PROJET N°4 :

ANTICIPEZ LES BESOINS EN CONSOMMATION ÉLECTRIQUE DE BÂTIMENTS

Soutenance du P4: le 15/04/2022

Version notebook : **6.3.0**
Version Python : **3.8.8**
Version Pandas : **1.2.4**
Version Seaborn : **0.11.1**
Version Matplotlib: **3.3.4**

Plan

- Contexte et présentation des Data-Set
- Traitement et nettoyage du Data-Set
- Analyse exploratoire
- Modélisation
- Différents modèles
- Résultats SiteEnrgyUse et CO2
- Modèles retenu et l'impact EnergyStarScore
- Conclusion

Contexte et présentation des Data-Set

Contexte:

- Des données prélevées de la consommation d'énergie des bâtiments de la ville de Seattle pour les années 2015 et 2016
- Cout important de prélèvement des données de consommations ainsi que fastidieuse à collecter

Mission:

- Prédire la consommation totale d'énergie ainsi que les émissions de CO2 de bâtiments pour lesquels elles n'ont pas encore été mesurées.
- Evaluer l'intérêt de **l'ENERGY STAR** Score" pour la prédiction d'émissions
- Mettre en place un modèle prédictif Robuste

Contexte et présentation des Data-Set

Présentation du jeu données:

Data_2015 :

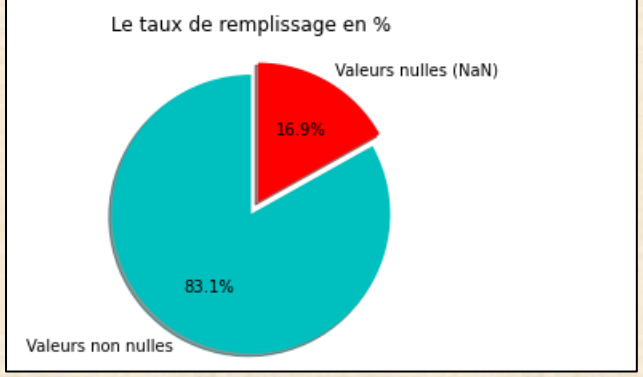
- il contient plus de 3340 prélèvements
- Taille : 3376 lignes, 46 colonnes, nombre de cases: 155296.
- Nombre de valeurs nulles: 26512
- Nombre de valeurs non nulles: 130468
- Le pourcentage des valeurs nulles: 16.9 %
- Le pourcentage des valeurs non nulles: 83.1 %

Data_2016 :

- il contient plus de 3376 prélèvements
- Taille : 3376 lignes, 46 colonnes, nombre de cases: 155296
- Nombre de valeurs nulles: 19952
- Nombre de valeurs non nulles: 135344
- Le pourcentage des valeurs nulles: 12.8 %
- Le pourcentage des valeurs non nulles: 87.2 %

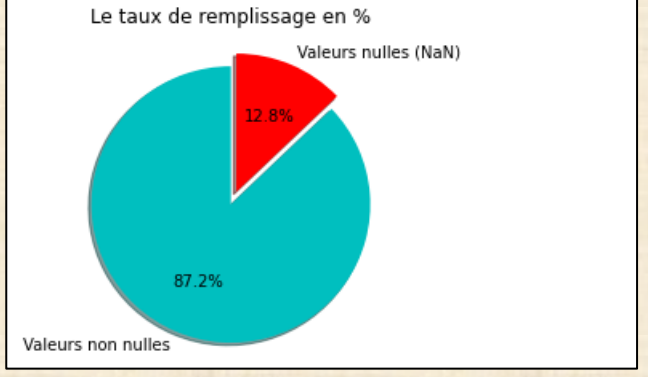
2015_building_energy_benchmarking

```
* Nombre de colonnes sans NaN -------: 15
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 32
* Nombre de ligne entièrement nulles : 0
* Nombre de ligne mixtes ---------- : 3340
* Nombre de ligne sans NaN ----------: 0
* Nombre de lignes ------------------------: 3340
* Nombre de colonnes ----------------------: 47
* Nombre de cases -------------------------: 156980
* Nombre de valeurs nulles ----------------: 26512
* Nombre de valeurs non nulles ------------: 130468
* le pourcentage des valeurs nulles ------: 16.9 %
* le pourcentage des valeurs non nulles --: 83.1 %
```



2016_building_energy_benchmarking

```
* Nombre de colonnes sans NaN -------: 20
* Nombre de colonnes NaN ------------: 1
* Nombre de colonnes mixtes----------: 25
* Nombre de ligne entièrement nulles : 0
* Nombre de ligne mixtes ---------- : 3376
* Nombre de ligne sans NaN ----------: 0
* Nombre de lignes ------------------------: 3376
* Nombre de colonnes ----------------------: 46
* Nombre de cases -------------------------: 155296
* Nombre de valeurs nulles ----------------: 19952
* Nombre de valeurs non nulles ------------: 135344
* le pourcentage des valeurs nulles ------: 12.8 %
* le pourcentage des valeurs non nulles --: 87.2 %
```



Traitement et nettoyage du Data-Set

Comparaison et merge de data_2015 et data_2016:

```
comp_data(data_2015,data_2016)
```

```
* Le nombre de colonnes absentes sur data_2016 :  10

 ['Location', 'OtherFuelUse(kBtu)', 'GHGEmissions(MetricTonsCO2e)', 'GHGEmissio
nsus Tracts', 'Seattle Police Department Micro Community Policing Plan Areas',
odes']

----------------------------------------------------------------------------------------------------------------------------
* Le nombre de colonnes absentes sur data_2015 : 9

 ['Address', 'City', 'State', 'ZipCode', 'Latitude', 'Longitude', 'Comments', 'TotalGHGEmissions', 'GHGEmissionsIntensity']
----------------------------------------------------------------------------------------------------------------------------
* Le nombre des colonnes communes : 37

 ['OSEBuildingID', 'DataYear', 'BuildingType', 'PrimaryPropertyType', 'PropertyName', 'TaxParcelIdentificationNumber', 'Coun
cilDistrictCode', 'Neighborhood', 'YearBuilt', 'NumberofBuildings', 'NumberofFloors', 'PropertyGFATotal', 'PropertyGFAParkin
g', 'PropertyGFABuilding(s)', 'ListOfAllPropertyUseTypes', 'LargestPropertyUseType', 'LargestPropertyUseTypeGFA', 'SecondLar
gestPropertyUseType', 'SecondLargestPropertyUseTypeGFA', 'ThirdLargestPropertyUseType', 'ThirdLargestPropertyUseTypeGFA', 'Y
earsENERGYSTARCertified', 'ENERGYSTARScore', 'SiteEUI(kBtu/sf)', 'SiteEUIWN(kBtu/sf)', 'SourceEUI(kBtu/sf)', 'SourceEUIWN(kB
tu/sf)', 'SiteEnergyUse(kBtu)', 'SiteEnergyUseWN(kBtu)', 'SteamUse(kBtu)', 'Electricity(kWh)', 'Electricity(kBtu)', 'Natural
Gas(therms)', 'NaturalGas(kBtu)', 'DefaultData', 'ComplianceStatus', 'Outlier']
```

Première itération avec la fonction comp_data

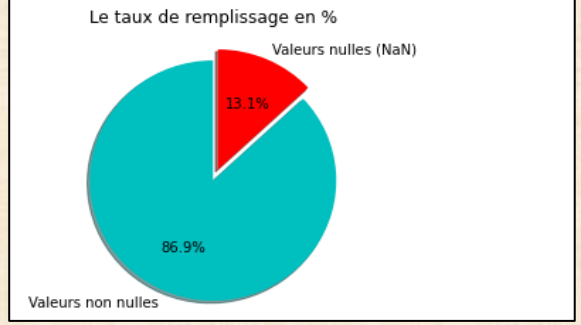
```
comp_data(data_2015_new, data_2016)
```

```
* Le nombre de colonnes absentes sur data_2016 :  0

 []

----------------------------------------------------------------------------------------------------------------------------
* Le nombre de colonnes absentes sur data_2015 : 0

 []
----------------------------------------------------------------------------------------------------------------------------
* Le nombre des colonnes communes : 46

 ['OSEBuildingID', 'DataYear', 'BuildingType', 'PrimaryPropertyType', 'PropertyName', 'Address', 'City', 'State', 'ZipCode',
'TaxParcelIdentificationNumber', 'CouncilDistrictCode', 'Neighborhood', 'Latitude', 'Longitude', 'YearBuilt', 'NumberofBuild
ings', 'NumberofFloors', 'PropertyGFATotal', 'PropertyGFAParking', 'PropertyGFABuilding(s)', 'ListOfAllPropertyUseTypes', 'L
argestPropertyUseType', 'LargestPropertyUseTypeGFA', 'SecondLargestPropertyUseType', 'SecondLargestPropertyUseTypeGFA', 'Thi
rdLargestPropertyUseType', 'ThirdLargestPropertyUseTypeGFA', 'YearsENERGYSTARCertified', 'ENERGYSTARScore', 'SiteEUI(kBtu/s
f)', 'SiteEUIWN(kBtu/sf)', 'SourceEUI(kBtu/sf)', 'SourceEUIWN(kBtu/sf)', 'SiteEnergyUse(kBtu)', 'SiteEnergyUseWN(kBtu)', 'St
eamUse(kBtu)', 'Electricity(kWh)', 'Electricity(kBtu)', 'NaturalGas(therms)', 'NaturalGas(kBtu)', 'DefaultData', 'Comments',
'ComplianceStatus', 'Outlier', 'TotalGHGEmissions', 'GHGEmissionsIntensity']
```

Maintenant on a 46 colonnes communes entre les deux Data-Set

```
calc_inf(data_uni,True)
```

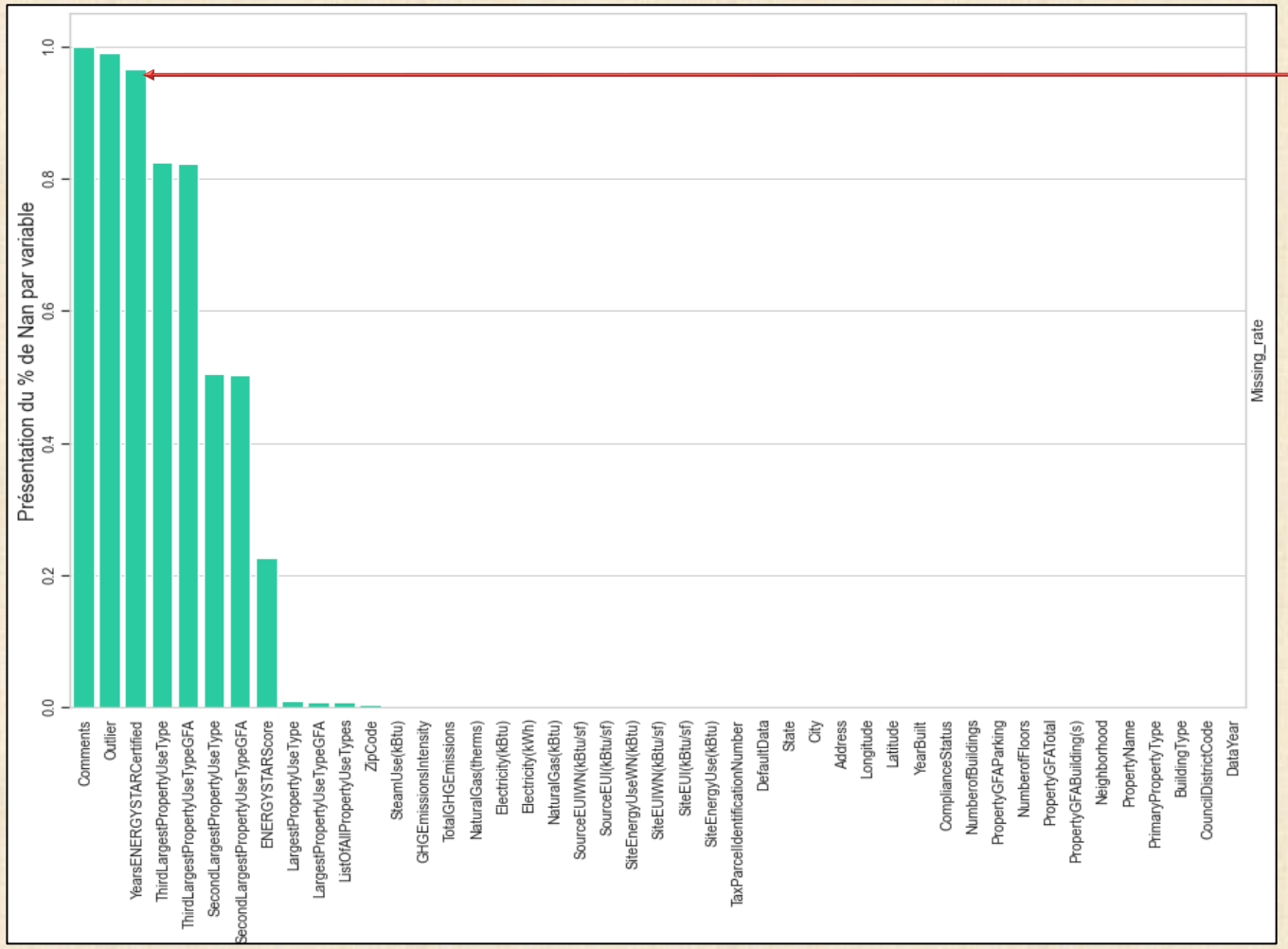
```
* Nombre de colonnes sans NaN -------: 19
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 26
* Nombre de ligne entièrement nulles : 0
* Nombre de ligne mixtes ------------ : 3432
* Nombre de ligne sans NaN ----------: 0
* Nombre de lignes ------------------------: 3432
* Nombre de colonnes ----------------------: 45
* Nombre de cases -------------------------: 154440
* Nombre de valeurs nulles ----------------: 20189
* Nombre de valeurs non nulles ------------: 134251
* le pourcentage des valeurs nulles ------: 13.1 %
* le pourcentage des valeurs non nulles --: 86.9 %
```



Les deux Data-Set 2015, 2016 sont maintenant concaténé et mergé
Ils sont groupés par OSEBuildingID
Et une agrégation par la moyenne pour l'ensemble des colonnes numérique.

Traitement et nettoyage du Data-Set

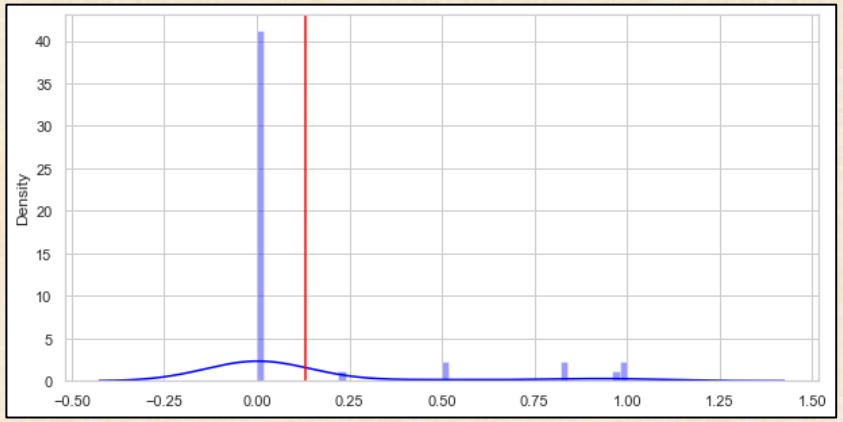
B.1. Présentation des valeurs NaN de toutes les colonnes:



Échantillon des colonnes presque entièrement nuls

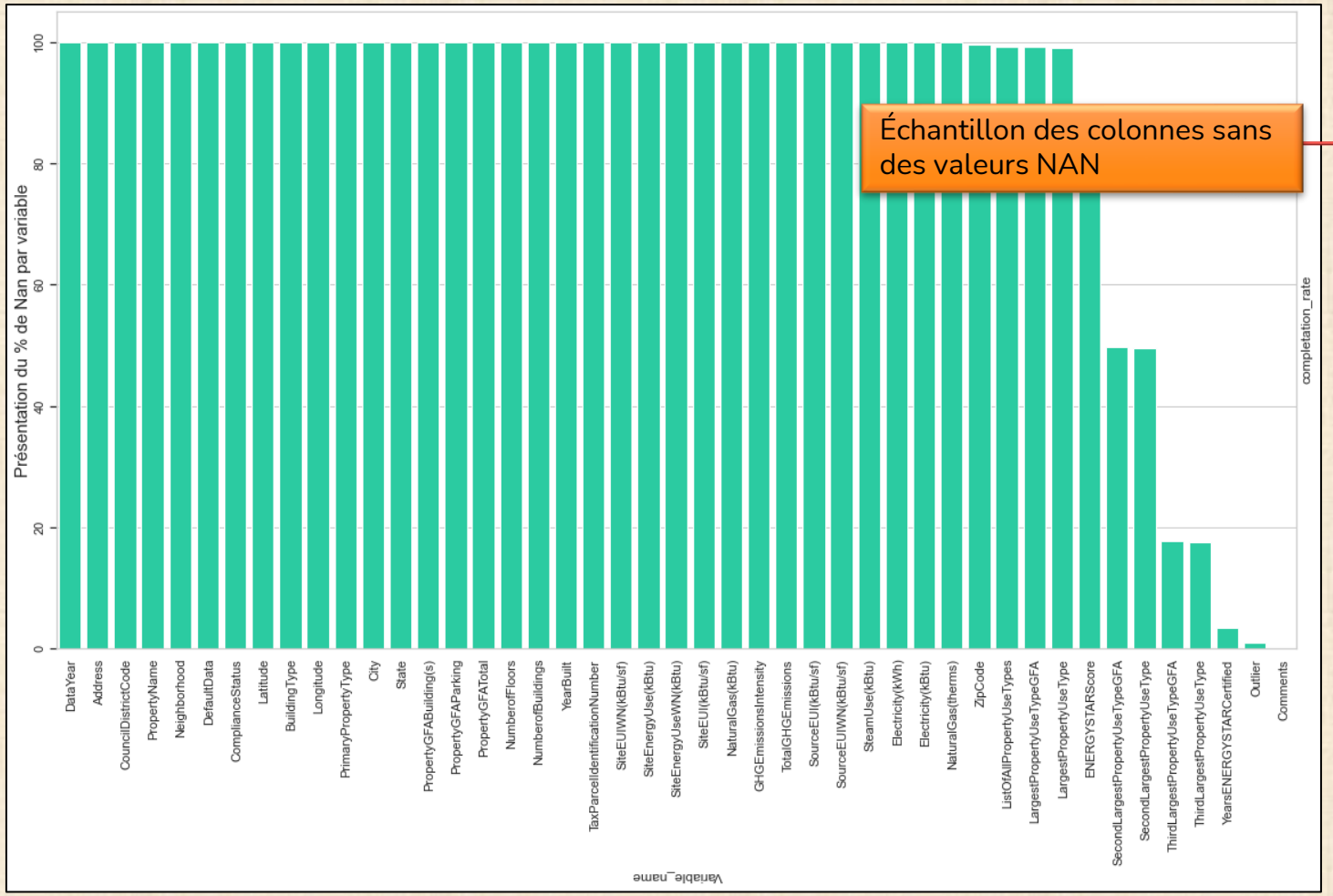
	Variable_name	Missing_values	Missing_rate
36	Comments	3431	0.999709
38	Outlier	3397	0.989802
34	YearsENERGY...	3312	0.965035
33	ThirdLarges...	2830	0.824592
10	ThirdLarges...	2825	0.823135
32	SecondLarge...	1734	0.505245
9	SecondLarge...	1728	0.503497

La distribution des valeurs NaN autour de la moyenne



Traitement et nettoyage du Data-Set

B.1. Présentation des valeurs non nulles de toutes les colonnes:



Échantillon des colonnes sans des valeurs NAN

	Variable_name	completation_values	completation_rate
0	DataYear	3432	100.000000
41	Address	3432	100.000000
1	CouncilDist...	3432	100.000000
27	PropertyName	3432	100.000000
29	Neighborhood	3432	100.000000
35	DefaultData	3432	100.000000
37	ComplianceS...	3432	100.000000
39	Latitude	3432	100.000000
25	BuildingType	3432	100.000000
40	Longitude	3432	100.000000
26	PrimaryProp...	3432	100.000000
42	City	3432	100.000000
43	State	3432	100.000000
7	PropertyGFA...	3432	100.000000
6	PropertyGFA...	3432	100.000000
5	PropertyGFA...	3432	100.000000
4	NumberofFloors	3432	100.000000
3	NumberofBui...	3432	100.000000
2	YearBuilt	3432	100.000000
28	TaxParcelId...	3431	99.970862

Traitement et nettoyage du Data-Set

Filtration des colonnes au dessus de 80% de NaN:

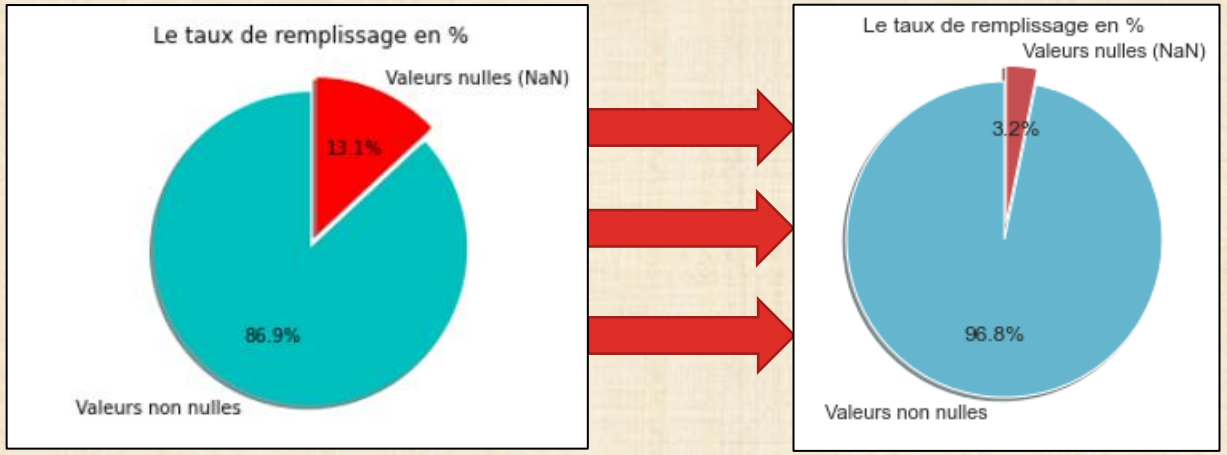
Le Via la fonction « flt_nan » on fixe le seuil à 80% pour supprimer les colonnes supérieur ou égale à 80% de NaN

```
1 data_uni = flt_nan(data_uni, 80) # Suppréssion des colonnes qui contiennent plus de 80% de NaN
2 data_uni.shape
```

```
(3432, 40)
```

Le Data-Set est passé de 45 colonnes à 40.

En therme de pourcentage, le Data-Set maintenant est à 96.8% de valeurs non nulles



Traitement et nettoyage du Data-Set

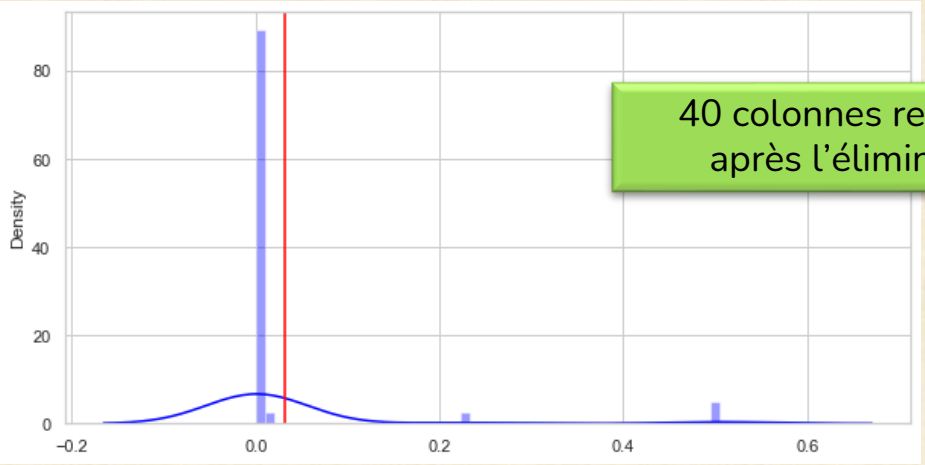
Seattle

Filtration des colonnes NaN:

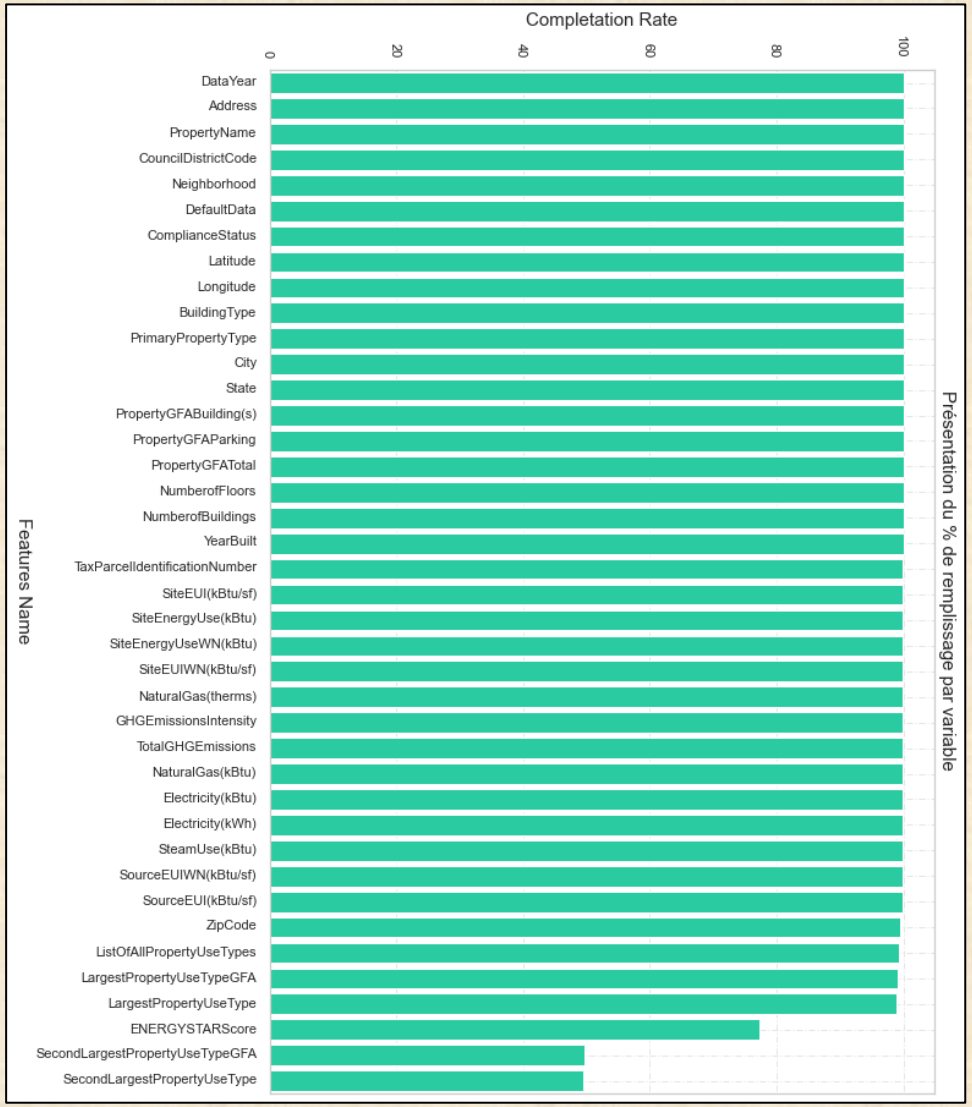
Sur le « Barplot » à droit, les trois variables qui restent ont un pourcentage plus de 80% de NaN, qui représentaient presque 20% de NaN dans le Data-Set non filtré

19 Colonnes sans NaN

La distribution des valeurs NaN autour de la moyenne



40 colonnes restantes après l'élimination



Traitement et nettoyage du Data-Set

Traitement des doublons et valeurs aberrantes:

Pour traiter les valeurs aberrantes il existe plusieurs méthodes comme z-score, Interquartiles, percentile.., dans un premier temps on va traiter les valeurs négatives, les doublons :

DataYear	2015
BuildingType	Campus
PrimaryPropertyType	Distributio...
Address	(ID23682) 3...
ZipCode	98006.0
CouncilDistrictCode	1
Neighborhood	BALLARD
Latitude	47.49917
Longitude	-122.41425
YearBuilt	1900
NumberofBuildings	-9223372036...
NumberofFloors	0
PropertyGFATotal	11285.0
PropertyGFAParking	-1.5
PropertyGFABuilding(s)	-8451.0
ENERGYSTARScore	1.0
SiteEUI(kBtu/sf)	0.0
SiteEUIWN(kBtu/sf)	0.0
SourceEUI(kBtu/sf)	0.0
SourceEUIWN(kBtu/sf)	-2.05
SiteEnergyUse(kBtu)	0.0
SteamUse(kBtu)	0.0
Electricity(kBtu)	7.0
NaturalGas(kBtu)	0.0
ComplianceStatus	Compliant
TotalGHGEmissions	0.09

```
data_uni[data_uni.duplicated() == True].shape[0]
```

```
0
```

- Maintenant on a que des valeurs positives, sauf la variable Longitude qui peut avoir des valeur entre -180° et 180°
- Concernant les lignes dupliqué on n'en a pas

DataYear	2015
BuildingType	Campus
PrimaryPropertyType	Distributio...
Address	(ID23682) 3...
ZipCode	98006.0
CouncilDistrictCode	1
Neighborhood	BALLARD
Latitude	47.49917
Longitude	-122.41425
YearBuilt	1900
NumberofBuildings	0
NumberofFloors	0
PropertyGFATotal	11285.0
PropertyGFAParking	0.0
PropertyGFABuilding(s)	5352.5
ENERGYSTARScore	1.0
SiteEUI(kBtu/sf)	0.0
SiteEUIWN(kBtu/sf)	0.0
SourceEUI(kBtu/sf)	0.0
SourceEUIWN(kBtu/sf)	0.0
SiteEnergyUse(kBtu)	0.0
SteamUse(kBtu)	0.0
Electricity(kBtu)	7.0
NaturalGas(kBtu)	0.0
ComplianceStatus	Compliant
TotalGHGEmissions	0.09

```
for col in ['SourceEUIWN(kBtu/sf)', 'PropertyGFABuilding(s)', 'PropertyGFAParking', 'NumberofBuildings'] :
    data_uni = data_uni[data_uni[col] >= 0]
```

Vérification des valeurs nulles dans:

'NumberofBuildings' et 'NumberofFloors'

```
(data_flt["NumberofBuildings"] ==0).value_counts()
```

```
False    2358
True       51
Name: NumberofBuildings, dtype: int64
```

```
(data_flt["NumberofFloors"] ==0).value_counts()
```

```
False    2404
True        5
Name: NumberofFloors, dtype: int64
```

```
(data_flt['PropertyGFATotal'] == 0).value_counts()
```

```
False    2409
Name: PropertyGFATotal, dtype: int64
```

```
for col in  ['NumberofBuildings', "NumberofFloors"]:
    data_flt[col] = data_flt[col].apply(lambda x : 1 if x == 0 else x)

```

```
(data_flt["NumberofBuildings"] ==0).value_counts()
```

```
False    2409
Name: NumberofBuildings, dtype: int64
```

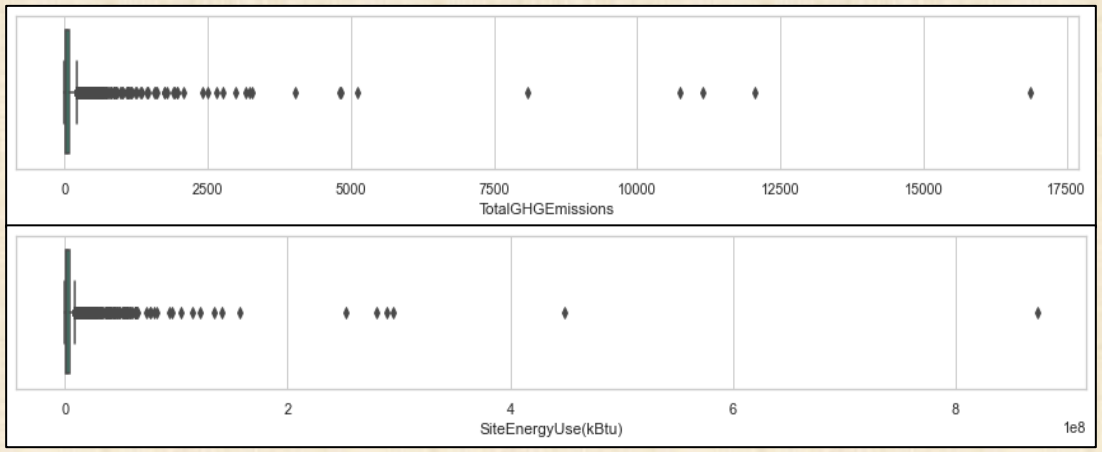
```
(data_flt["NumberofFloors"] ==0).value_counts()
```

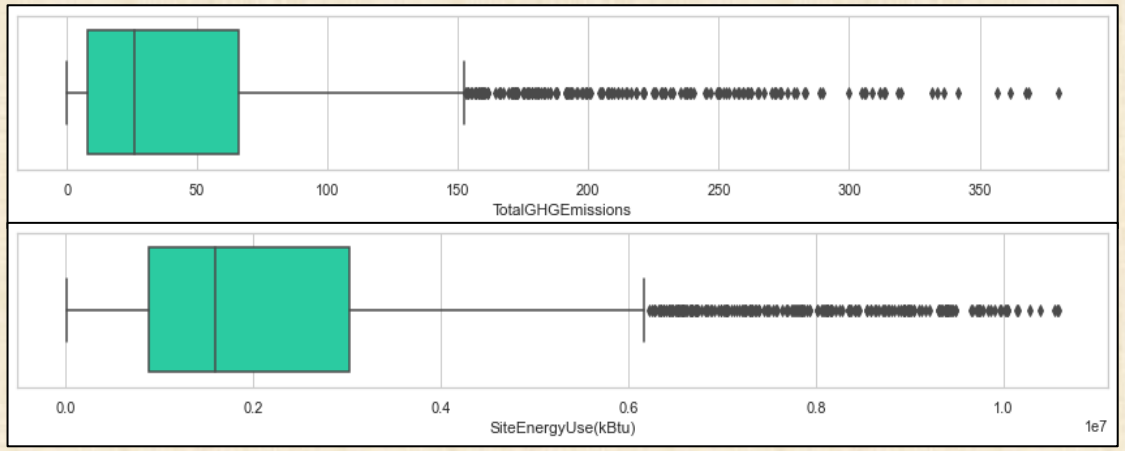
```
False    2409
Name: NumberofFloors, dtype: int64
```

Traitement et nettoyage du Data-Set

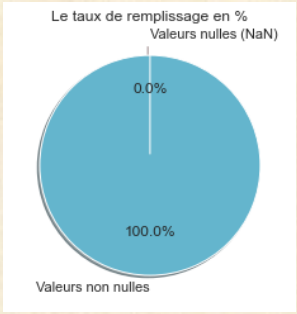
Traitement des valeurs aberrantes: la méthode « Percentile »

Le Boxplot en dessous représente les deux targets les émissions CO2 et la consommation d'énergie totale avant et après l'application de la méthode Percentile, ou on a enlevé les 5% près de 100%, puis supprimer les valeurs nulles de notre Data-Set






```
for col in data_uni[outliers].columns:
    data_uni.loc[data_uni[col] > data_uni[col].quantile(0.95)] = np.nan
data_uni = data_uni.dropna()
```



Traitement et nettoyage du Data-Set

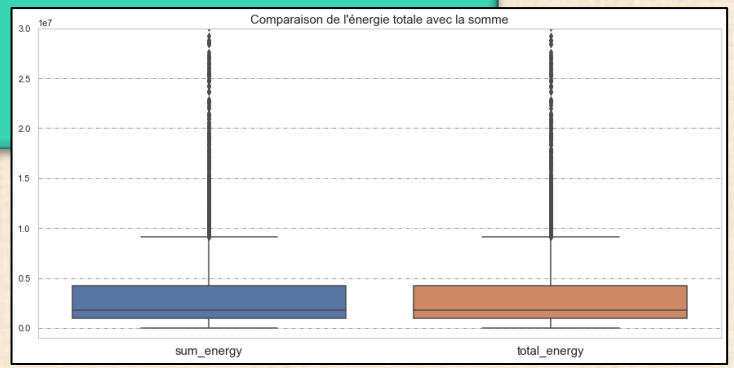
Sélection des variables pour la modélisation:

Après le premier filtrage, maintenant on doit garder que les variables qui nous intéressent pour notre prédiction, en dessous la liste des variables potentielles

```
lst_to_dlt = ['DataYear','ZipCode',
              'CouncilDistrictCode',
              'YearBuilt',
              'PropertyGFAParking',
              'PropertyGFABuilding(s)',
              'SiteEUI(kBtu/sf)','SiteEUIWN(kBtu/sf)',
              'SourceEUI(kBtu/sf)','SourceEUIWN(kBtu/sf)','SteamUse(kBtu)',
              'Electricity(kBtu)','NaturalGas(kBtu)','Address',
              'ComplianceStatus','TaxParcelIdentificationNumber']
```

Des variables 'Object' qui concernent le type de bâtiments, le nom du produit, type de propriété principal, et le quartier ou se trouve le bâtiment

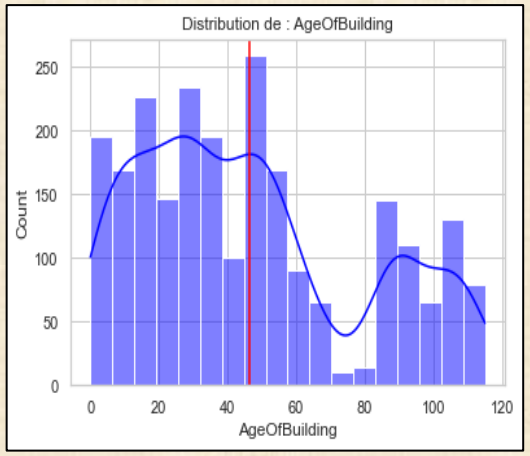
Des variables numériques comme l'âge de bâtiments que j'ai calculé en fonction de la variable YearOfBuilt, Latitude, Longitude, le nombre d'étage et le numéro du bâtiment
La surface GFA totale après la vérification de la somme de la surface de parking et habitable
ENERGYSTARSScore
Les deux targets:
SiteEnergyUse(kBtu)
TotalGHGEmissions

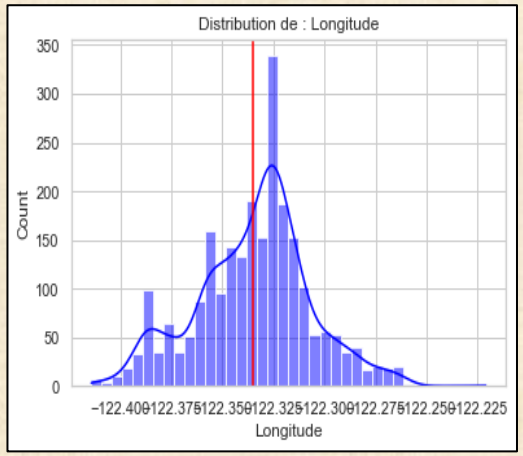


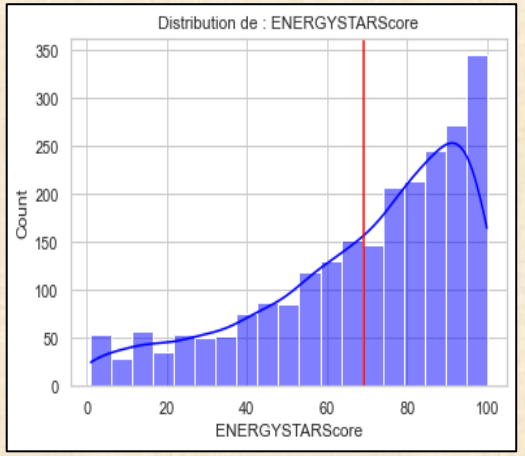
#	Column		
0	DataYear	2409 non-null	float64
1	BuildingType	2409 non-null	object
2	PrimaryPropertyType	2409 non-null	object
3	Address	2409 non-null	object
4	ZipCode	2409 non-null	float64
5	TaxParcelIdentificationNumber	2409 non-null	object
6	CouncilDistrictCode	2409 non-null	float64
7	AgeOfBuilding	2409 non-null	float64
8	Neighborhood	2409 non-null	object
9	Latitude	2409 non-null	float64
10	Longitude	2409 non-null	float64
11	YearBuilt	2409 non-null	float64
12	NumberofBuildings	2409 non-null	float64
13	NumberofFloors	2409 non-null	float64
14	PropertyGFATotal	2409 non-null	float64
15	PropertyGFAParking	2409 non-null	float64
16	PropertyGFABuilding(s)	2409 non-null	float64
17	ENERGYSTARScore	2409 non-null	float64
18	SiteEUI(kBtu/sf)	2409 non-null	float64
19	SiteEUIWN(kBtu/sf)	2409 non-null	float64
20	SourceEUI(kBtu/sf)	2409 non-null	float64
21	SourceEUIWN(kBtu/sf)	2409 non-null	float64
22	SiteEnergyUse(kBtu)	2409 non-null	float64
23	SteamUse(kBtu)	2409 non-null	float64
24	Electricity(kBtu)	2409 non-null	float64
25	NaturalGas(kBtu)	2409 non-null	float64
26	ComplianceStatus	2409 non-null	object
27	TotalGHGEmissions	2409 non-null	float64

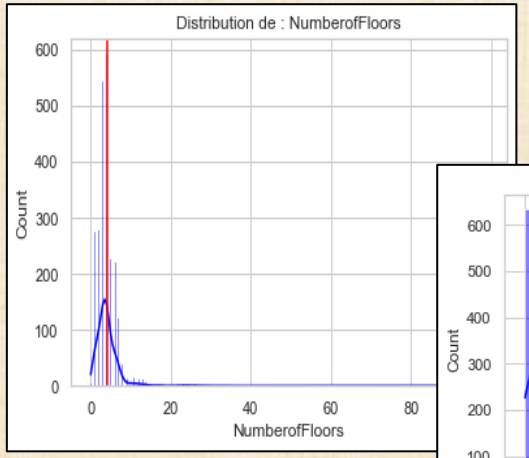
Analyse exploratoire

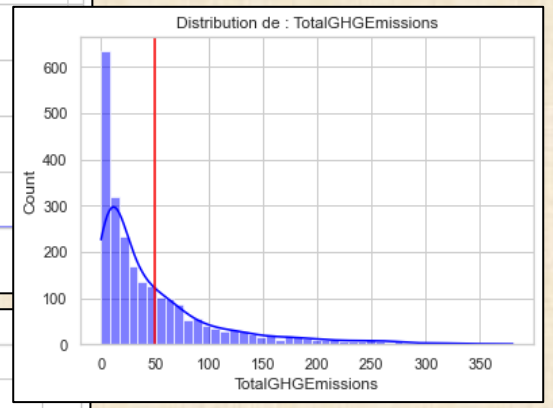
Analyse univariée: Distribution des données quantitatives par rapport à la moyenne

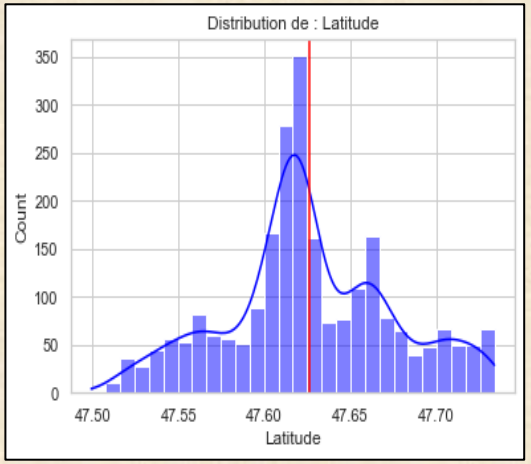


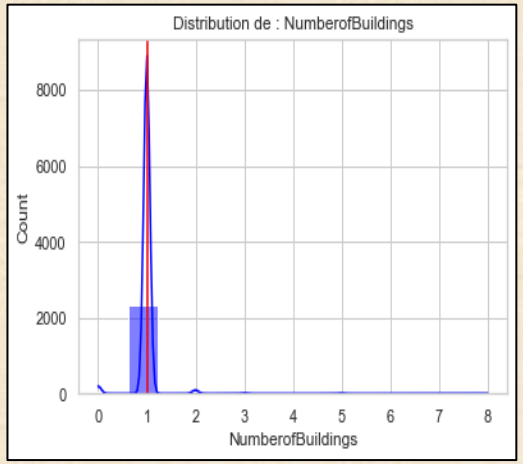


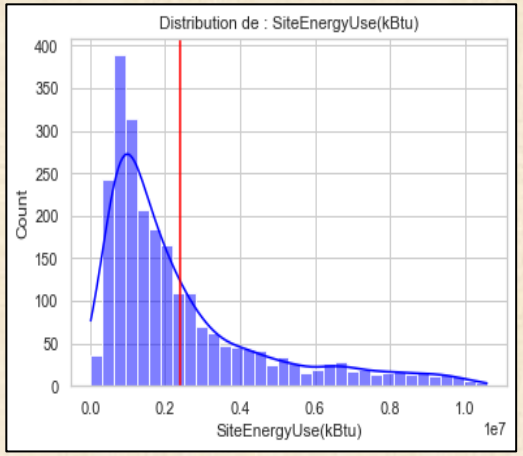


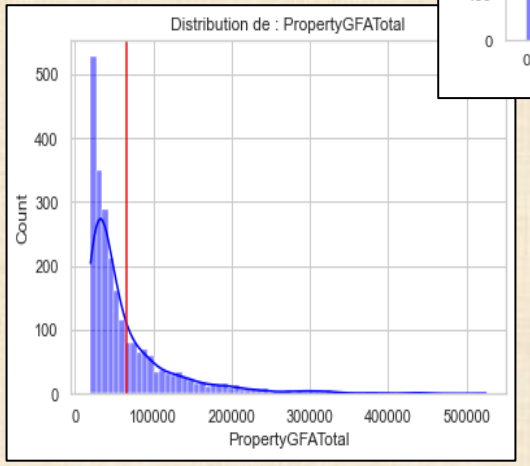










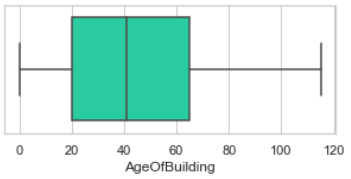


Analyse exploratoire

Analyse univariée: distribution quantitatives et qualitatives

Indicateurs de distribution pour AgeOfBuilding

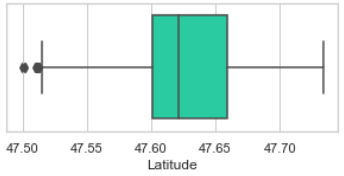
count	2409.00000
mean	46.43379
std	32.55982
min	0.00000
25%	20.00000
50%	41.00000
75%	65.00000
max	115.00000



Name: AgeOfBuilding, dtype: float64

Indicateurs de distribution pour Latitude

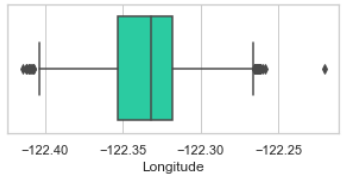
count	2409.000000
mean	47.626025
std	0.049012
min	47.499170
25%	47.601100
50%	47.620940
75%	47.659260
max	47.733870



Name: Latitude, dtype: float64

Indicateurs de distribution pour Longitude

count	2409.000000
mean	-122.334783
std	0.027978
min	-122.414250
25%	-122.353220
50%	-122.331970
75%	-122.318380
max	-122.220966



Name: Longitude, dtype: float64

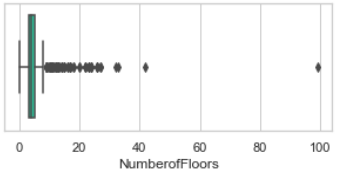
Indicateurs de distribution pour NumberofBuildings

count	2409.000000
mean	1.003321
std	0.303614
min	0.000000
25%	1.000000
50%	1.000000
75%	1.000000
max	8.000000

Name: NumberofBuildings, dtype: float64

Indicateurs de distribution pour NumberofFloors

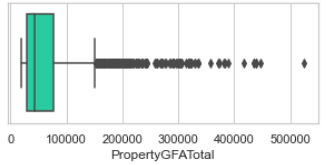
count	2409.00000
mean	4.18846
std	3.62476
min	0.00000
25%	3.00000
50%	4.00000
75%	5.00000
max	99.00000



Name: NumberofFloors, dtype: float64

Indicateurs de distribution pour PropertyGFATotal

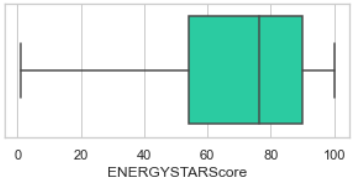
count	2409.000000
mean	65700.444168
std	60432.256717
min	20000.000000
25%	28568.000000
50%	43183.000000
75%	77800.000000
max	524720.000000



Name: PropertyGFATotal, dtype: float64

Indicateurs de distribution pour ENERGYSTARScore

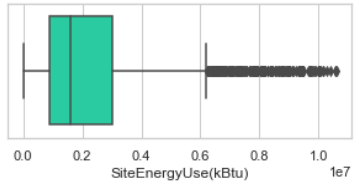
count	2409.000000
mean	69.108136
std	25.677070
min	1.000000
25%	54.000000
50%	76.500000
75%	90.000000
max	100.000000



Name: ENERGYSTARScore, dtype: float64

Indicateurs de distribution pour SiteEnergyUse(kBtu)

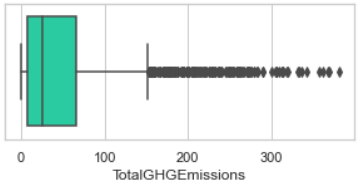
count	2.409000e+03
mean	2.407919e+06
std	2.198796e+06
min	1.340900e+04
25%	8.840302e+05
50%	1.592651e+06
75%	3.020066e+06
max	1.058778e+07



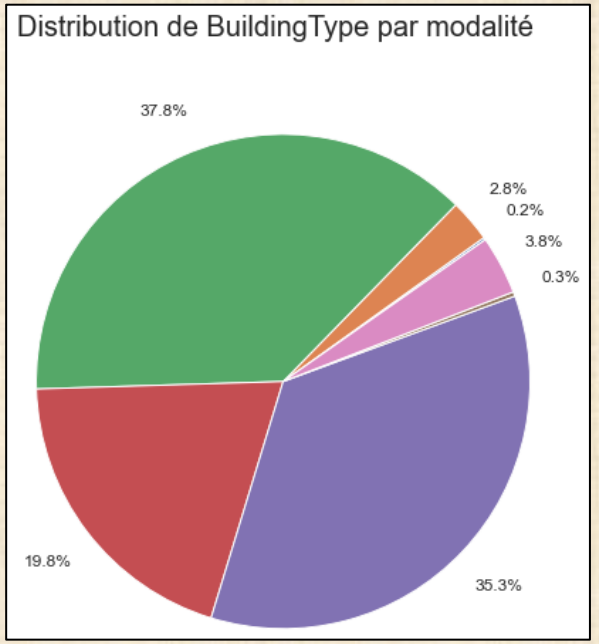
Name: SiteEnergyUse(kBtu), dtype: float64

Indicateurs de distribution pour TotalGHGEmissions

count	2409.000000
mean	50.063547
std	61.378108
min	0.090000
25%	8.180000
50%	26.185000
75%	65.850000
max	380.270000

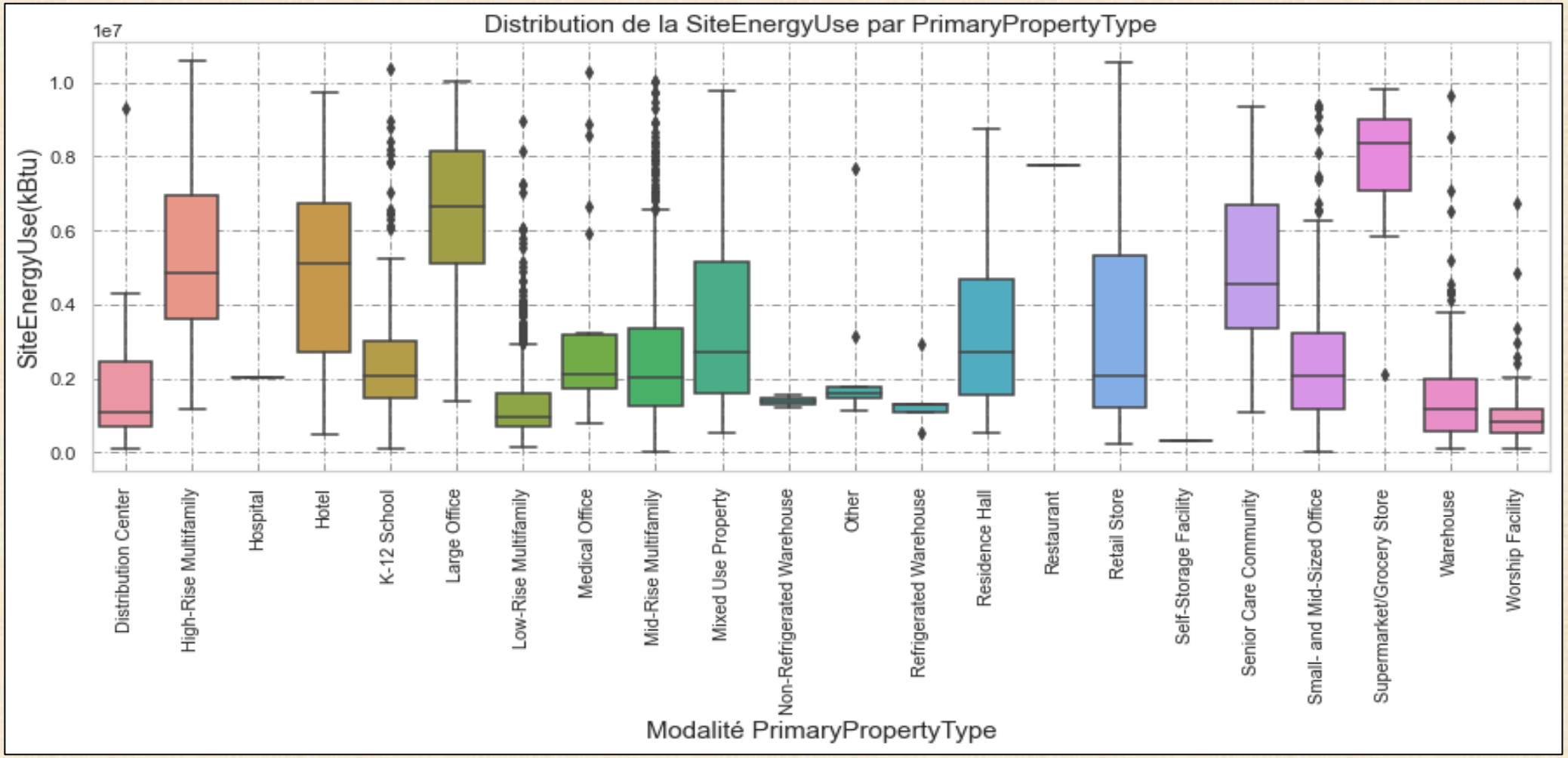


Name: TotalGHGEmissions, dtype: float64



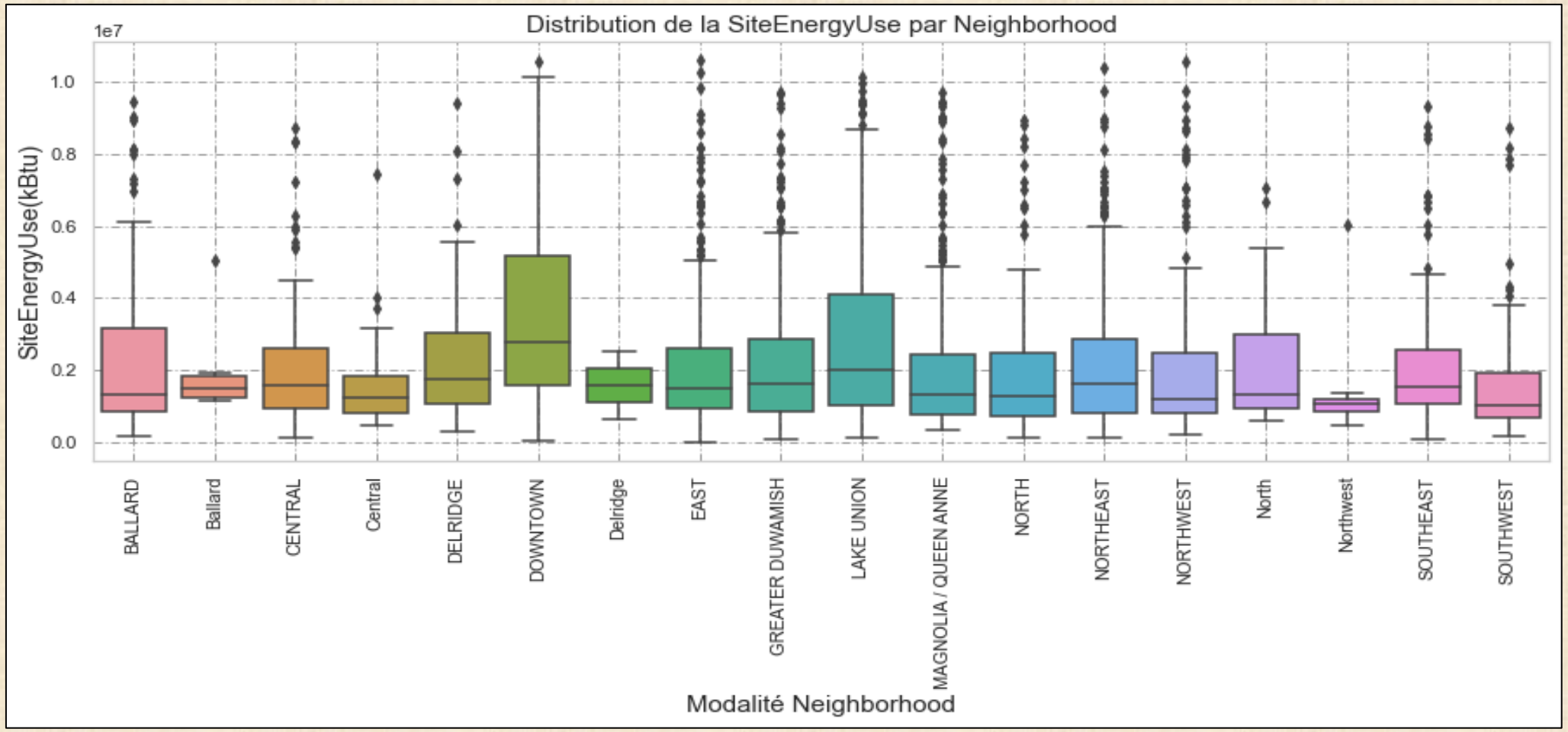
Analyse exploratoire

Analyse bivariée: la distribution de SiteEnergyUse par PrimaryPropertyType:



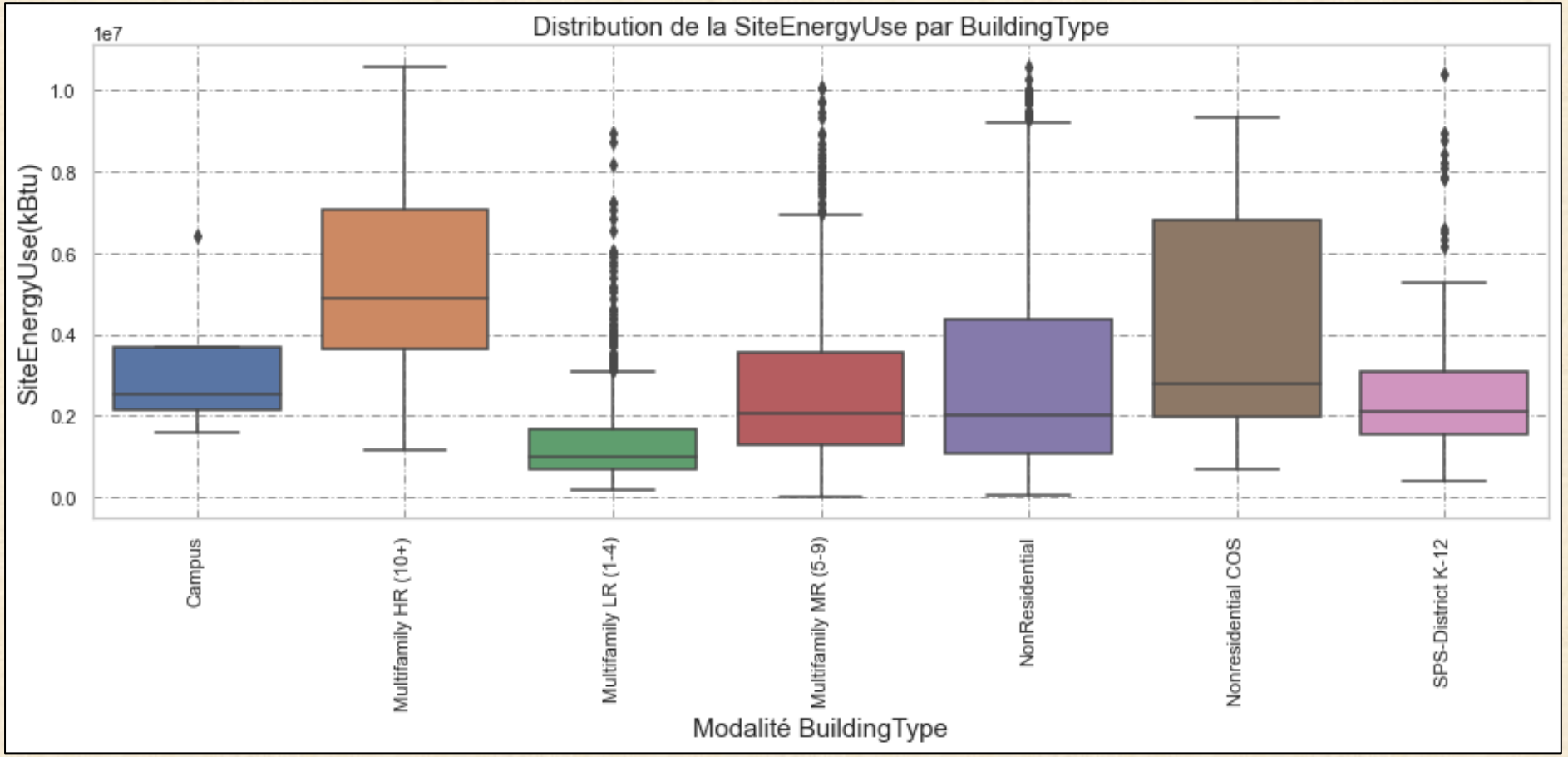
Analyse exploratoire

Analyse bivariée: la distribution de SiteEnergyUse par Neighborhood:



Analyse exploratoire

Analyse bivariée: la distribution de SiteEnergyUse par BuildingType:



C. Analyse exploratoire

C.2. Analyse bivariée:

On constate la présence de corrélations entre les variables suivantes:

- SiteEnergyUse et TotalGHGEmissions = 0.78

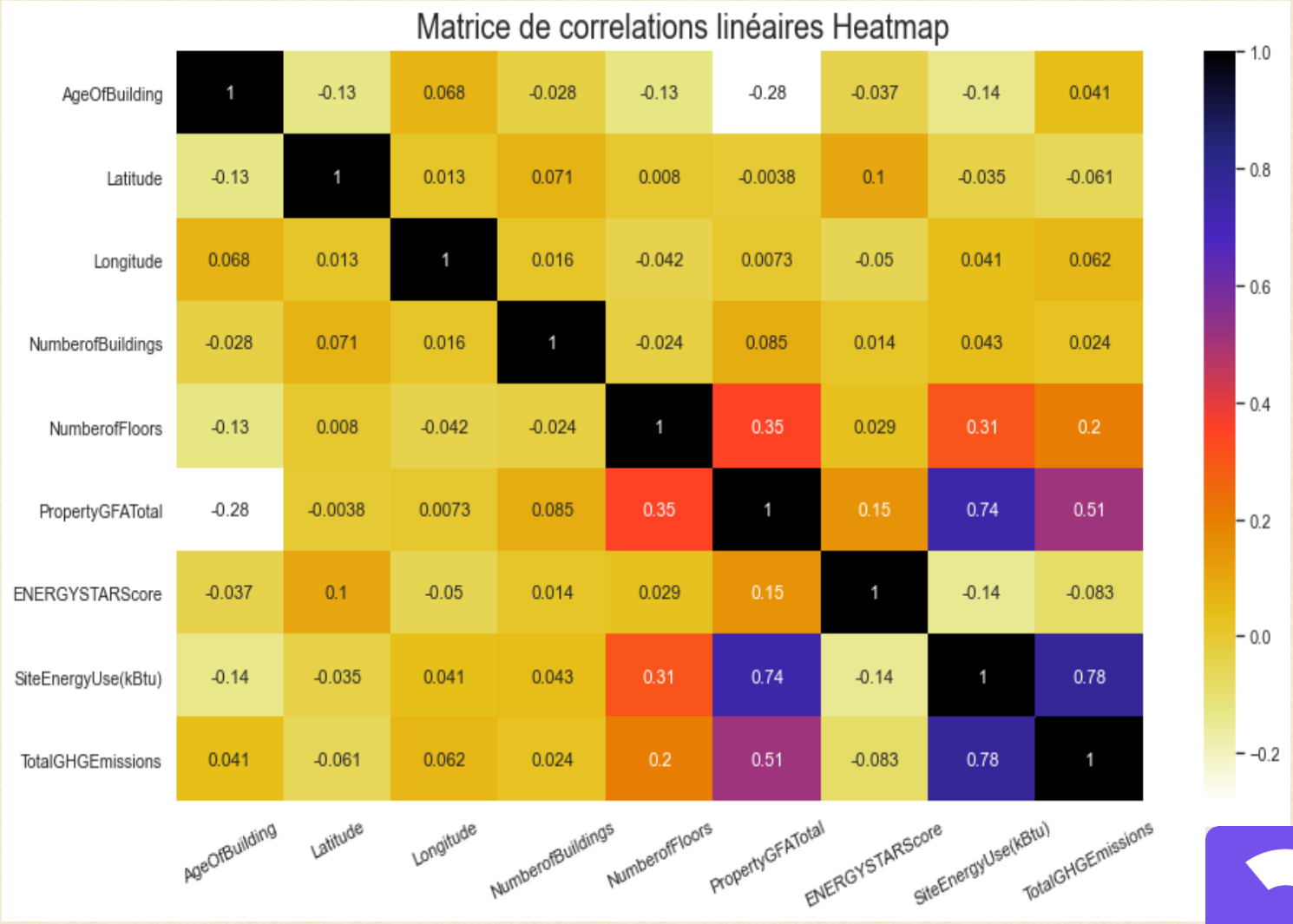
Une forte corrélation entre les deux plus la consommation augmentent plus les émissions augmentent

- TotalGHGEmissions et PropertyGFATotal= 0.51

- SiteEnergyUse et PropertyGFATotal = 0.74

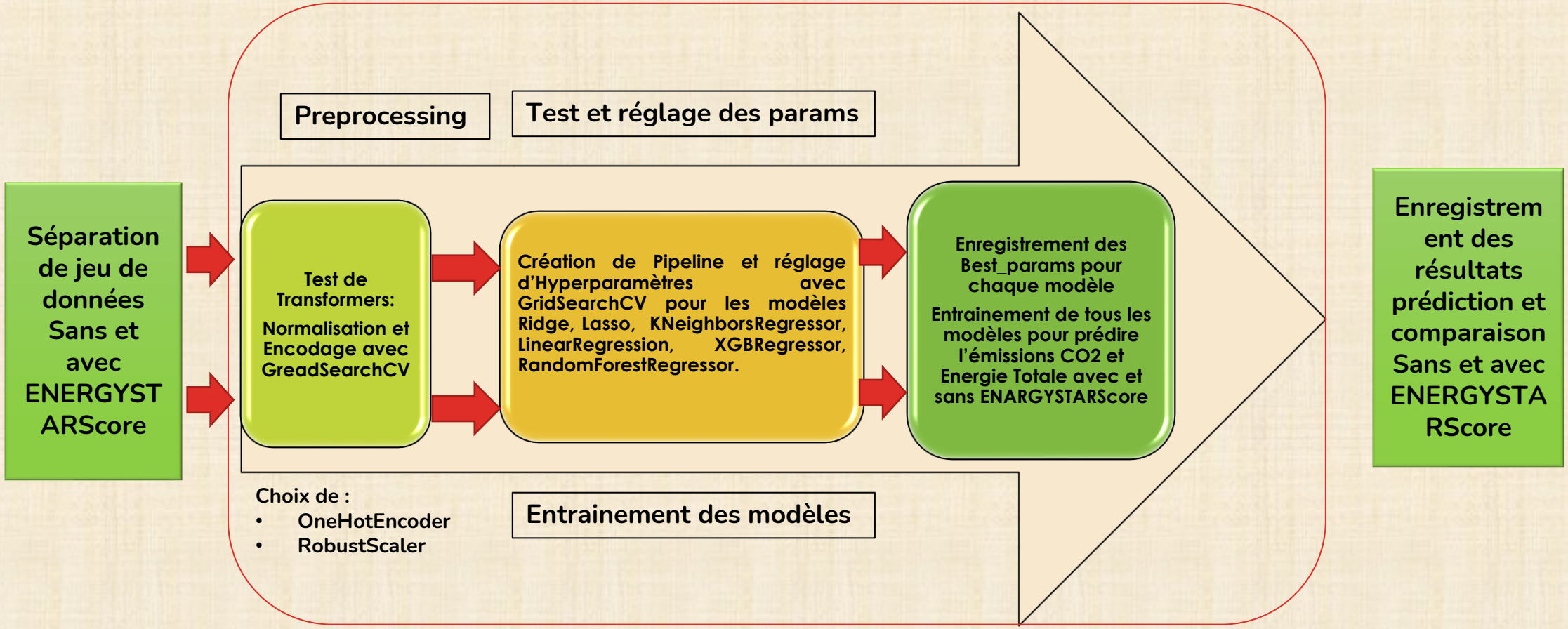
Il y a aussi une forte corrélation entre l'énergie cons et la suface totale GFA

- Concernant ENERGYSTARSScore on constate une faible corrélation avec le reste des variables



Modélisation

Démarche de modélisation:



Différents modèles

Test et paramétrage de modèles: Ex de Ridge

```
# pour l'imputation on a choisi SimpleImputer
float_trans = SimpleImputer(strategy="median")

# Et pour l'encodage OneHotEncoder
cat_trans = OneHotEncoder(sparse = False, handle_unknown="ignore")

preprocessor_ = ColumnTransformer(transformers=[
                                ("Numeric", float_trans, make_column_selector(dtype_include = np.number)),
                                ("OneHot" , cat_trans  , make_column_selector(dtype_exclude = np.number)) ],
                               remainder="passthrough")
```

```
# Utilisant la fonction train_split() pour la séparation de jeu donnée
X_train, X_test, y_train, y_test = train_test_split(X, y_SiteEnergyUse, test_size=0.2)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((1927, 9), (482, 9), (1927,), (482,))
```

```
# Définition du pipeline
ridge_pipe = Pipeline([("preprocessor", preprocessor_),
                       ("scaler", RobustScaler()),
                       ("estimator", Ridge())])
n_alphas = 10
alphas = np.logspace(-5, 5, n_alphas)
param_grid = {'estimator__alpha':alphas,
              'estimator__tol':[0.01, 0.03,0.05,0.07]}

ri_grid = GridSearchCV( ridge_pipe, param_grid, cv = 5, n_jobs = -1, verbose = 3)

ri_grid.fit(X_train, y_train)
```

- Ridge
- Lasso
- RandomForestRegressor
- XGBRegressor
- XGBRegressor
- LinearRegression

```
[[{'estimator__alpha': 0.2782559402207126, 'estimator__tol': 0.01}],
 [{'estimator__alpha': 599.4842503189421, 'estimator__tol': 0.1}],
 [{'estimator__leaf_size': 1,
   'estimator__n_neighbors': 15,
   'estimator__weights': 'distance'}],
 [{'estimator__n_estimators': 100}],
 [{'estimator__learning_rate': 0.05,
   'estimator__max_depth': 6,
   'estimator__n_estimators': 100}],
 [{'estimator__copy_X': True,
   'estimator__fit_intercept': False,
   'estimator__normalize': True}]]
```

Différents modèles

Test et paramétrage de modèles:

```
Fitting 5 folds for each of 40 candidates, totalling 200 fits

GridSearchCV(cv=5,
             estimator=Pipeline(steps=[('preprocessor',
                                        ColumnTransformer(remainder='passthrough',
                                                          transformers=[('Numeric',
                                                                         SimpleImputer(strategy='median'),
                                                                         <sklearn.compose._column_transformer.make_column_se
lector object at 0x00000208CA9316A0>),
                                                                        ('OneHot',
                                                                         OneHotEncoder(handle_unknown='ignore',
                                                                                       sparse=False),
                                                                         <sklearn.compose._column_transformer.make_column_se
lector object at 0x00000208CA1DBF40>)])),
                                       ('scaler', RobustScaler()),
                                       ('estimator', Ridge())]),
             n_jobs=-1,
             param_grid={'estimator__alpha': array([1.00000000e-05, 1.29154967e-04, 1.66810054e-03, 2.15443469e-02,
       2.78255940e-01, 3.59381366e+00, 4.64158883e+01, 5.99484250e+02,
       7.74263683e+03, 1.00000000e+05]),
                         'estimator__tol': [0.01, 0.03, 0.05, 0.07]},
             verbose=3)
```

```
1 ri_grid.best_params_
```

```
{'estimator__alpha': 0.2782559402207126, 'estimator__tol': 0.01}
```

```
1 ri_grid.best_score_
```

```
0.7166313257224435
```

```
1 ri_grid.score(X_test, y_test)
```

```
0.6996285132903666
```

```
1 ridge_mdl = ri_grid.best_estimator_
```

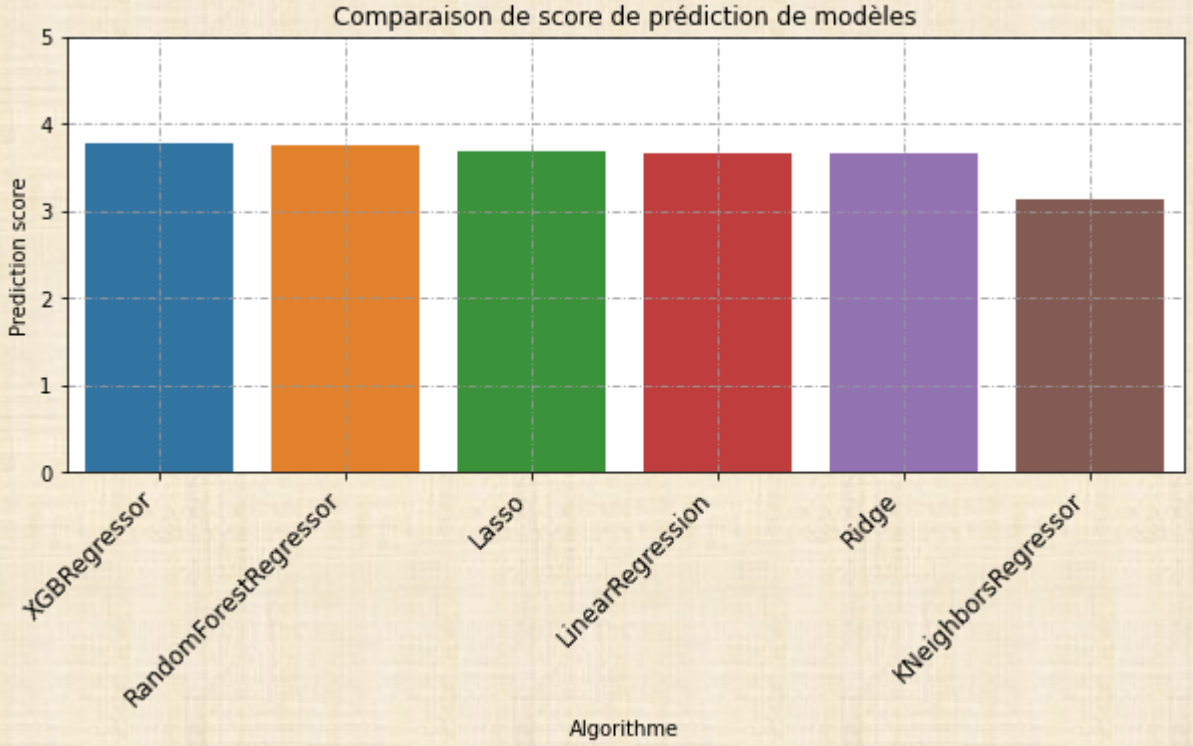
	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_estimator__alpha	param_estimator__tol	params
19	0.040199	7.360865e-03	0.006930	0.007205	0.278256	0.07	{'estimator...
18	0.046862	9.879328e-03	0.009372	0.007653	0.278256	0.05	{'estimator...
17	0.040615	7.651191e-03	0.009371	0.007652	0.278256	0.03	{'estimator...
16	0.043737	6.249785e-03	0.009373	0.007653	0.278256	0.01	{'estimator...
12	0.040614	7.652047e-03	0.009370	0.007651	0.021544	0.01	{'estimator...
13	0.037490	7.652086e-03	0.009372	0.007652	0.021544	0.03	{'estimator...
14	0.043738	6.247568e-03	0.009372	0.007652	0.021544	0.05	{'estimator...

split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
0.733557	0.682347	0.745201	0.697558	0.724494	0.716631	0.023254	1
0.733557	0.682347	0.745201	0.697558	0.724494	0.716631	0.023254	1
0.733557	0.682347	0.745201	0.697558	0.724494	0.716631	0.023254	1
0.733557	0.682347	0.745201	0.697558	0.724494	0.716631	0.023254	1
0.731734	0.681781	0.744910	0.698351	0.723747	0.716105	0.022916	5
0.731734	0.681781	0.744910	0.698351	0.723747	0.716105	0.022916	5
0.731734	0.681781	0.744910	0.698351	0.723747	0.716105	0.022916	5

Résultats SiteEnrgyUse et CO2

Résultats SiteEnrgyUse sans ENERGYSTARScore:

En dessous le résultat de prédiction de tous les modèles sans ENERGYSTARScore qui sont classés par « Prediction score » on voit bien que les deux modèles XGBRegressor et RandomForestRegressor ils ont un score bien meilleur que les autres modèles .



'without_EnergyStars'

	Algorithme	Training score	Prediction score	RMSE	MAE	Median abs err
5	XGBRegressor	0.896470	0.755356	702211.343943	1.087316e+06	418234.421875
4	RandomFores...	0.964531	0.750773	711300.655217	1.097452e+06	400163.937267
2	Lasso	0.731961	0.734279	753427.119761	1.133185e+06	455008.564522
0	LinearRegre...	0.733019	0.733448	754365.912827	1.134956e+06	455379.029121
1	Ridge	0.732762	0.732078	757220.863397	1.137870e+06	456769.511903
3	KNeighborsR...	1.000000	0.626263	858585.778540	1.343913e+06	504850.820124

Résultats SiteEnrgyUse et CO2

Résultats SiteEnrgyUse avec ENERGYSTARScore:

En dessous le résultat de prédiction de tous les modèles avec ENERGYSTARScore qui sont classés par « Prediction score » on voit bien que les deux modèles XGBRegressor et RandomForestRegressor ils ont un score bien meilleur que les autres modèles .

'with_EnergyStars'

	Algorithme	Training score	Prediction score	RMSE	MAE	Median abs err
5	XGBRegressor	0.937198	0.862763	540042.863345	8.362501e+05	352245.187500
4	RandomFores...	0.969861	0.846467	576772.388718	8.845098e+05	350596.369959
2	Lasso	0.777954	0.789451	673547.367237	1.035805e+06	440496.917909
1	Ridge	0.779176	0.787560	679668.026583	1.040446e+06	437546.356484
0	LinearRegre...	0.779326	0.784533	681910.399676	1.047832e+06	437245.505404
3	KNeighborsR...	1.000000	0.701416	823732.004507	1.233486e+06	521209.732771

Résultats SiteEnrgyUse et CO2

Résultats CO2 sans et avec ENERGYSTARScore:

```
# Modèles à tester
mdl_ = {
    'LinearRegression'      : LinearRegression(copy_X= True,fit_intercept= False,normalize= True),
    'Ridge'                 : Ridge(alpha= 0.2782559402207126, tol= 0.01),
    'Lasso'                 : Lasso(alpha= 599.4842503189421, tol= 0.1),
    'KNeighborsRegressor'   : KNeighborsRegressor(leaf_size= 1,n_neighbors= 15,weights= 'distance'),
    'RandomForestRegressor' : RandomForestRegressor(max_depth=None, n_estimators= 100),
    'XGBRegressor'          : XGBRegressor(learning_rate= 0.05, max_depth= 6, n_estimators= 100)
    }
```

	Algorithme	Training score	Prediction score	RMSE	MAE	Median abs err
5	XGBRegressor	0.812485	0.458436	26.767344	41.595171	17.011964
4	RandomFores...	0.927989	0.407396	27.343807	43.511116	16.922925
1	Ridge	0.480183	0.355654	30.074681	45.370936	19.302765
0	LinearRegre...	0.481086	0.353767	30.101168	45.437308	19.457187
3	KNeighborsR...	1.000000	0.317080	28.853885	46.709258	16.747819
2	Lasso	0.000000	-0.000392	41.429722	56.533158	37.795077

	Algorithme	Training score	Prediction score	RMSE	MAE	Median abs err
4	RandomFores...	0.930238	4.659640e-01	3.010626e+01	4.736281e+01	17.903475
5	XGBRegressor	0.840633	4.625731e-01	3.000327e+01	4.751294e+01	17.196985
1	Ridge	0.489647	4.283513e-01	3.293464e+01	4.900234e+01	21.266504
3	KNeighborsR...	1.000000	4.086764e-01	3.093430e+01	4.983848e+01	17.717074
2	Lasso	0.000000	-2.539851e-03	4.580369e+01	6.489377e+01	37.660000
0	LinearRegre...	0.490242	-2.329901e+23	1.424943e+12	3.128390e+13	21.541677

Test des modèles non paramétrés

```
mdl = {
    'LinearRegression'      : LinearRegression(),
    'Ridge'                 : Ridge(),
    'Lasso'                 : Lasso(),
    'KNeighborsRegressor'   : KNeighborsRegressor(),
    'RandomForestRegressor' : RandomForestRegressor( ),
    'XGBRegressor'          : XGBRegressor()
    }
```

```
score_with_EnergyStars_co2_np
```

	Algorithme	Training score	Prediction score	RMSE	MAE	Median abs err
4	RandomFores...	0.931292	0.524492	27.786386	44.915362	16.111250
5	XGBRegressor	0.977748	0.507142	28.452177	45.727448	16.226764
1	Ridge	0.494617	0.431895	32.005185	49.094229	19.889769
0	LinearRegre...	0.500088	0.424018	32.355153	49.433413	20.165000
2	Lasso	0.381024	0.386526	33.456430	51.016923	21.450541
3	KNeighborsR...	0.589998	0.360313	32.113351	52.095448	18.802000

Modèles retenu et l'impact EnergyStarScore

Seattle

Modèles retenu:

Le modèle retenu dans les deux cas avec et sans ENERGYSTARScore c'est XGBRegressor

	Algorithme	Training score	Prediction score	RMSE	MAE	Median abs err
6	XGBRegressor	0.937198	0.862763	540042.863345	8.362501e+05	352245.187500
7	RandomFores...	0.969861	0.846467	576772.388718	8.845098e+05	350596.369959
8	Lasso	0.777954	0.789451	673547.367237	1.035805e+06	440496.917909
9	Ridge	0.779176	0.787560	679668.026583	1.040446e+06	437546.356484
10	LinearRegre...	0.779326	0.784533	681910.399676	1.047832e+06	437245.505404
0	XGBRegressor	0.896470	0.755356	702211.343943	1.087316e+06	418234.421875
1	RandomFores...	0.964531	0.750773	711300.655217	1.097452e+06	400183.937267
2	Lasso	0.731961	0.734279	753427.119761	1.133185e+06	455008.564522
3	LinearRegre...	0.733019	0.733448	754365.912827	1.134956e+06	455379.029121
4	Ridge	0.732762	0.732078	757220.863397	1.137870e+06	456769.511903
11	KNeighborsR...	1.000000	0.701416	823732.004507	1.233486e+06	521209.732771
5	KNeighborsR...	1.000000	0.626263	858585.778540	1.343913e+06	504850.820124

Avec

Sans

Impact EnergyStarScore:

L'Energy Star Score est un score qui reflète 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.

- L'influence sur la prédiction de la consommation d'énergie totale est remarquable comme c'est illustré sur le tableau à gauche
- Par contre l'impact sur la prédiction des émissions CO2 il est faible par rapport à la prédiction de l'énergie totale

Conclusion

❑ **Prédiction de la consommation d'énergie totale et les émission CO2:**

- La modélisation est faite avec 5 modèles avec des résultats acceptable
- Les deux meilleurs modèles XGBRegressor et RandomForestRegressor
- La variable ENERGYSTARScore a une faible corrélation avec les autres Features mais un impact remarquable sur la prédiction des modèles

❑ **Axes d'amélioration:**

- Pour améliorer la performance des modèles il faudrait optimiser les paramètres des différents algorithmes via la validation croisée