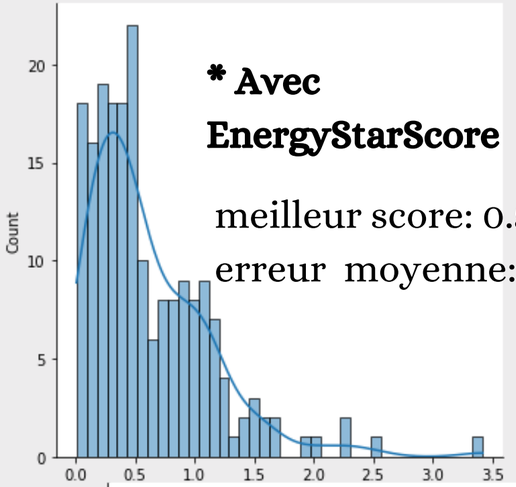
erreur totale moyenne: 30.907131113589923

Models linéaires

Nous allons utiliser Ridge, Lasso et Elasticnet avec du GridSearchCV

Nous obtenons une précision plus grande avec l'utilisation de l'EnergyStarScore

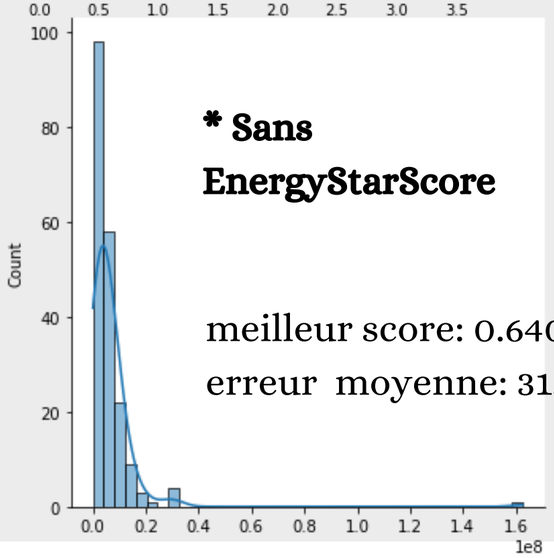
1. Ridge



*** Avec EnergyStarScore**

meilleur score: 0.5088

erreur moyenne: 30.3953

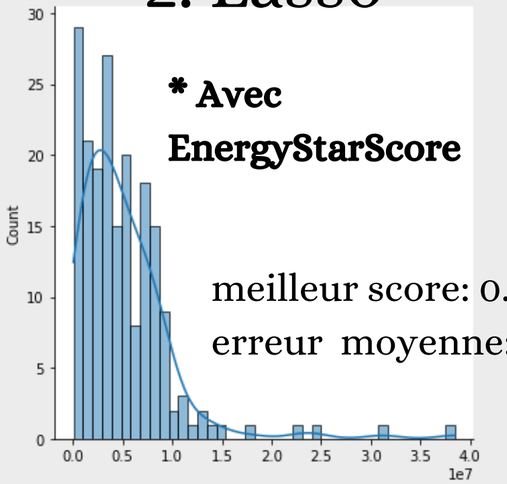


*** Sans EnergyStarScore**

meilleur score: 0.64061

erreur moyenne: 31.5

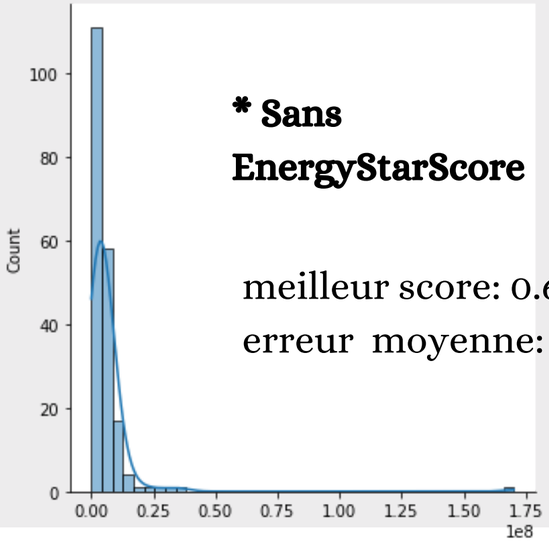
2. Lasso



*** Avec EnergyStarScore**

meilleur score: 0.625

erreur moyenne: 30.13

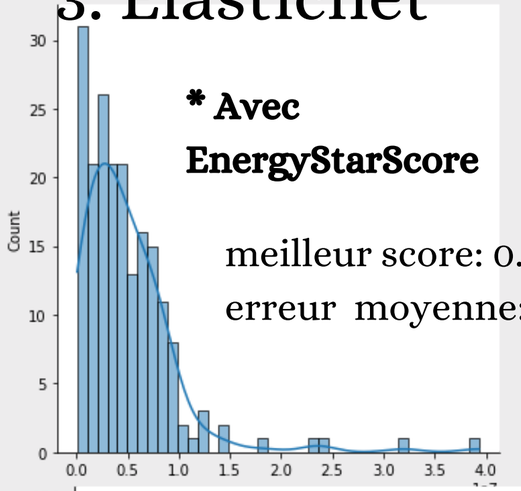


*** Sans EnergyStarScore**

meilleur score: 0.632

erreur moyenne: 31.5278

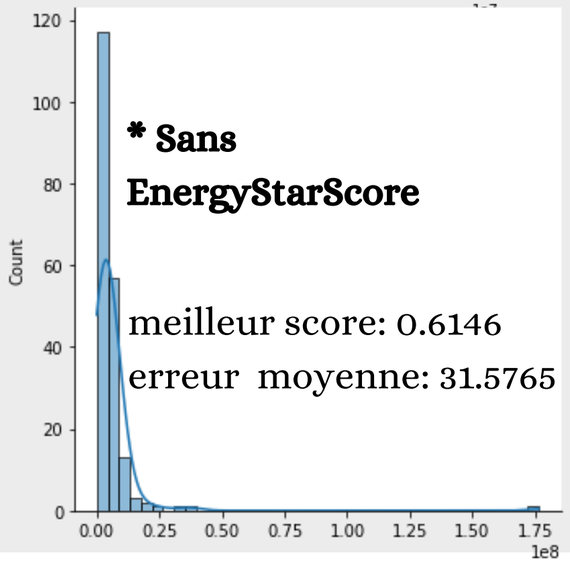
3. Elasticnet



*** Avec EnergyStarScore**

meilleur score: 0.6304

erreur moyenne: 30.1272



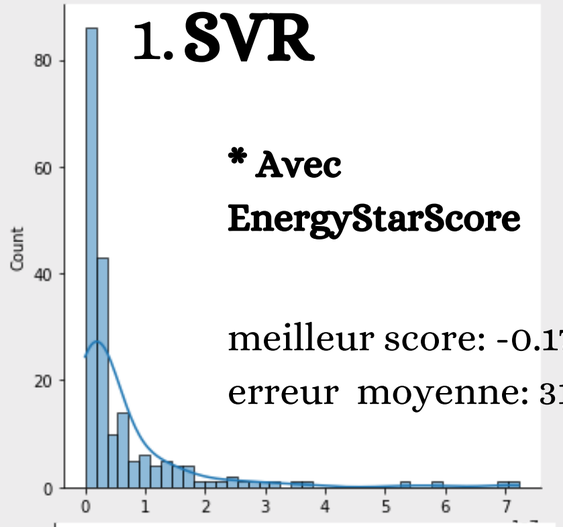
*** Sans EnergyStarScore**

meilleur score: 0.6146

erreur moyenne: 31.5765

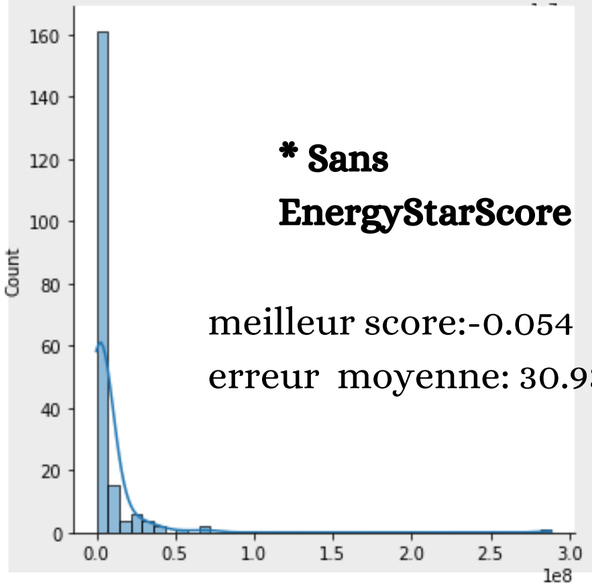
Models non linéaires

1. SVR



*** Avec EnergyStarScore**

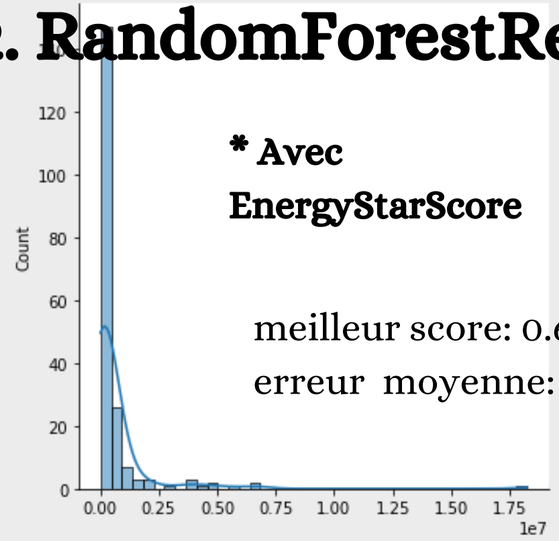
meilleur score: -0.1767
erreur moyenne: 31.28



*** Sans EnergyStarScore**

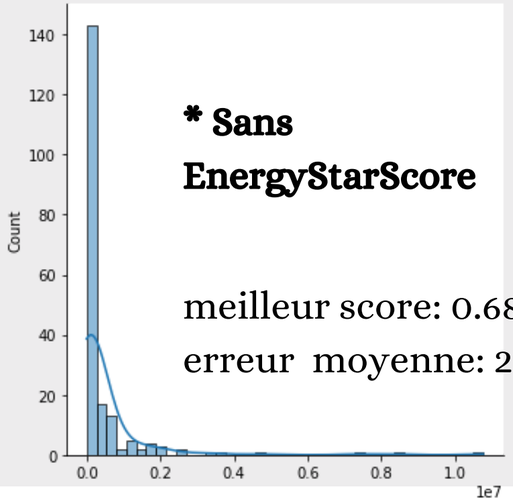
meilleur score:-0.054
erreur moyenne: 30.938

2. RandomForestRegressor



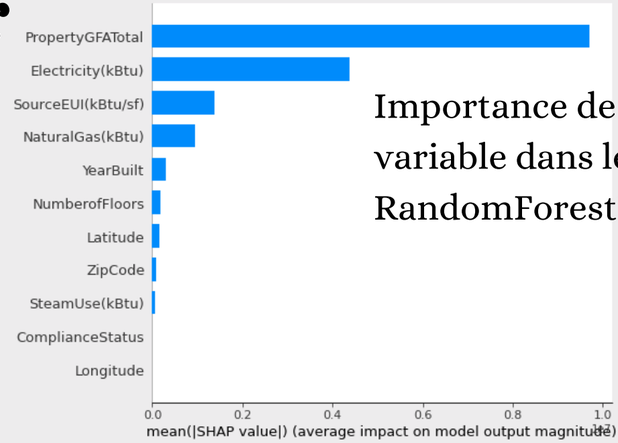
*** Avec EnergyStarScore**

meilleur score: 0.67
erreur moyenne: 27.38



*** Sans EnergyStarScore**

meilleur score: 0.684
erreur moyenne: 26.807



Importance de chaque variable dans le model RandomForestRegressor

De tous les modèles, le RandomForestRegressor sans l'EnergyStarScore est le plus performant avec un score de 68% et la moyenne d'erreur la plus basse.
n n'a donc pas besin de calculer l'EnergyStarScore qui est couteux .

Merci !