Problématique de la ville de Seattle

Anticipez les besoins en consommation électrique de bâtiments

---

Azadeh POHIER

Août 2021

Contents

Abstract

Dans ce projet nous travaillons pour la ville de Seattle. Pour atteindre son objectif de ville neutre en émissions de carbone en 2050, notre somme intéresse par des bâtiments non destinés à l'habitation.

Nous travaillons sur un projet d'apprentissage supervisé de type régression et en utilisant les modèles de Random Forest Regressor, Gradient Boosting Regressor, SVR & ExtraTreesRegressor.

Après analyser et tester tous les modelés, nous restons sur le modèle Gradient Boosting qui donne le meilleur résultat (moins d'erreurs) pour la consommation d'énergie et l'Emission CO2 également.

En fin, nous essayons de voir l'effet de score d'Énergie sur l'émission CO2.

Introduction:

La ville de Seatle souhaite atteindre son objectif de ville neutre en émissions de carbonne en 2050. Pour cela, elle souhaite dans un premier temps :

- Obtenir une prédiction des émissions de Co2 et de la consommation totale d'énergie des bâtiments pour lesquels elles n'ont pas été mesurées.
- Evaluer l'intérêt de l'"ENERGY STAR Score" pour la prédiction d'émissions.

** Notre équipe s'intéresse de près aux émissions des bâtiments **non destinés à l'habitation**.

Les données des bâtiments sont disponibles à cette adresse :

https://www.kaggle.com/city-of-seattle/sea-building-energy-benchmarking#2015-building-energy-benchmarking.csv

Des informations complémentaires sur le calcul de l'Energy Star Score sont disponibles ici :

https://www.energystar.gov/buildings/facility-owners-and-managers/existing-buildings/use-portfolio-manager/understand-metrics/how-1-100

Les informations sur les données:

- Le type de bâtiment, l'année de construction, l'usage principal du bâtiment ainsi que son emplacement et son nom
- Le nombre d'étages de la propriété, le nombre de bâtiments dans la propriété
- Les consommations annuelles en énergie
- Le score (ENERGY STAR SCORE) de la propriété qui fournit un aperçu énergétique de la performance de la propriété.

Le projet était fait en 2 partis :

1. Réaliser une courte analyse exploratoire.
   - Description des données
   - Structure des données
   - Nettoyage des données
   - Data Exploration
   - Préparation des données
     - Fuite de données
     - Valeurs aberrantes
   - Feature Engineering

2. Tester différents modèles de prédiction afin de répondre au mieux à la problématique.
   - Preprocessing
     - Pipeline
   - Sélection du modèle
     - Grid Search
   - Evaluation de model sélectionné
     - Mean absolute error (MAE)
     - Root of mean square error (RMSE)
   - Feature importances

3. Évaluer l'intérêt de l'"ENERGY STAR Score" pour la prédiction d'émissions

Partie 1 :

1.1 Description des données

Nous commençons par téléchargement de données (CSV fichiers) :

- 2015 (3340 propriétés et 47 informations)
- 2016 (3376 propriétés et 46 informations)

Apres vérification de data.info(), on trouve des colonnes avec beaucoup des valeurs manquants que on va les supprimer pour les deux données.

```
data_2015.drop(['Comment','Outlier','YearsENERGYSTARCertified','City Council Districts',
                '2010 Census Tracts','ThirdLargestPropertyUseTypeGFA','ThirdLargestPropertyUseType'],axis=1,
               inplace=True)
```

```
data_2016.drop(['Comments','Outlier','YearsENERGYSTARCertified','ThirdLargestPropertyUseTypeGFA',
                'ThirdLargestPropertyUseType'],axis=1,inplace=True)
```

Ensuite, on va essayer de comparer les noms des colonnes entre les deux data sets, il faut garder les mêmes colons et corriger les noms de colon qu'ils ont le même sens.

```
# On renomme les colonnes de 2015 pour que les noms des variables soient les mêmes que pour 2016 (Energy et CO2)
data_2015.rename(columns={'GHGEmissions(MetricTonsCO2e)': 'TotalGHGEmissions',
                          'GHGEmissionsIntensity(kgCO2e/ft2)': 'GHGEmissionsIntensity',
                          'Zip Codes':'ZipCode'}, inplace=True)
```

La colonne '**Location'** en base de données 2015 est inclus: address, state , zip, longitude, latitude and… donc, nous pouvons la séparer dans en 'latitude' et 'longitude' et après supprimer cette colonne.

```
import ast
data_2015['Latitude']=data_2015.Location.apply(lambda x:ast.literal_eval(x)['latitude'])
data_2015['Longitude']=data_2015.Location.apply(lambda x:ast.literal_eval(x)['longitude'])
```

```
data_2015.drop('Location', axis=1, inplace=True)
```

Enfin, supprimassions des colonnes pas intéressante dans les deux datasets.

```
data_2015.drop(['OtherFuelUse(kBtu)', 'Seattle Police Department Micro Community Policing Plan Areas',
 'SPD Beats'], axis=1, inplace= True)
```

```
data_2016.drop(['Address', 'City', 'State'],inplace=True, axis=1)
```

après toutes les étapes que nous avons faites, maintenant on a les 2 data sets avec les même colonnes et on peut utiliser le ' concat ' pour concaténer les deux donnée pour avoir un seul jeu de données (6716 lignes et 38 colonnes).

1.2 Pré-nettoyage des données

D'abord, on commence par vérification des variables dupliquées entre les deux colonnes :

```
data.duplicated(['OSEBuildingID','DataYear']).sum()
```

```
0
```

Le résultat égal à zéro signifie qu'il n'y a pas des valeurs en double. Nous pouvons maintenant commencer notre partie de nettoyage des données.

Sélection de caractéristique

Dans le cadre de nos modélisations, les variables à prédire sont la **consommation d'énergie du bâtiment** (SiteEnergyUse(kBtu)) et ses **émissions de CO2** (TotalGHGEmissions).

Certaines colonnes ne sont pas utiles pour notre projet, nous allons donc les supprimer :

```
data.drop(['OSEBuildingID','PropertyName','TaxParcelIdentificationNumber','CouncilDistrictCode',
           'DefaultData','ComplianceStatus','ZipCode'], axis=1, inplace= True)
```

Informations sur les caractéristiques:

- Electricity(kBtu): The annual amount of electricity consumed by the property on-site, including electricity purchased from the grid and generated by onsite renewable systems, measured in thousands of British thermal units (kBtu).
- Electricity(kWh): The annual amount of electricity consumed by the property on-site, including electricity purchased from the grid and generated by onsite renewable systems, measured in kWh.
- NaturalGas(therms): The annual amount of utility-supplied natural gas consumed by the property, measured in therms.
- NaturalGas(kBtu): The annual amount of utility-supplied natural gas consumed by the property, measured in British thermal unit(s).

1 therm equals 100,000 Btu. Donc on va garder jute les colonnes avec l'échelle de (kBtu).

```
#Suppression des variables redondantes
redundant_features = ['NaturalGas(therms)','Electricity(kWh)']
data.drop(redundant_features, axis=1, inplace=True)
```

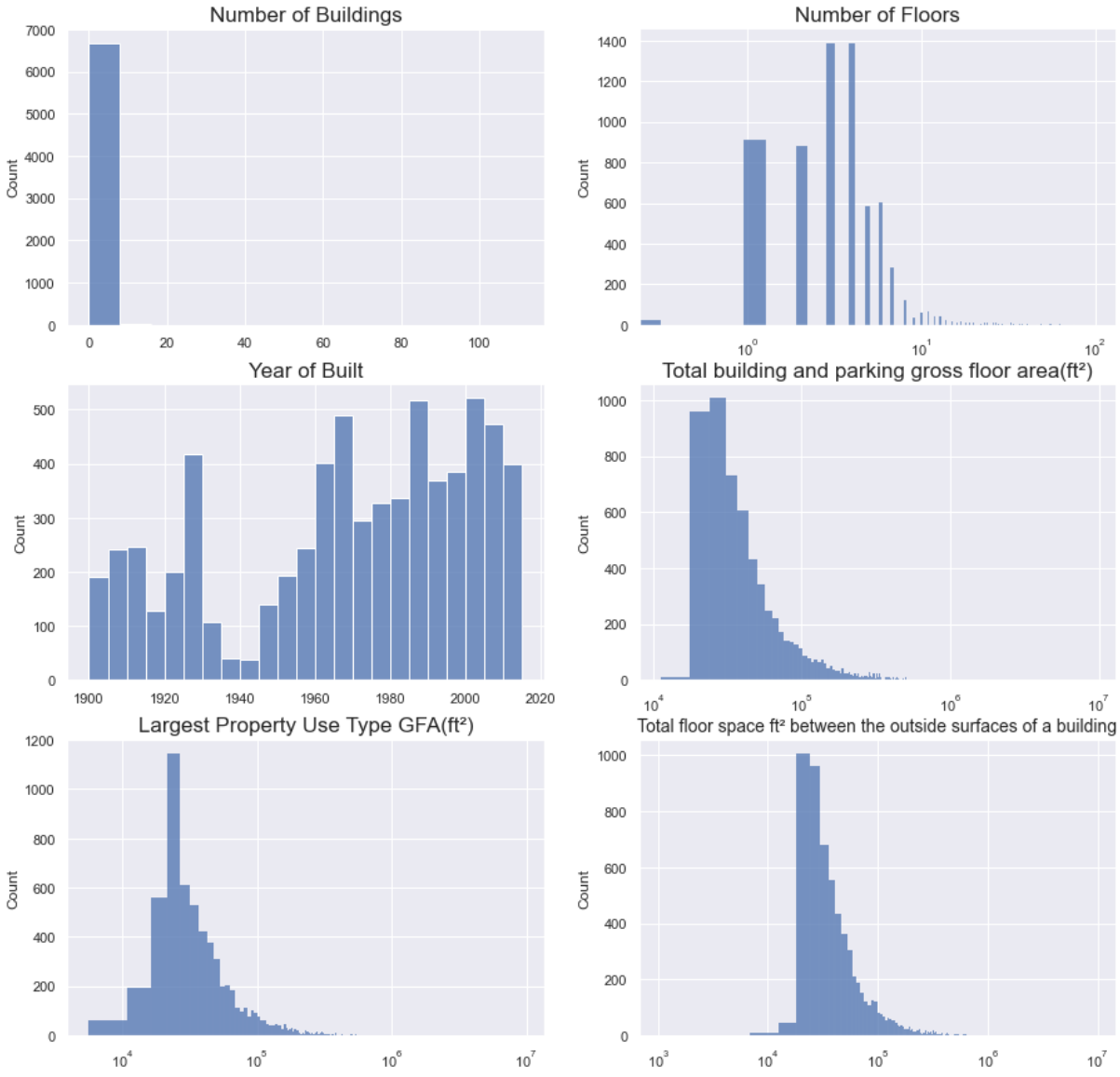
Ensuite, comme notre objective est analyser les variables de consommation d'énergie du bâtiment (SiteEnergyUse(kBtu)) et les émissions de CO2 (TotalGHGEmissions), nous gardons l'observation non nulle.

```
data = data[~((data['SiteEnergyUse(kBtu)'].isnull()) | (data['TotalGHGEmissions'].isnull()))]
```

1.3 Analyse exploratoire

Variables numérique:

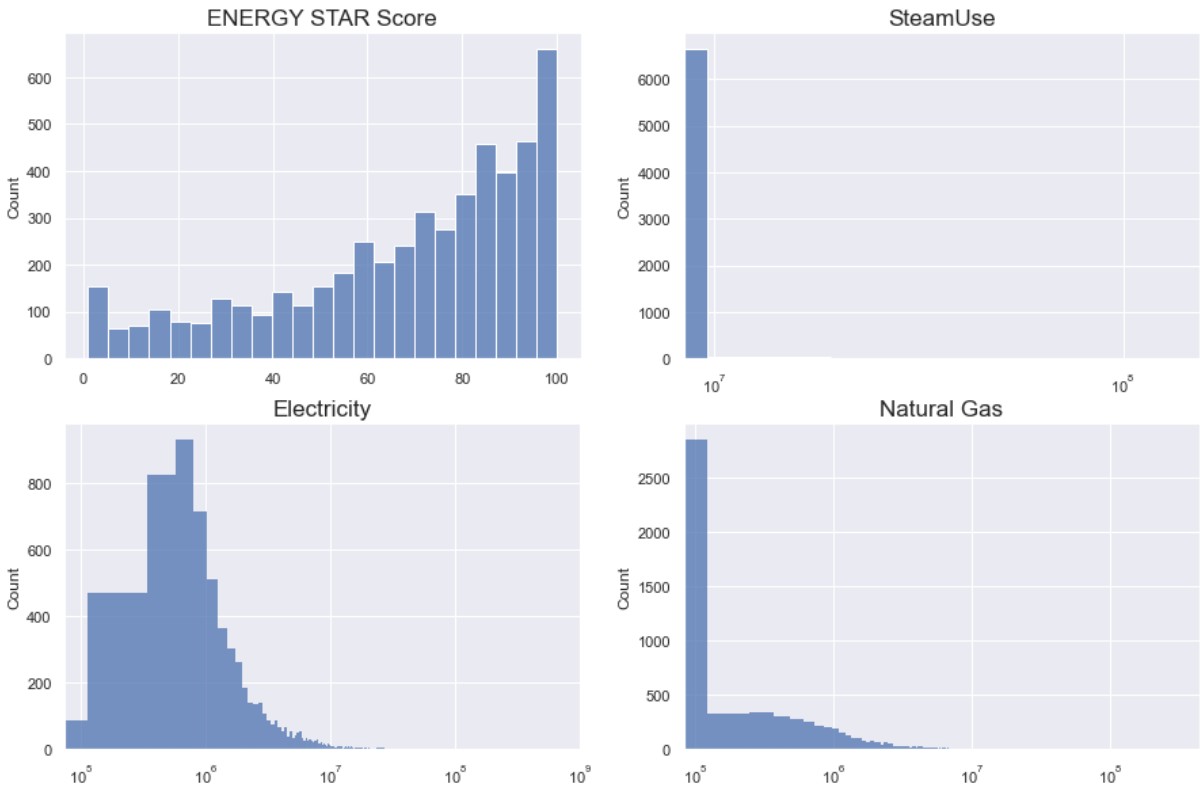
— Visualisation des propriétés ;



Pour rendre les graphiques plus clairs, nous avons utilisé la fonction log.

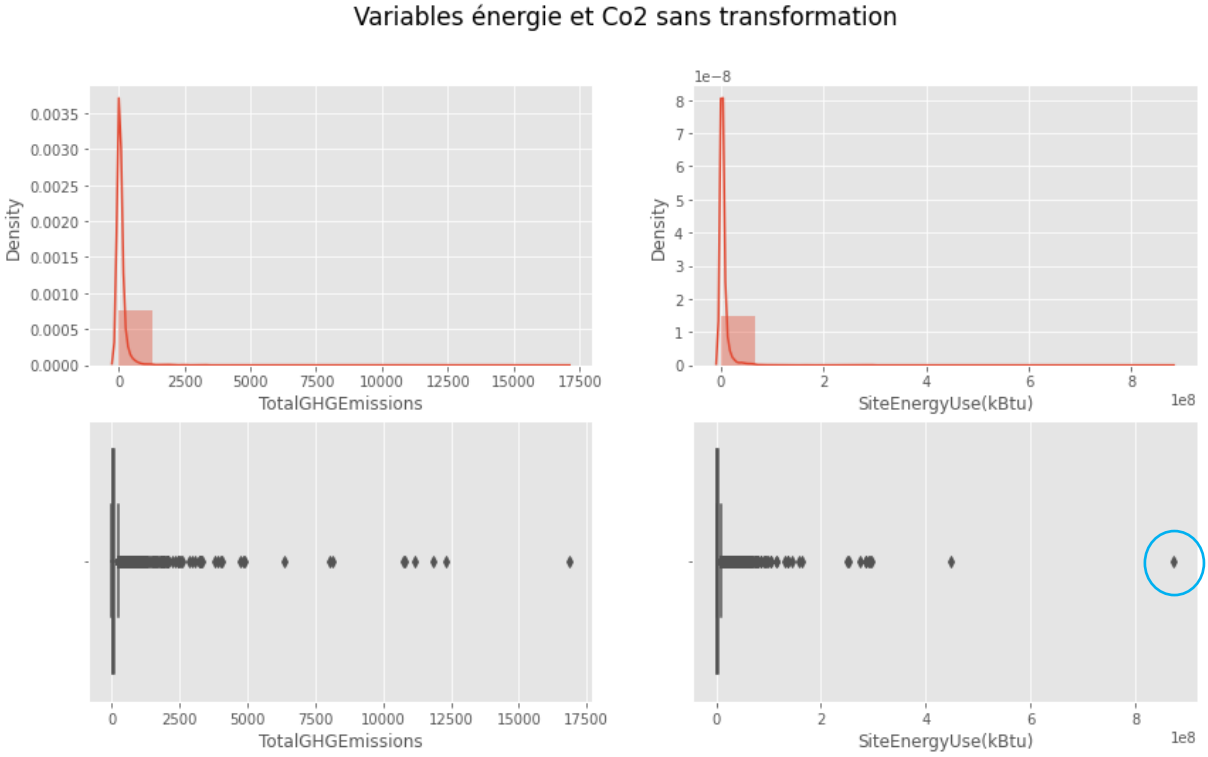
- Les bâtiments ont été construits entre 1900 et 2020, donc leurs âges sont compris entre 0 et 120 ans.
- Les superficies des propriétés sont comprises entre 10,000 et 100,000.

— Visualisation de l'énergie;

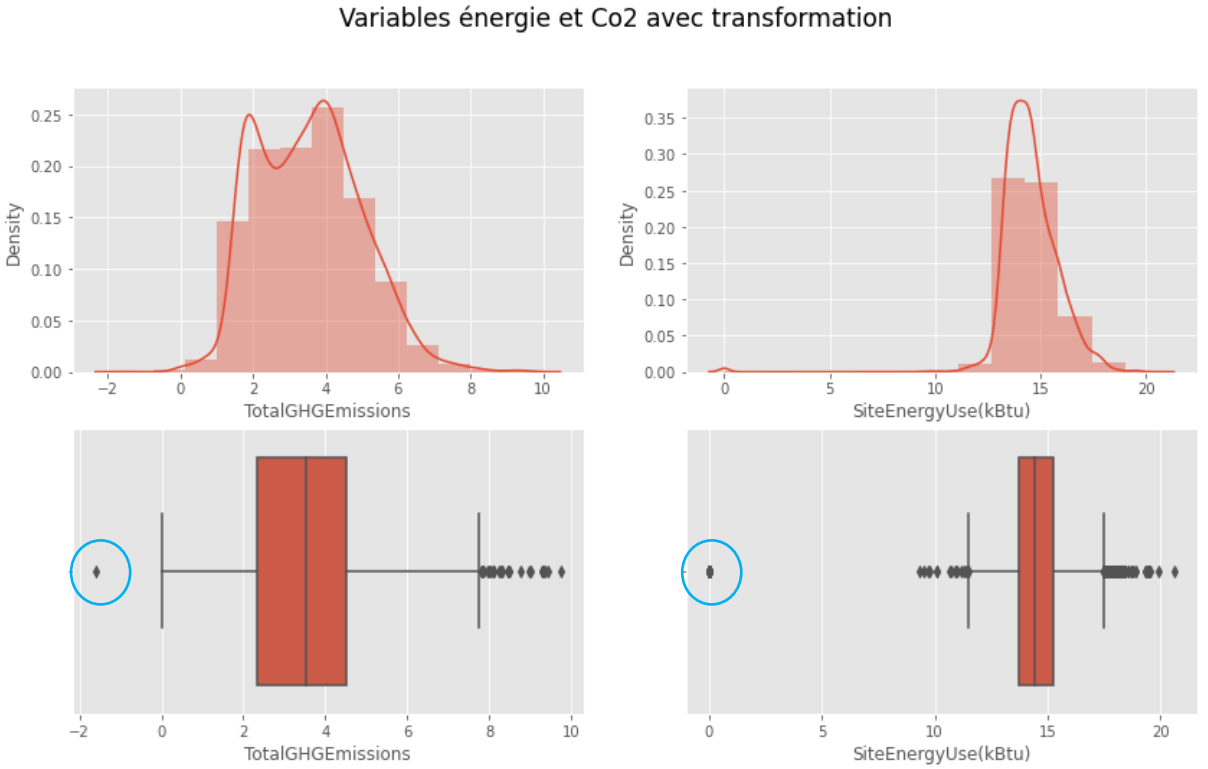


L'Energy score pour les bâtiments est entre 0 et 100, normalement, les bâtiments qui consomment le plus d'énergie obtiennent un score très faible.

— CO2 & Energy:

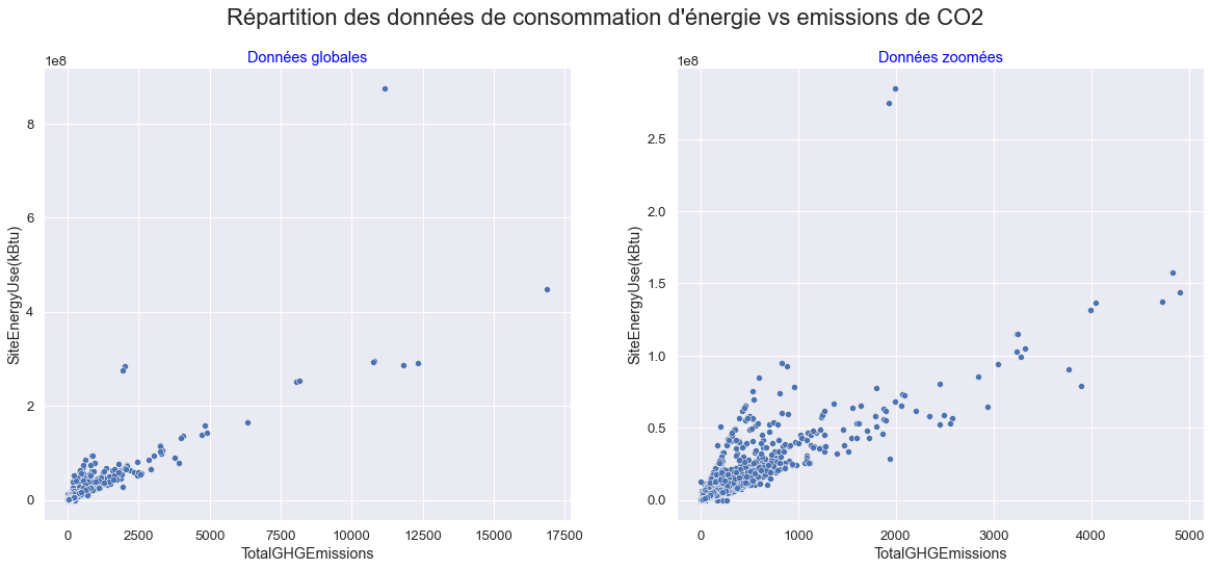


Certains bâtiments sont des outliers en consommation d'énergie, il y a un bâtiment qui a une consommation supérieur à 1e8 kBtu.



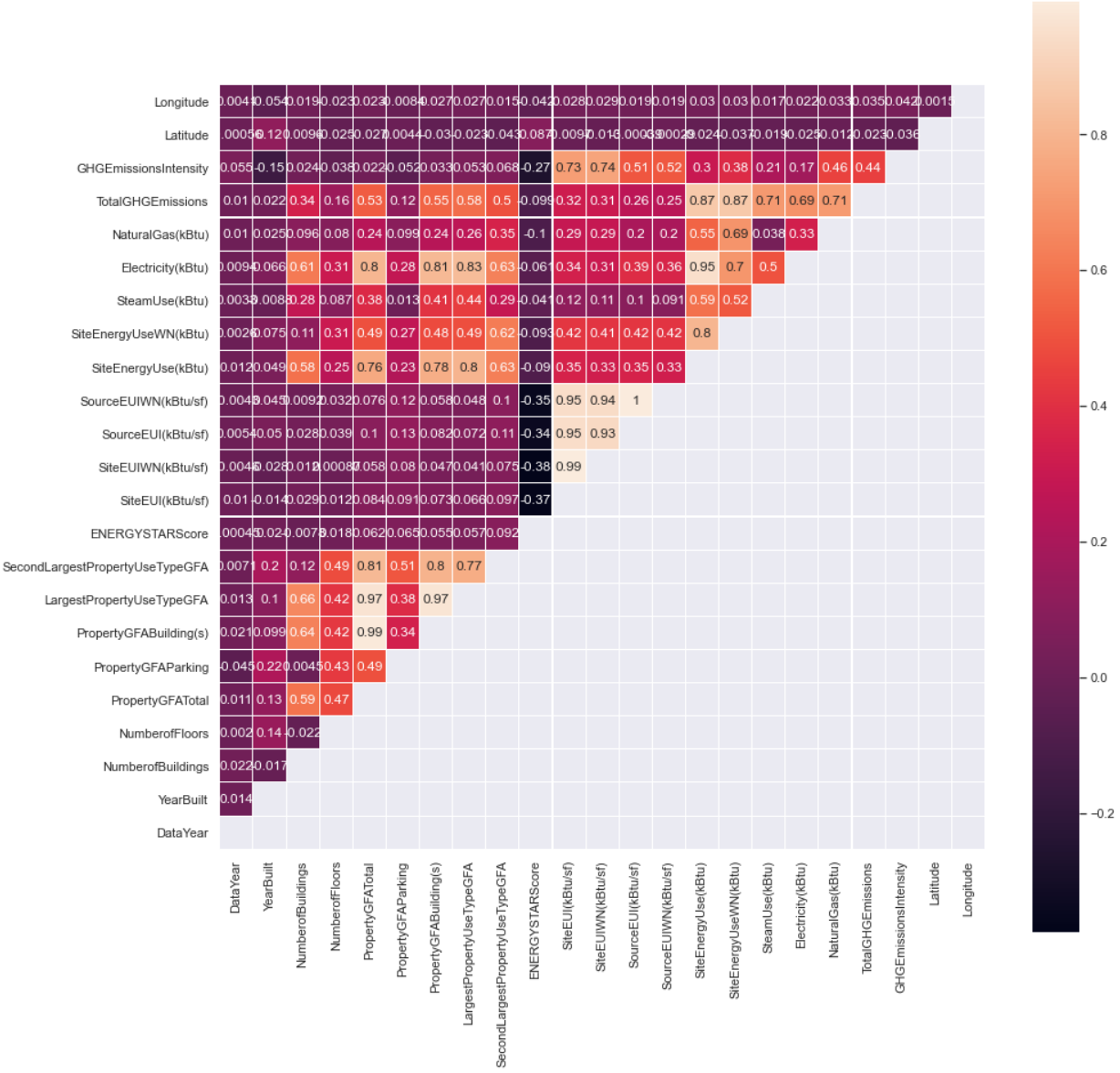
Il y a les valeurs aberrantes avec des valeurs négatives.

Analyse bivariée entre consummation d'energie et emission de CO2 ;



On remarque ici que la répartition des données d'émission de CO2 en fonction de la consommation d'énergie ne suivent pas uniquement 1 seule droite de régression linéaire si l'on zoom sur les données les plus représentées.

Matrice de corrélation



Corrélation entre les variables :

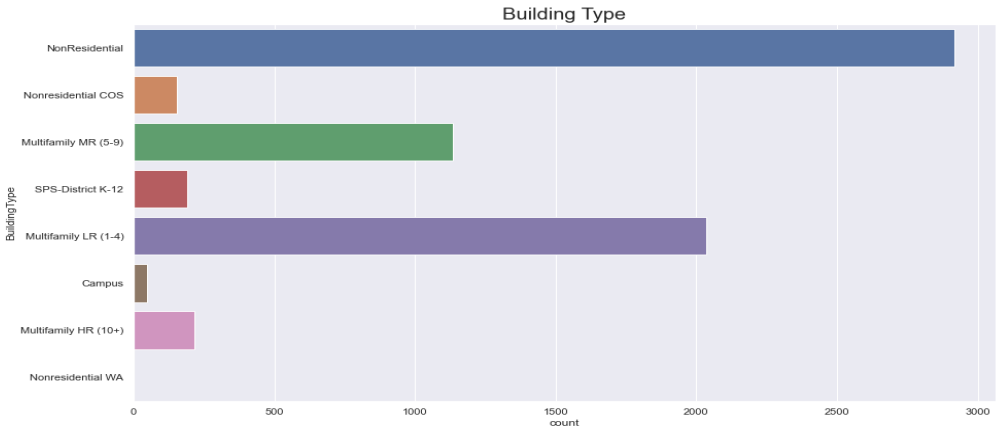
- Le "DataYear" est fortement corrélée à la "ZipCode".
- Le 'SourceEUIWN(kBtu/sf)' est fortement corrélée à les 'SiteEUI(kBtu/sf)', 'SiteEUIWN (kBtu/sf)' et 'SourceEUI(kBtu/sf)'.

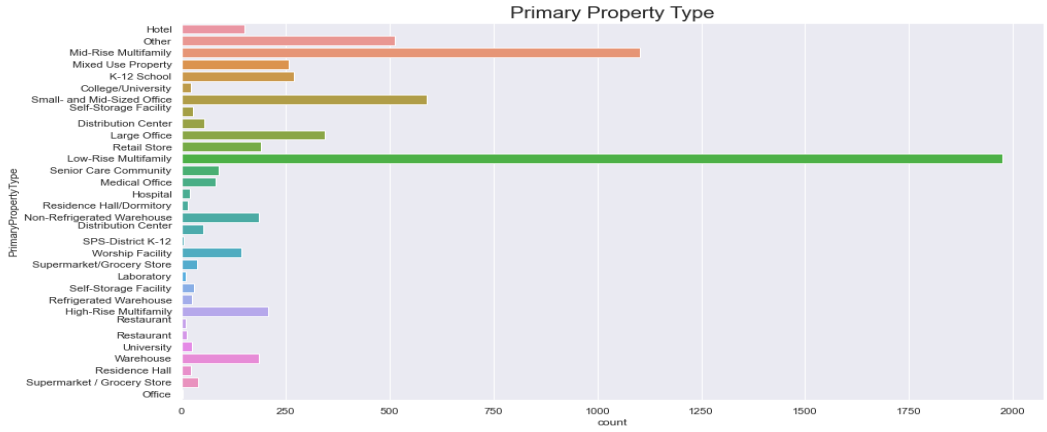
La consommation totale d'énergie "Electricity(kBtu)" est fortement corrélée à la surface des bâtiments "PropertyGFABuilding(s)", "PropertyGFATotal" & "Largest PropertyUseTypeGFA".

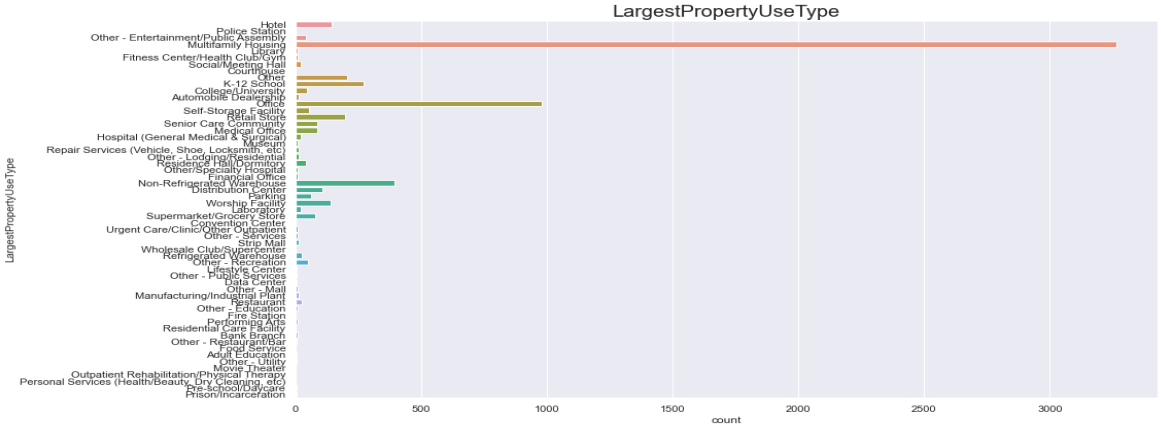
- Les émissions de Co2 sont fortement corrélées à la consommation totale d'énergie.

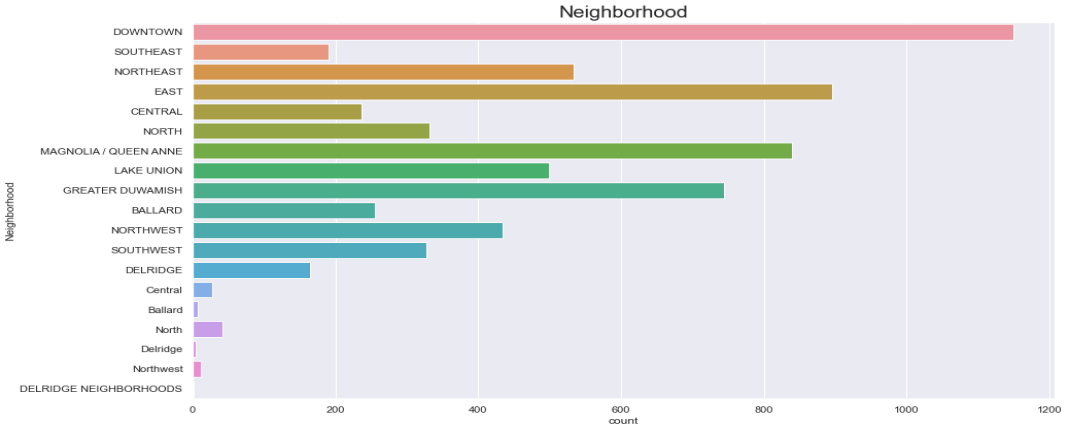
On va supprimer les variables avec des corrélations fortes pour éviter la redondance d'informations.

Maintenant, on continue notre analyse exploratoire pour les variables qualitatives :







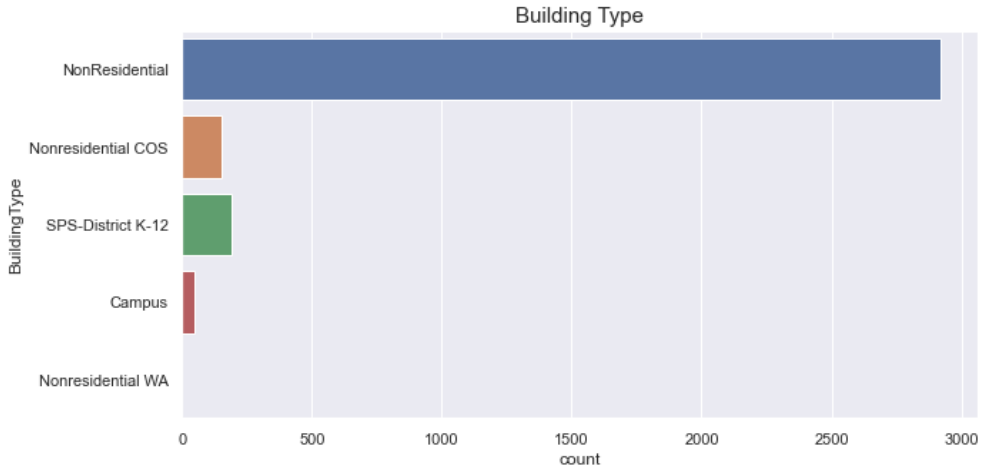


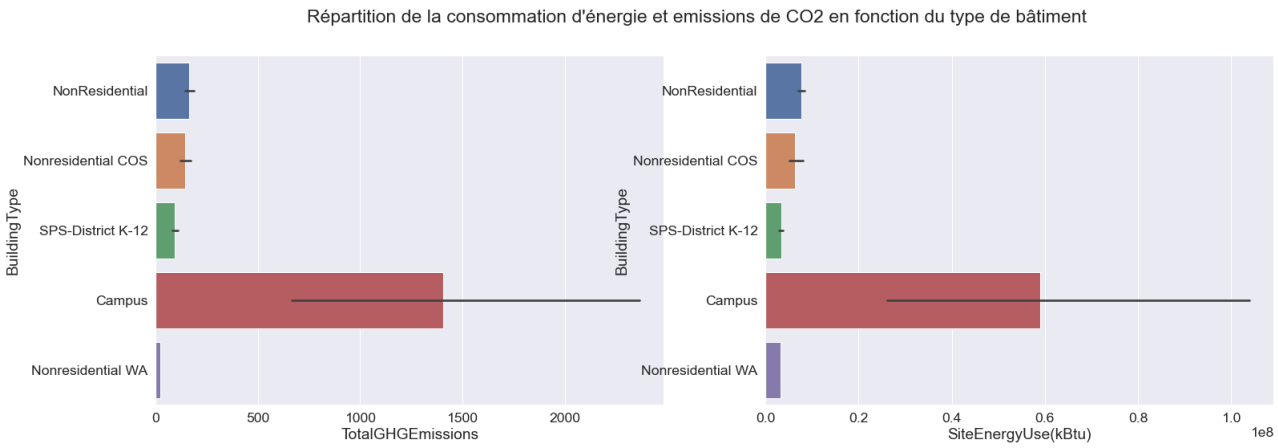
Nous devons manipuler notre ensemble de données car il y a plus d'observations qui sont répétées dans la même feature.

Par exemple dans le colon ‘Neighbors’ il y a ‘CENTRAL’ et ‘central’.

1.4 Préparation des données

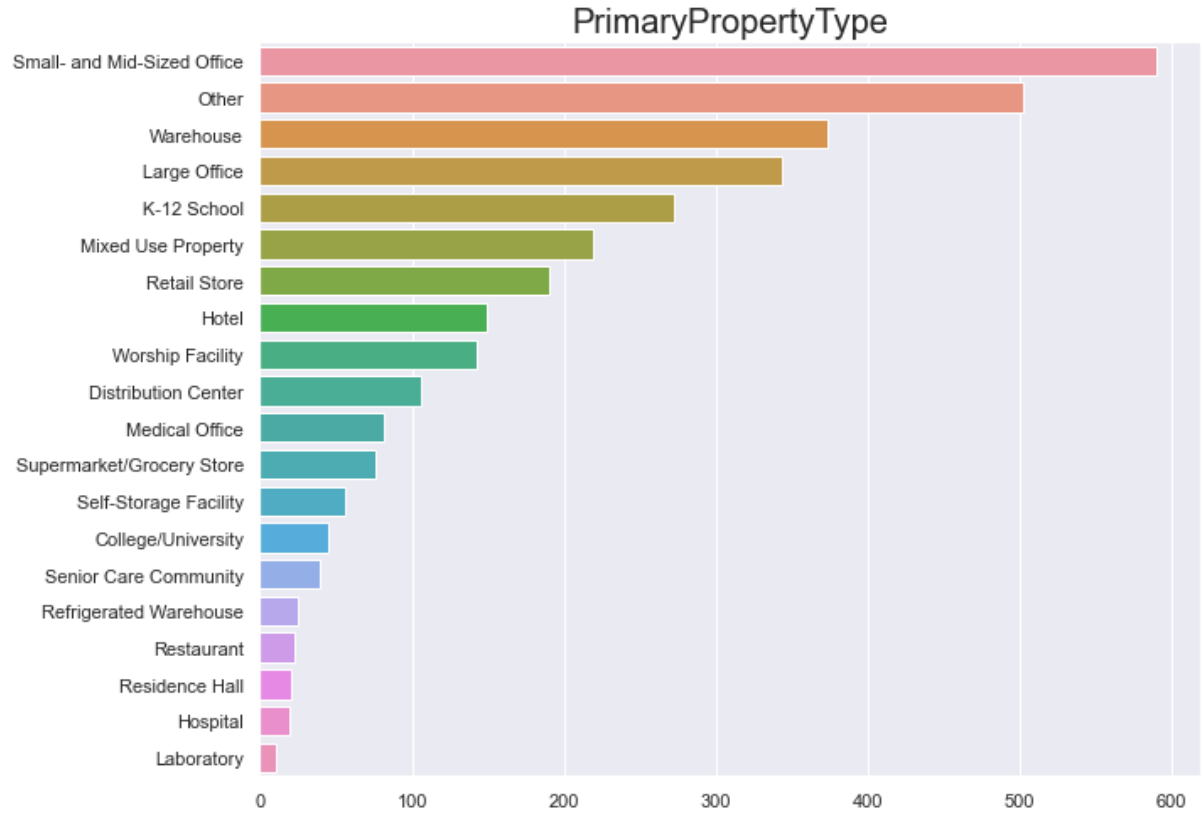
L'objectif de nos émissions est de prédire les émissions de CO2 et la consommation énergétique totale des bâtiments non résident. Donc, de la colonne ‘BuildingType’ on garde les bâtiments non résident.

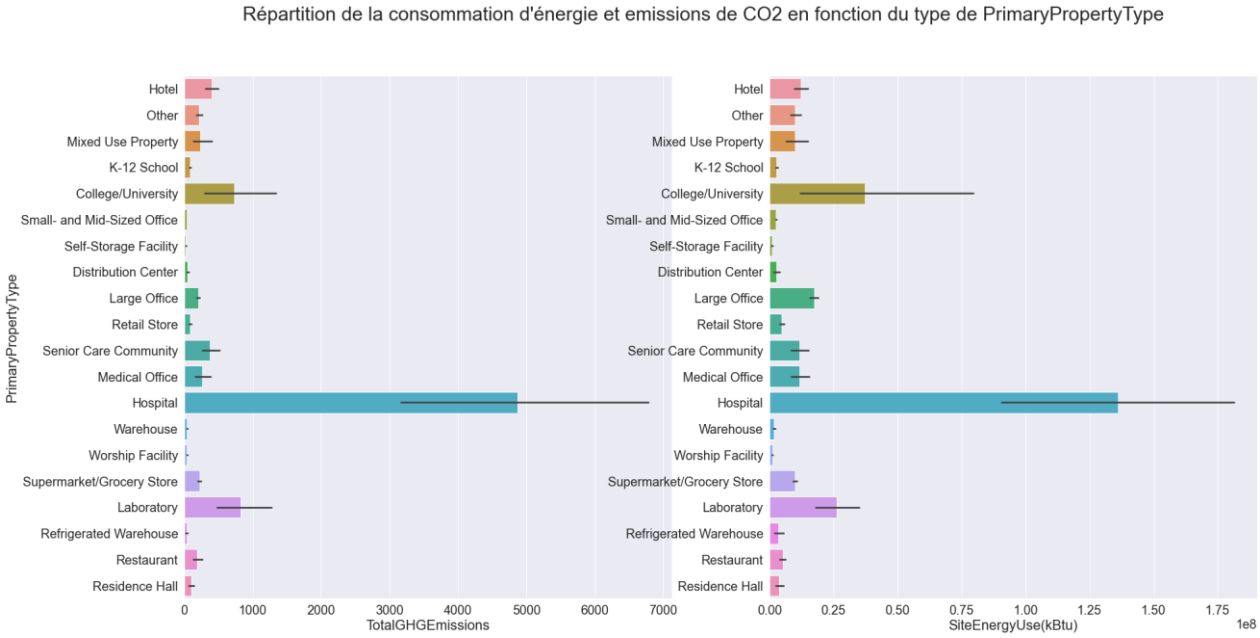




La consommation d'énergie est plus consommée dans la partie campus et également pour émission de co2.

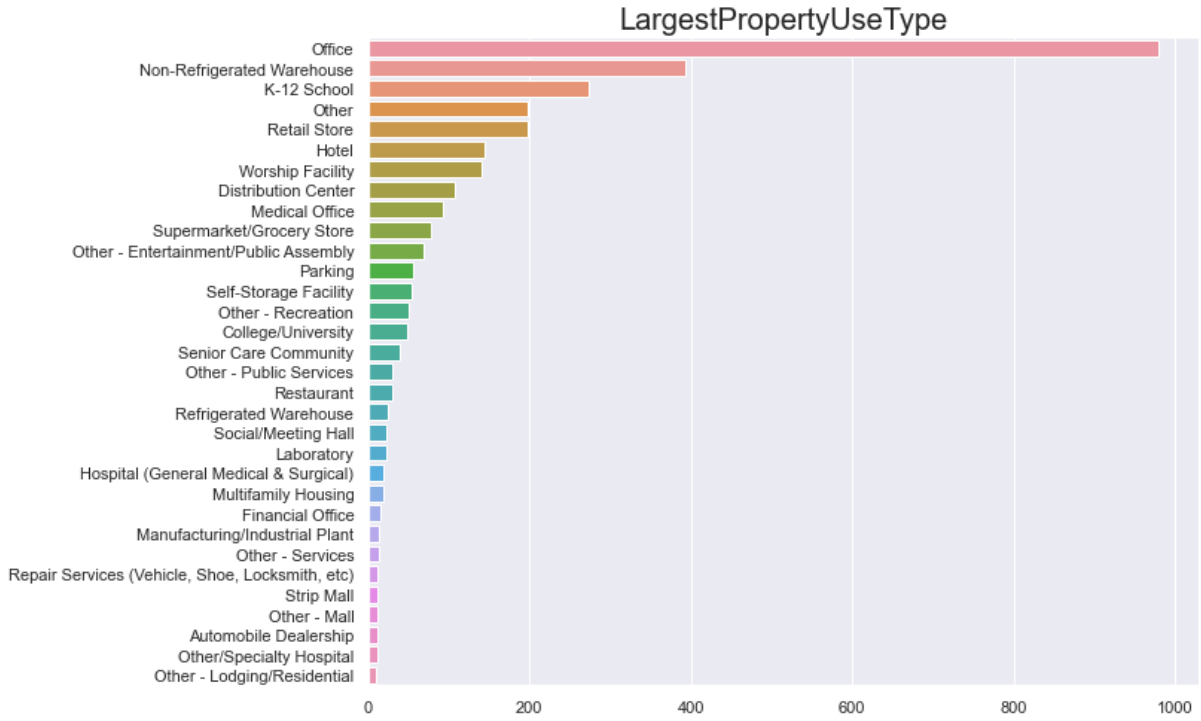
Regroupement des données pour ‘PrimaryPropertyType’ :



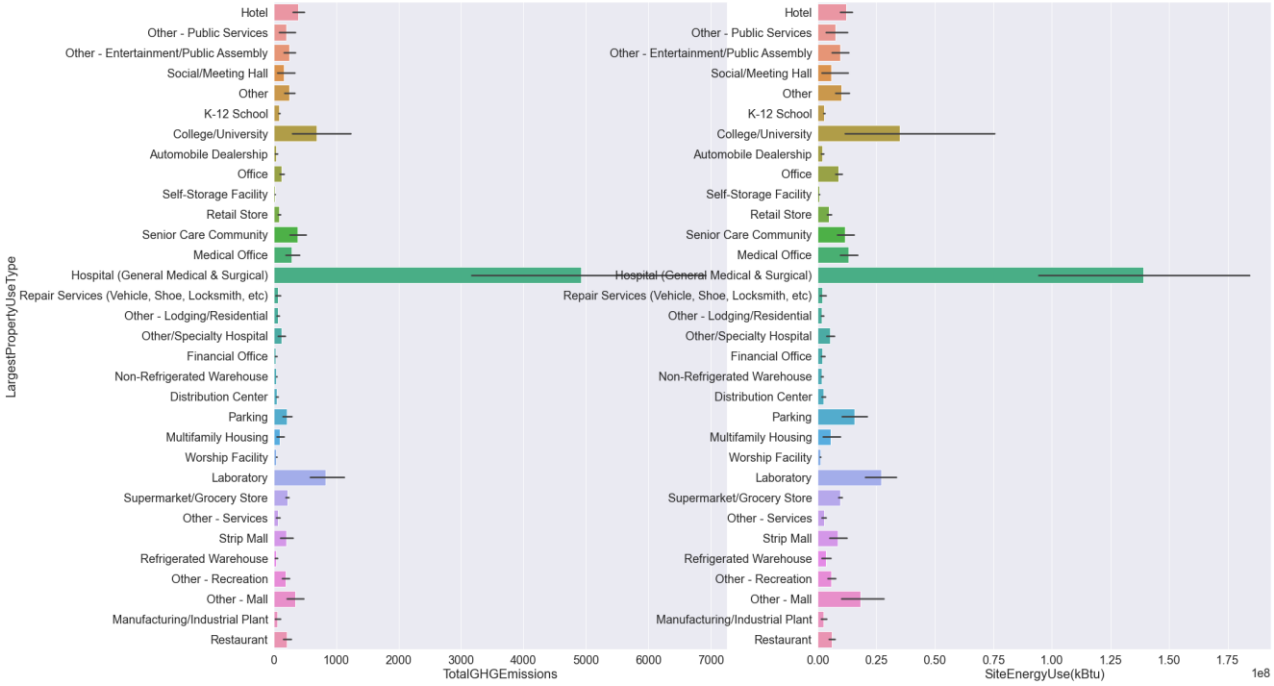


La consommation d'énergie est plus consommée par les hôpitaux.

Regroupement des données pour ‘LargestPropertyUseType’:

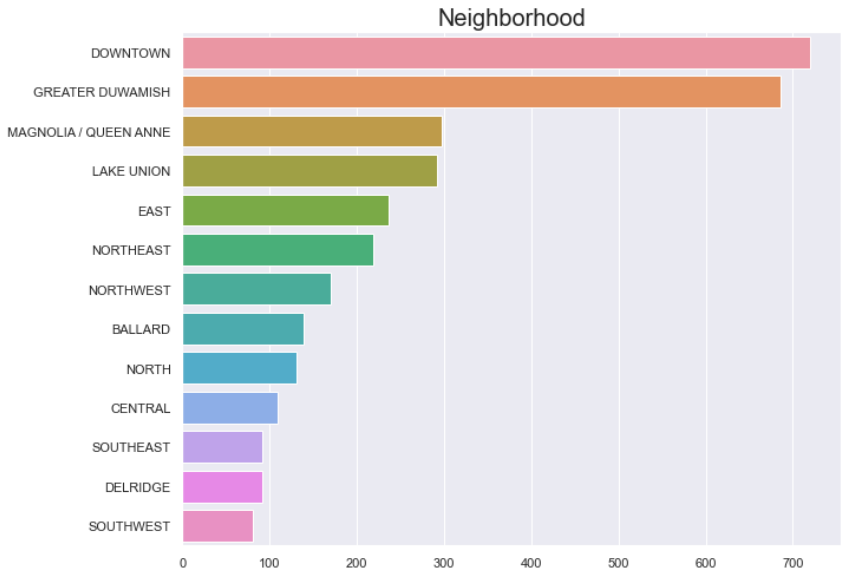


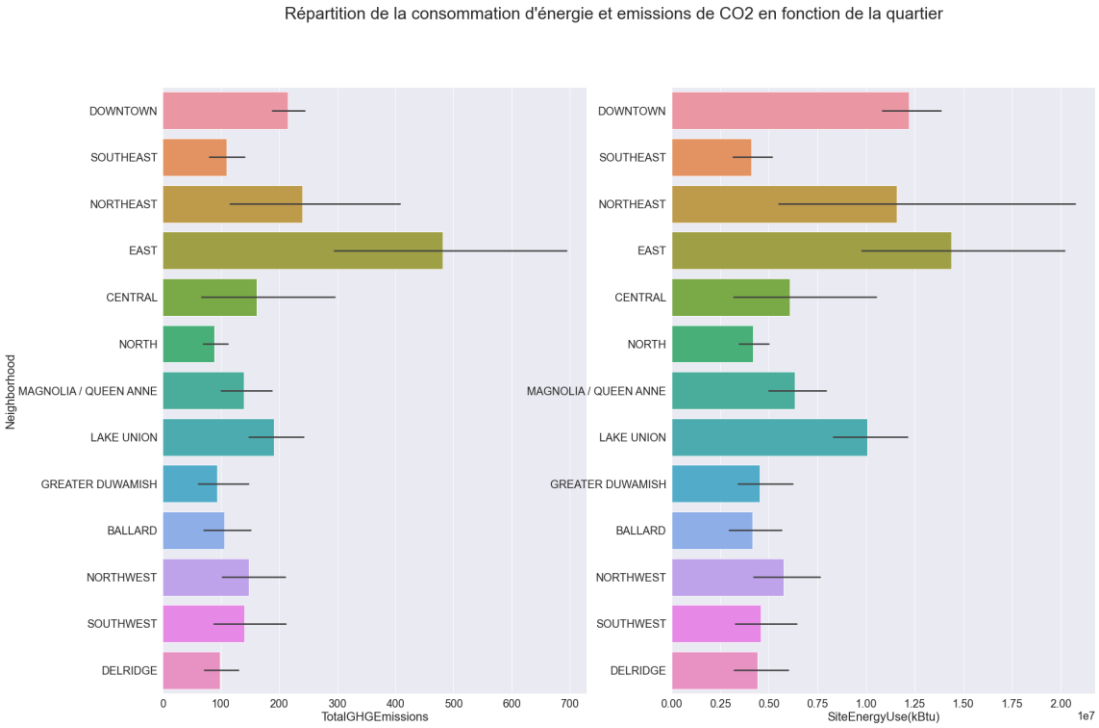
Répartition de la consommation d'énergie et emissions de CO2 en fonction du type de PrimaryPropertyType



Avec comparaison les deux graphes, ‘LargestPropertyUseType’ et ‘PrimaryPropertyType ‘, on peut avoir presque les mêmes informations sur les données, donc on peut supprimer le colon ‘LargestPropertyUseType’.

Regroupement des données pour ‘Neighborhood’:



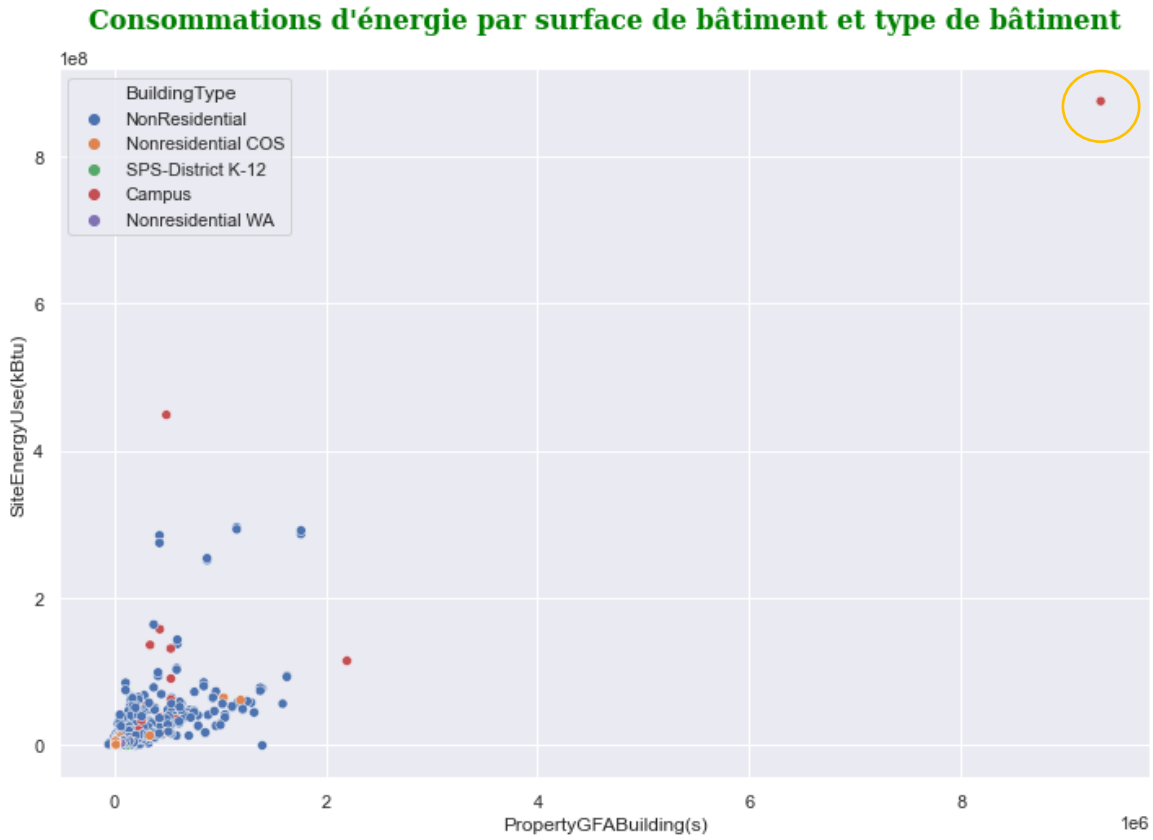


La consommation d'énergie est plus consommée dans la partie ‘East, Northeast et Downtown’ et également pour émission de co2.

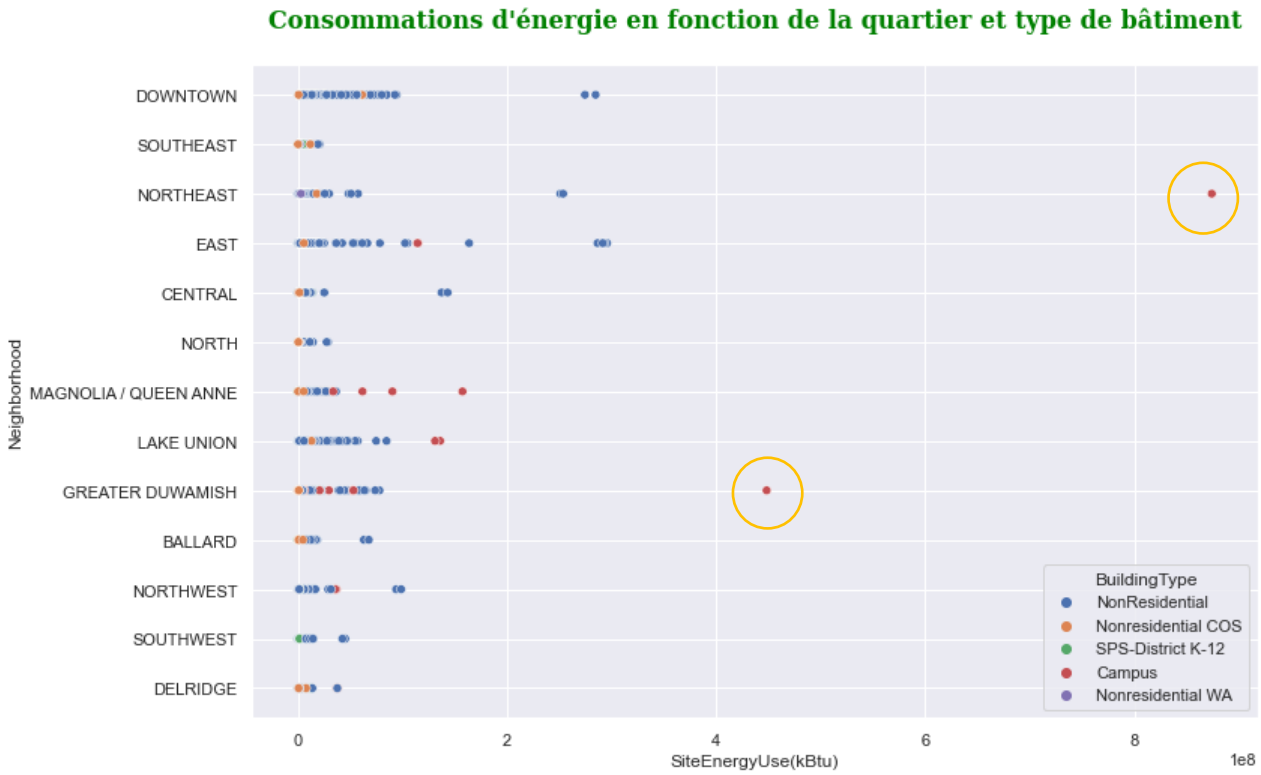
Valeurs aberrantes ;

	DataYear	YearBuilt	NumberofBuildings	NumberofFloors	PropertyGFAParking	PropertyGFABuilding(s)	ENERGYSTARScore	SiteEnergyUse(kBtu)	Tota
count	3264.000000	3264.000000	3264.000000	3256.000000	3264.000000	3.264000e+03	2178.000000	3.264000e+03	
mean	2015.502451	1961.636642	1.119792	4.136057	13463.986826	1.031731e+05	64.851699	8.136796e+06	
std	0.500071	32.678736	2.237244	6.610812	43918.521313	2.358556e+05	28.590074	2.541927e+07	
min	2015.000000	1900.000000	0.000000	0.000000	-2.000000	-5.055000e+04	1.000000	0.000000e+00	
25%	2015.000000	1930.000000	1.000000	1.000000	0.000000	2.858550e+04	47.000000	1.219834e+06	
50%	2016.000000	1965.000000	1.000000	2.000000	0.000000	4.763750e+04	72.500000	2.515115e+06	
75%	2016.000000	1989.000000	1.000000	4.000000	0.000000	9.455325e+04	89.000000	6.992292e+06	
max	2016.000000	2015.000000	111.000000	99.000000	512608.000000	9.320156e+06	100.000000	8.739237e+08	

En examinant le tableau de description, nous voyons qu'il y a des valeurs négatives dans nos données pour la superficie des bâtiments qui sont considérés comme des valeurs aberrantes.



Ici, il y a une valeur aberrante pour l'énergie qui appartient au campus.



Dans ce graphique, il y a deux valeurs aberrantes qui appartiennent au campus.

```
data.shape
```

```
(3264, 15)
```

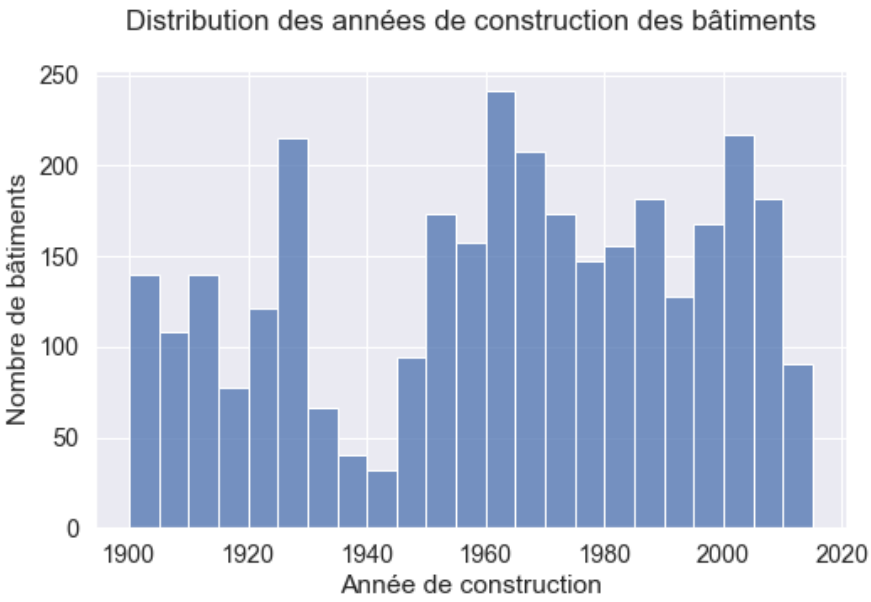
```
data_NO_out = data[~((data.PropertyGFAParking < 0) | (data.PropertyGFABuilding < 0) |
                     (data.PropertyGFABuilding > 4e+6)|(data.TotalGHGEmissions < 0) |
                     (data.SiteEnergyUse > 4e+8))]
```

```
data_NO_out.shape
```

```
(3258, 15)
```

Enfin nous supprimons 6 valeurs aberrantes.

1.5 Feature Engineering

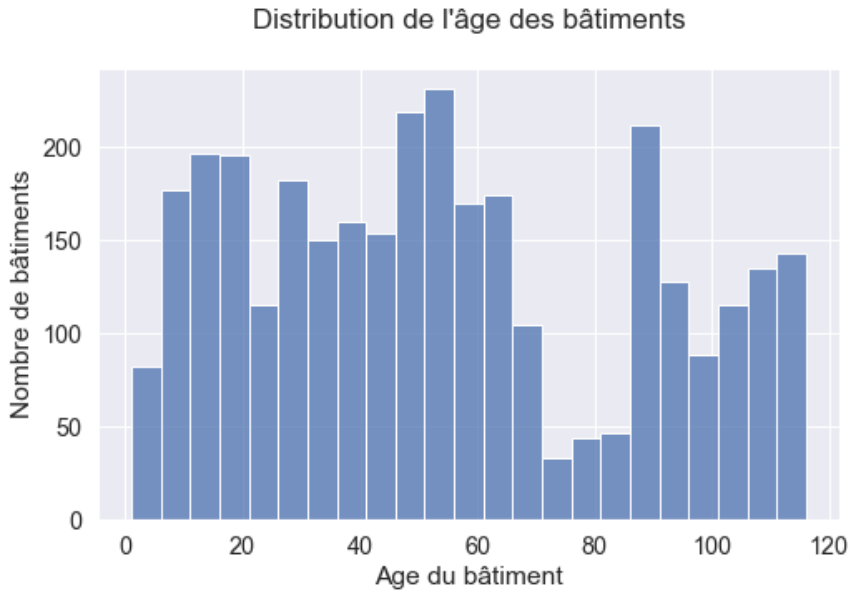


Plus que l'année de construction, il serait intéressant de traiter l'Age des bâtiments pour réduire la dispersion des données et lier l'année des relevés.

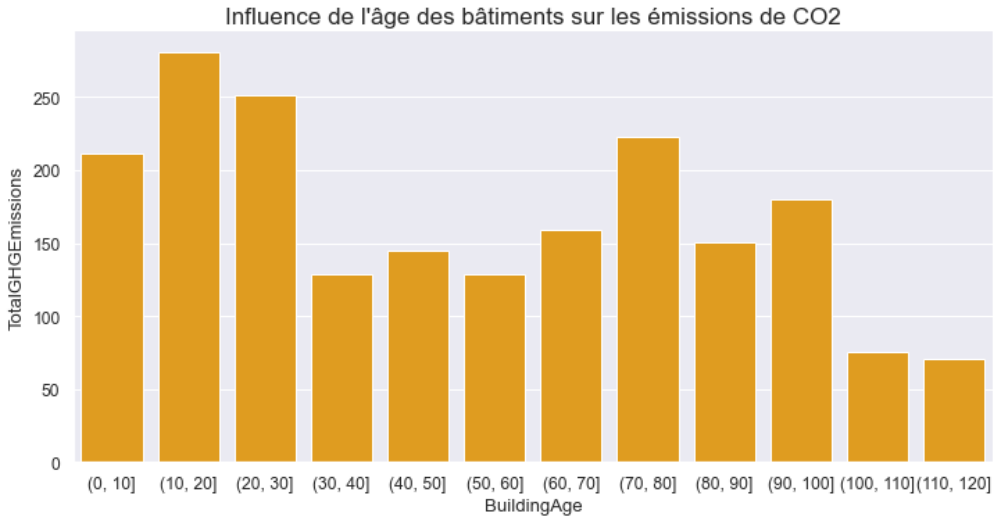
Nous allons donc créer cette nouvelle variable et supprimer l'année de construction :

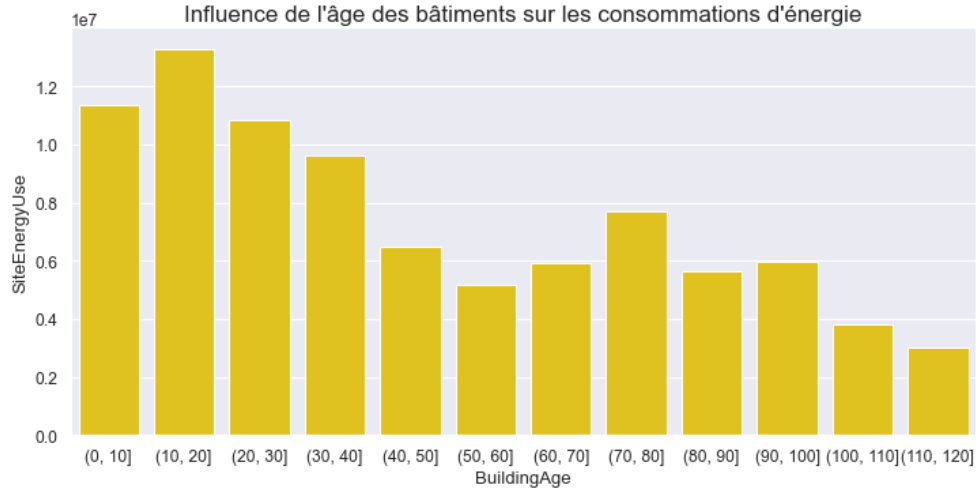
```
sns.set(font_scale=1.3)
new_df['BuildingAge'] = new_df['DataYear'] - new_df['YearBuilt']
new_df.drop(['YearBuilt','DataYear'], axis=1, inplace=True)

fig = plt.figure(figsize=(8,5))
ax = sns.histplot(data=new_df, x='BuildingAge', bins=int((new_df.BuildingAge.max() - new_df.BuildingAge.min())/5))
ax.set_xlabel("Age du bâtiment")
ax.set_ylabel("Nombre de bâtiments")
plt.title("Distribution de l'âge des bâtiments\n",  fontsize=17)
```



Relation entre « énergie & co2 » et « l'âge des bâtiments & energy score» :





Les bâtiments avec l'âge de 10-20 consument plus de énergie et produisent plus de CO2.

Note: La consumation d'énergie dépend d'autres éléments aussi comme surface des bâtiments.

Data Leakage

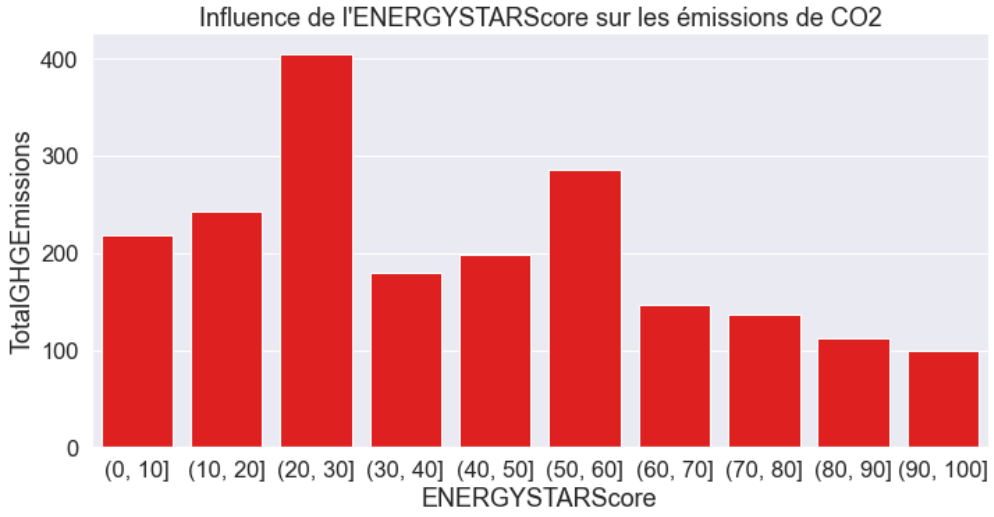
Une fuite de données se produit lorsque les données d'entraînement contiennent des informations sur la cible, mais des données similaires ne seront pas disponibles lorsque le modèle est utilisé pour la prédiction. Cela conduit à des performances élevées sur l'ensemble d'apprentissage (et peut-être même sur les données de validation), mais le modèle fonctionnera mal en production.

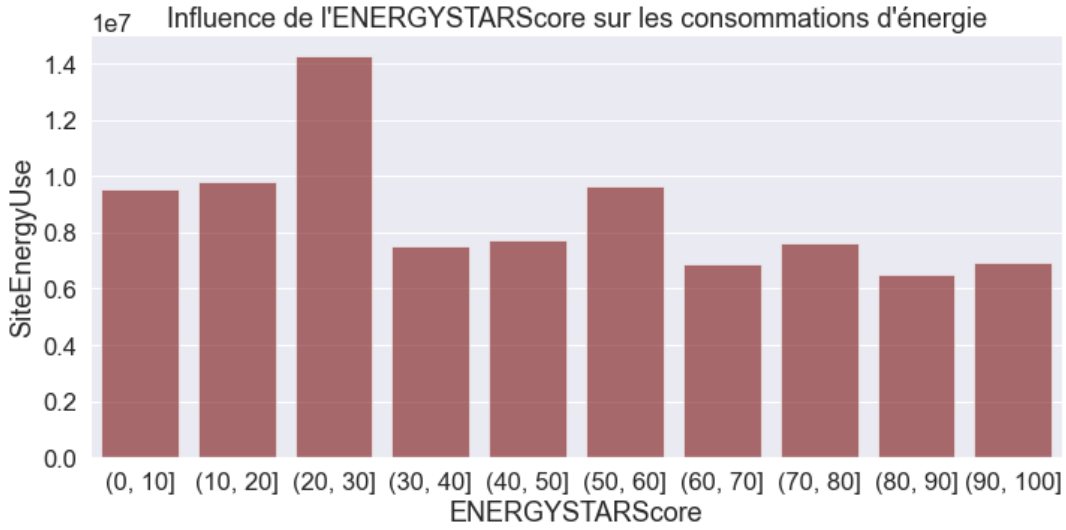
Ici, il faut supprimer les données de relevés d'énergie (électricité, gaz...) car c'est une fuite de données (data leakage).

```
# Calcul du mix énergétique des bâtiments
new_df["Steam%"] = (new_df["SteamUse(kBtu)"]/new_df["SiteEnergyUse"])*100
new_df["Electricity%"] = (new_df["Electricity(kBtu)"]/new_df["SiteEnergyUse"])*100
new_df["NaturalGas%"] = (new_df["NaturalGas(kBtu)"]/new_df["SiteEnergyUse"])*100
new_df["OtherEnergies%"] = 100 - (new_df["Steam%"]+new_df["Electricity%"]+new_df["NaturalGas%"])
```

```
new_df.drop(['SiteEUI(kBtu/sf)','SteamUse(kBtu)', 'NaturalGas(kBtu)','Electricity(kBtu)'], axis=1, inplace= True)
```

Influence de l'ENERGYSTARScore;



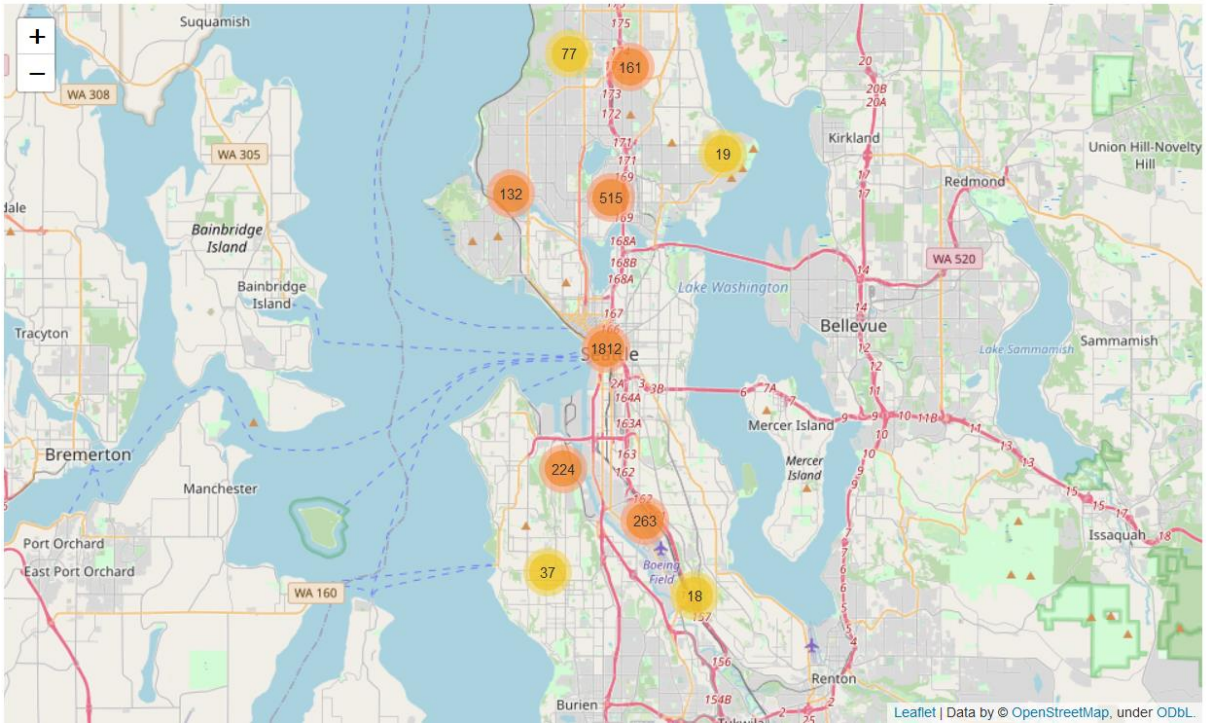


Les bâtiments avec score de 20-30 consument plus de énergie et produisent plus de CO2.

Note : certainement mois des consumations d'énergie, nous donne pas la meilleure énergie score.

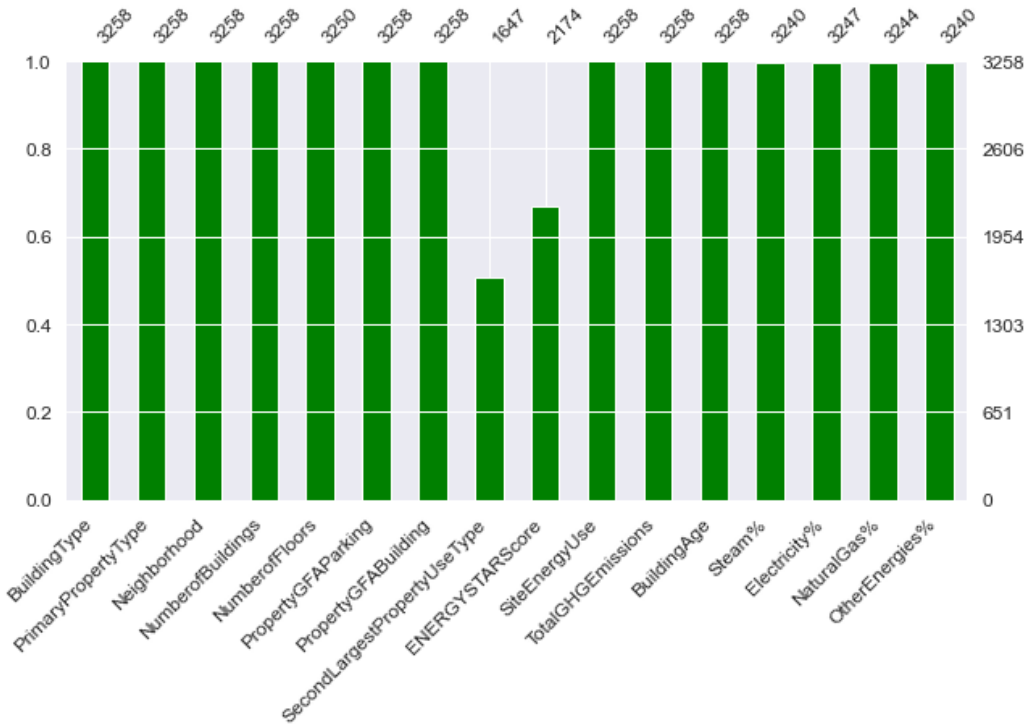
Géolocalisation des bâtiments non-résident

Avec les données 'Latitude', 'Longitude' on peut montrer le géolocalisation des bâtiments.



Maintenait on peut supprimer les deux colonnes 'Latitude', 'Longitude'.

Et dernier étape de parti 1 est regardant pour la dernière fois les **valeurs manquants**.



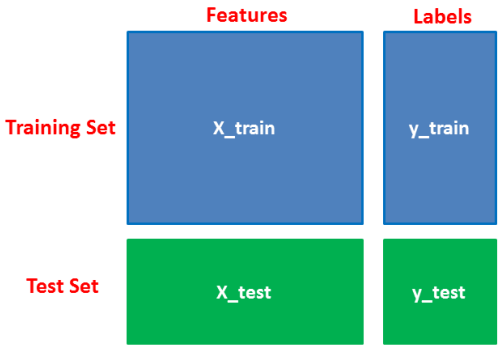
Et le dernier supprimassions est pour le colon ‘SecondLargestPropertyUseType’ avec plus de 50% valeurs manquants.

Partie 2

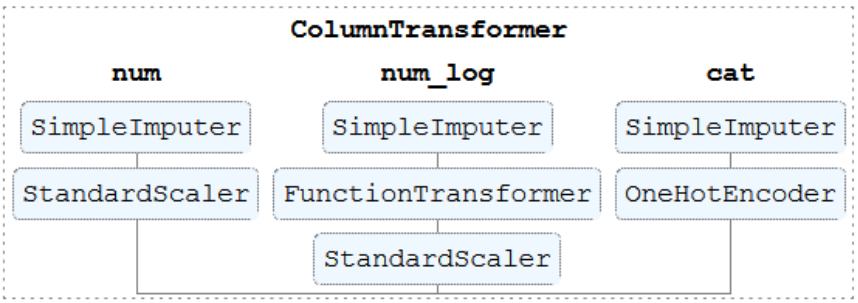
2.1 Preprocessing

Dans la deuxième partie de notre projets, il faut diviser les données en deux parties **"train" & "test"**. Nous entraînons l'algorithme dans les parties de train et les évaluons dans la partie test. Normalement, nous gardons 80% de nos données pour la partie Train et 20% pour la partie Test.

Aussi nous faisons nos données en deux parties de X et y. Dans la partie X, nous mettons les variables indépendantes et la partie y notre cible 'TotalGHGEmissions','SiteEnergyUse'.



Maintenant, nous commençons à faire le prétraitement des données. Nous devons d'abord faire l'imputation des données puis la normalisation pour les variables numériques et l'encodeur pour les variables catégorielles. On peut faire toutes ces étapes en utilisant ***pipeline*.**



Apres nous devons nous adapter notre processus de pipeline avec les parties Train.

Sélection et entrainement de modèles sur la variable SiteEnergyUse

En utilisant 4 modèles de régression :

- **Gradient Boosting**

  Construit un arbre à la fois,

  Combine les résultats en cours de route.

- **Random Forest Regressor**

  Construisent chaque arbre indépendamment,

  Combinent les résultats à la fin du processus (par moyenne).

- **Separate Vector Regression**

  Trouver la meilleure ligne d'ajustement. Dans SVR, la meilleure ligne d'ajustement est l'hyperplan qui a le nombre maximum de points.

- **Extra Trees Regressor**

  La forêt aléatoire utilise des répliques bootstrap, c'est-à-dire qu'elle ***sous-échantillonne*** les données d'entrée avec remplacement, tandis que les arbres supplémentaires utilisent l'ensemble de l'échantillon d'origine.

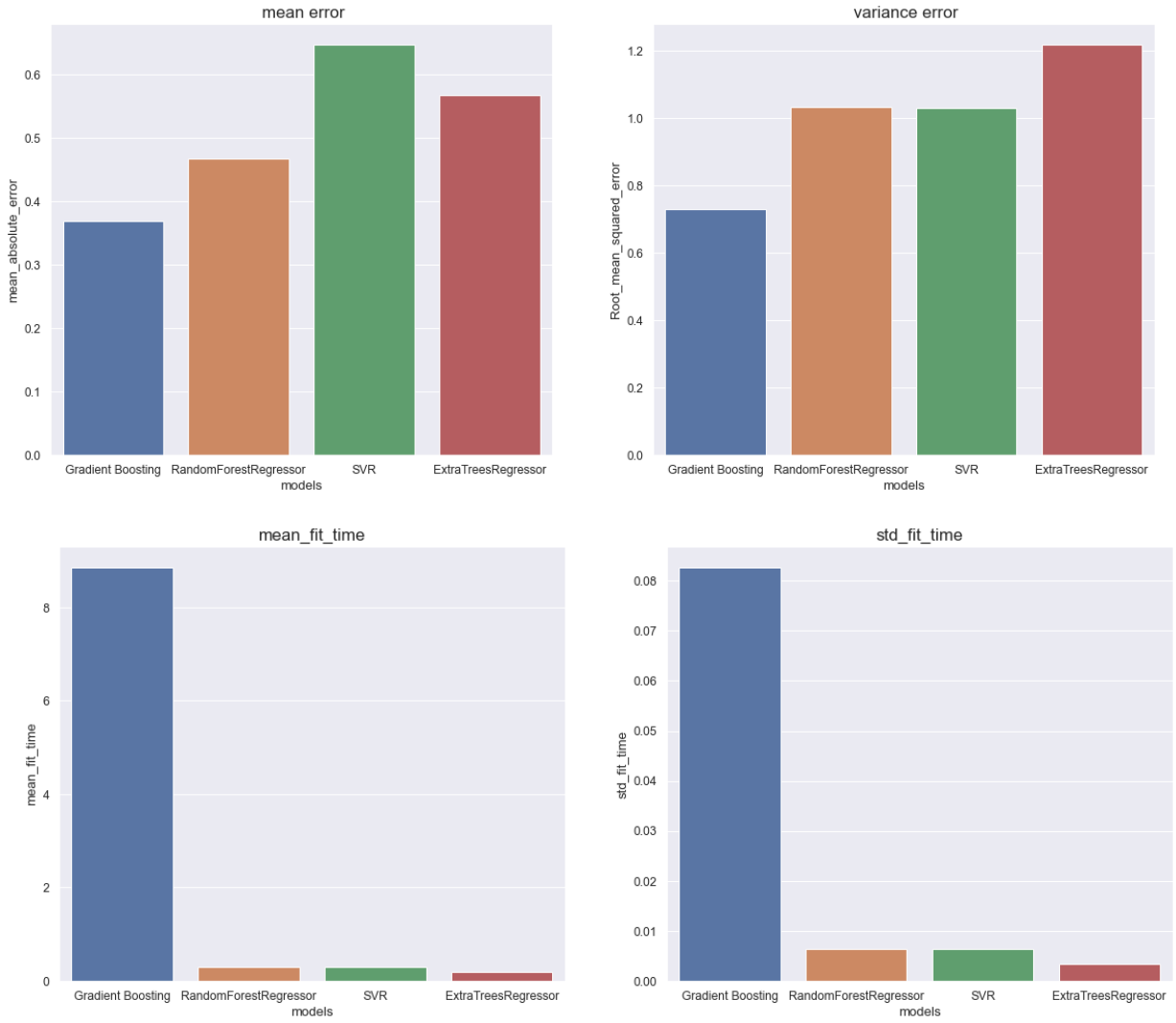
Grid Search

Pour avoir le bon résultait de modélisation il faut trouver les meilleurs hyper paramètres. Normalement, c'est très difficile de trouver les meilleurs hyper paramètres mais la méthode de **Grid Search** ou **Grid Randomized** peut nous aider beaucoup. Ici, nous utilisons la méthode Grid Search pour tous les algorithmes.

Apres trouver les meilleurs paramètres, nous adaptons notre modèle avec la partie de train de notre ensemble de données « .fit() ». Ensuite, nous retrouverons le score de notre modèle « .score() » sur la partie de test. Si le résultat est assez bon, signifie que l'algorithme a bien appris sur la partie train et nous pouvons utiliser la fonction predict() sur la partie X_test pour prédire la partie de y_test.

Pour évaluation de nos modèles, nous utilisons l'erreur absolue moyenne(MAE) et la racine carrée d'erreur absolue moyenne(RMSE). Le moins valeur de MAE et RMSE indique le meilleur modèle.

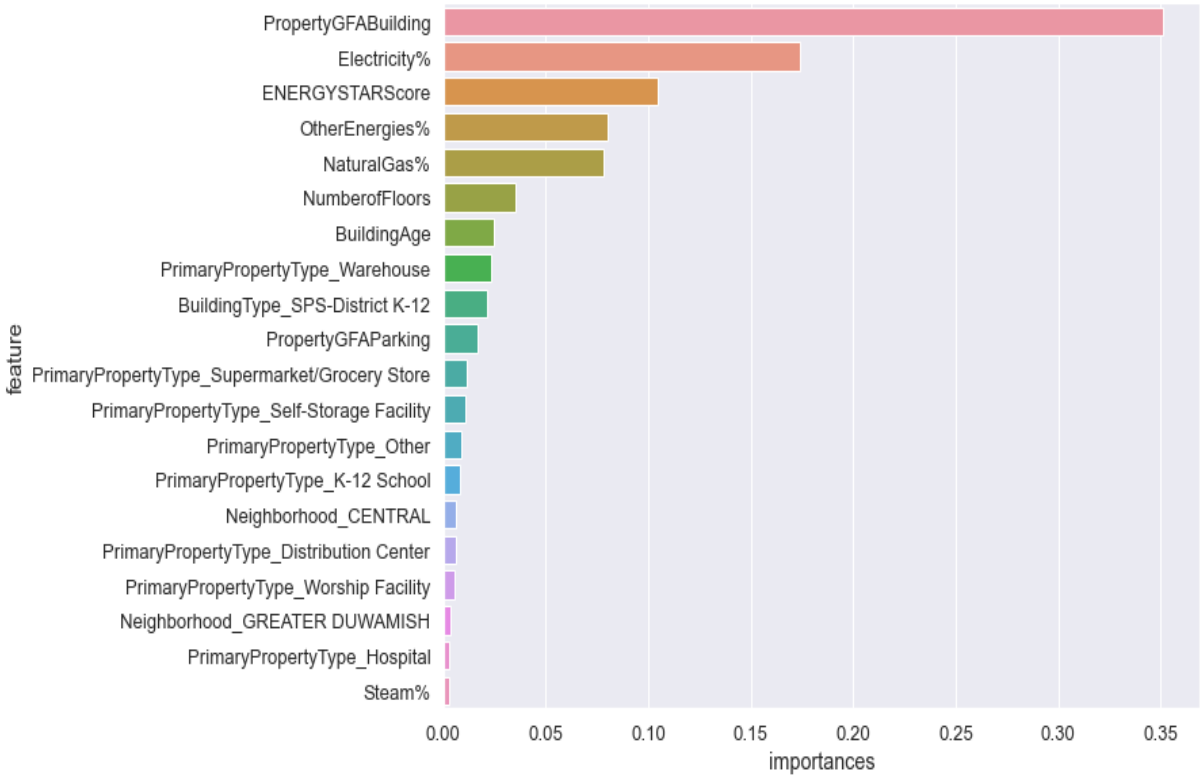
Sélection et entrainement de modèles sur la variable de la consumation d'energy



Ici notre meilleur modèle est **Gradient Boosting** avec moins d'erreur mais a passé plus de temps que les autres.

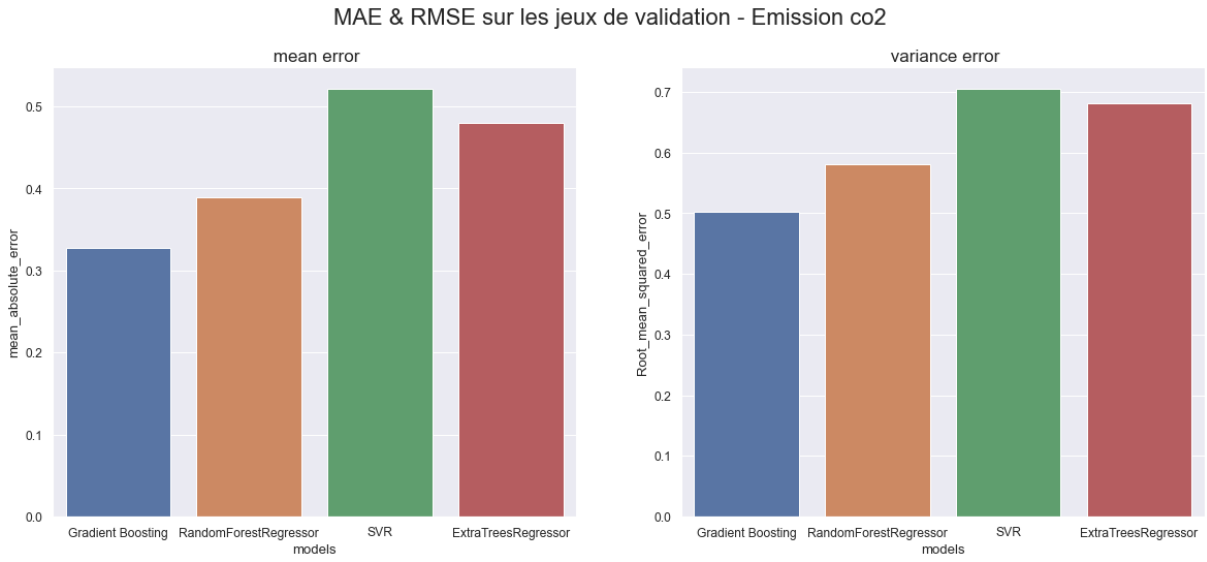
Feature importance

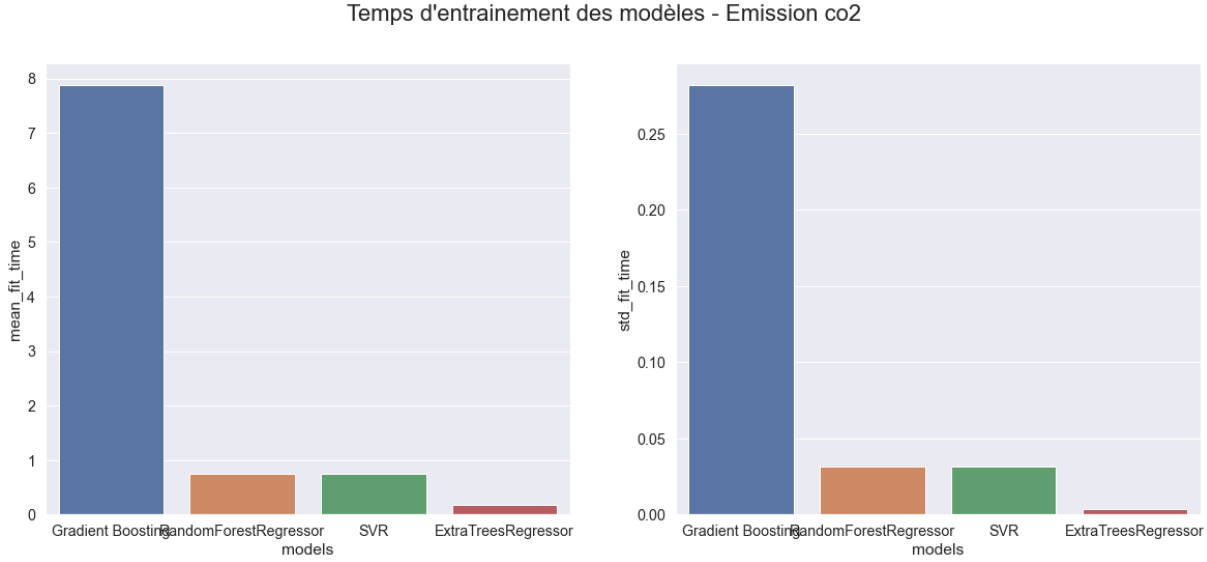
Pour voir quelles fonctionnalités sont importantes pour notre résultat, nous utilisons la fonction featur_importances_.



Ce graphe nous montre les 20 caractéristiques plus importantes sur la consumation d'énergie.

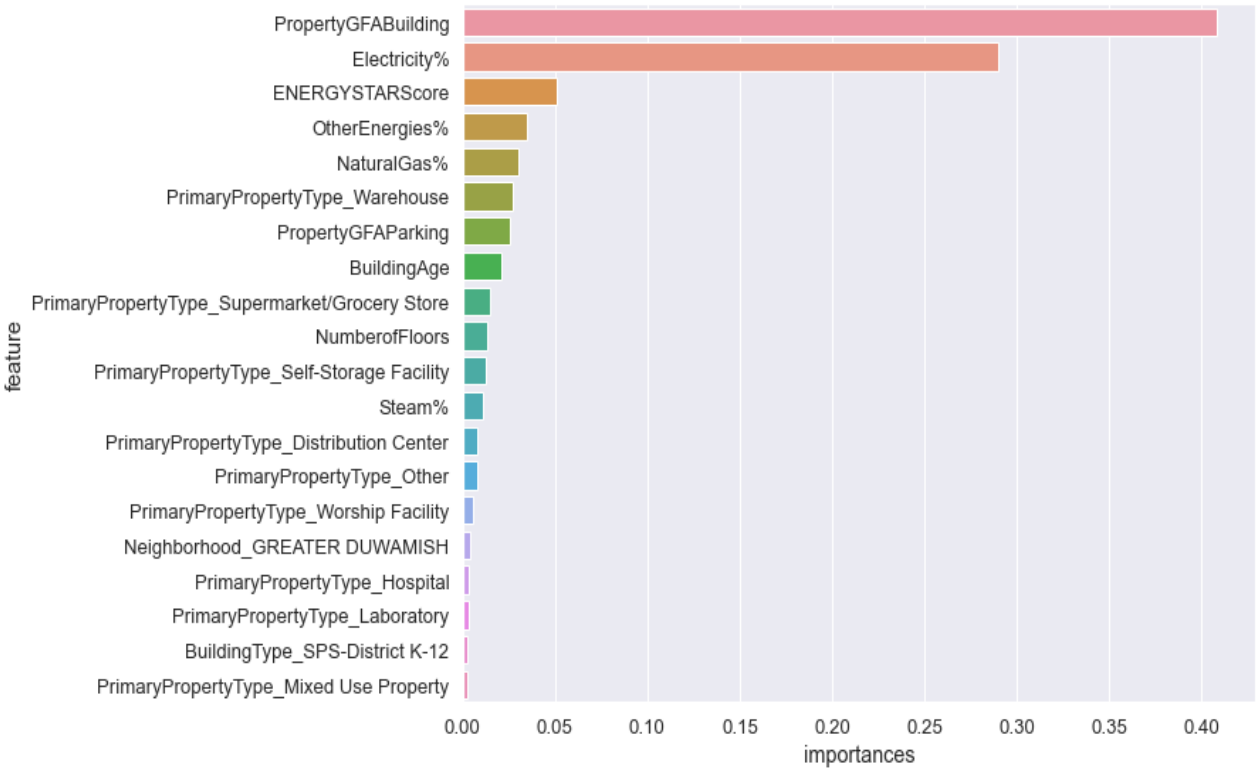
Sélection et entrainement de modèles sur la variable Emission CO2





Le modèle **Gradient Boosting** offre le meilleur résultat en erreur.

Feature importance sur la partie d'Emission de CO2.

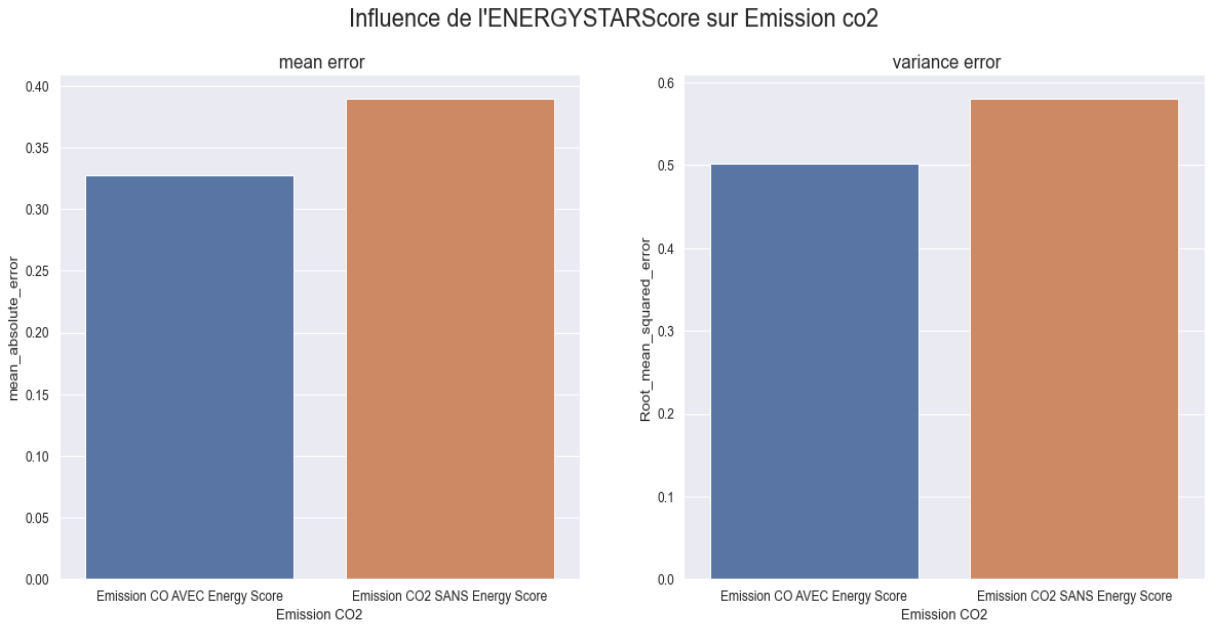


« Energy Star Score » est l'une des caractéristiques importantes pour l'Emission de CO2.

2.2 Influence de l'ENERGYSTARScore

L'Energy Star Score est un score entre 1 et 100 qui reflètent l'efficacité énergétique d'un bâtiment parmi les bâtiments similaires déjà certifiés. Un score de 50 signifie qu'il est dans la médiane alors qu'un score au-dessus de 75 indique qu'il s'agit d'un bâtiment à haute performance.

Comme « Energy Star Score » était une caractéristique plus importantes pour l'Emission de CO2, ici nous essayons de recalculer notre model « Gradient Boosting » sans avoir les donnes de EnergystarScore pour savoir est-ce que il a d'effet sur notre résultats ou pas !



Le graphe montre que l'ENERGY STAR Score a très peu d'influence sur le calcul de la prédiction du CO2.

Conclusion :

- Le **prétraitement des données** fait partie intégrante de l'apprentissage automatique, car la qualité des données et les informations utiles qui peuvent en être dérivées affectent directement la capacité d'apprentissage de notre modèle.
- **Grid Search** nous aide à trouver les meilleurs paramètres.
- Le modèle **Gradient boosting** a donné le meilleur résultait sur la **consumation d'énergie**.
- Le modèle **Gradient boosting** a donné le meilleur résultait pour **l'émission de CO2**.
- Et enfin Le label ENERGY STAR Score a **très peu d'influence** sur le calcul de la prédiction du CO2.