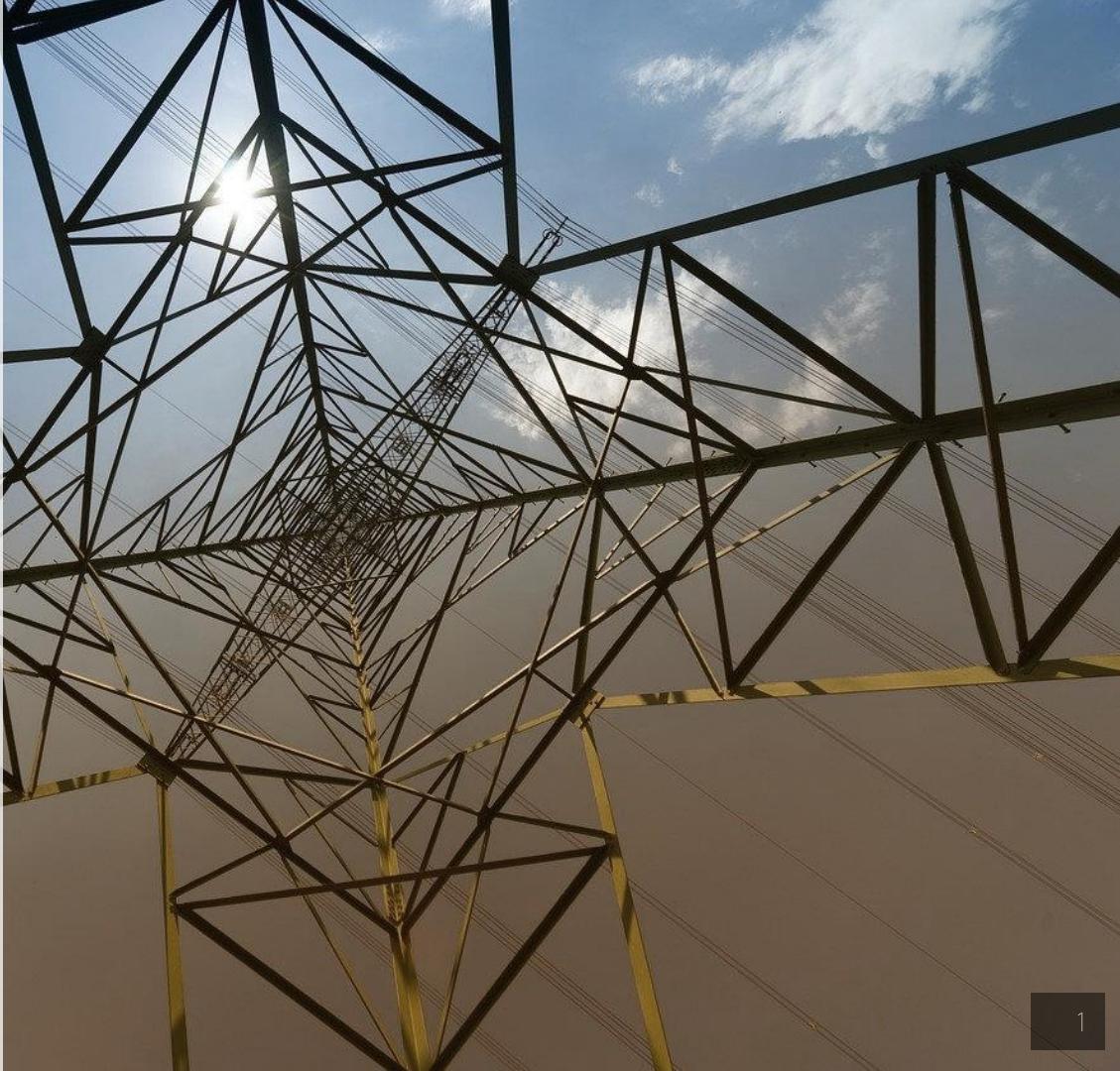


# Projet 4

## Data scientist

Anticipez les besoins  
en consommation  
électrique de  
bâtiments



## 1. présentation des données

- a, le jeu de données de 2015
- b, le jeu de données de 2016
- c, regroupement des données

## 2. nettoyage des données

- a, les bâtiments qui ne sont pas à usage d'habitation
- b, les outliers
- c, les données manquantes et négatives
- d, la colonne Primary\_property\_type

## 3. analyse des données

- a, premières analyses univariées
- b, analyse logarithmique
- c, premières analyses bivariées
- d, détection d'outliers par la méthode Isolation forest
- e, test de normalité des variables énergie et GHGE
- f, analyses finales

## 4. mise en place d'un modèle

- a, préparation des données
- b, choix des algorithmes
- c, méthode
- d, modèle de prédiction de la variable d'émission des gaz à effet de serre
- e, modèle de prédiction de la variable énergie site

## 5. pertinence de la variable énergie star score

- a, essai pour la prédiction de la variable d'émission des gaz à effet de serre
- b, essai pour la prédiction de la variable énergie site

# 1, présentation des données

- a, le jeu de données de 2015
- b, le jeu de données de 2016
- c, regroupement des données



# a, le jeu de données de 2015

3340 lignes et 47 colonnes

26512 données manquantes soit **17%**

Pas de doublons

10 colonnes présentent en 2015 mais pas en 2016

**56 lignes** présentent en 2015 mais pas en 2016

Création des colonnes **prop\_elec** et **prop\_gaz**



# b, le jeu de données de 2016

3376 lignes et 46 colonnes

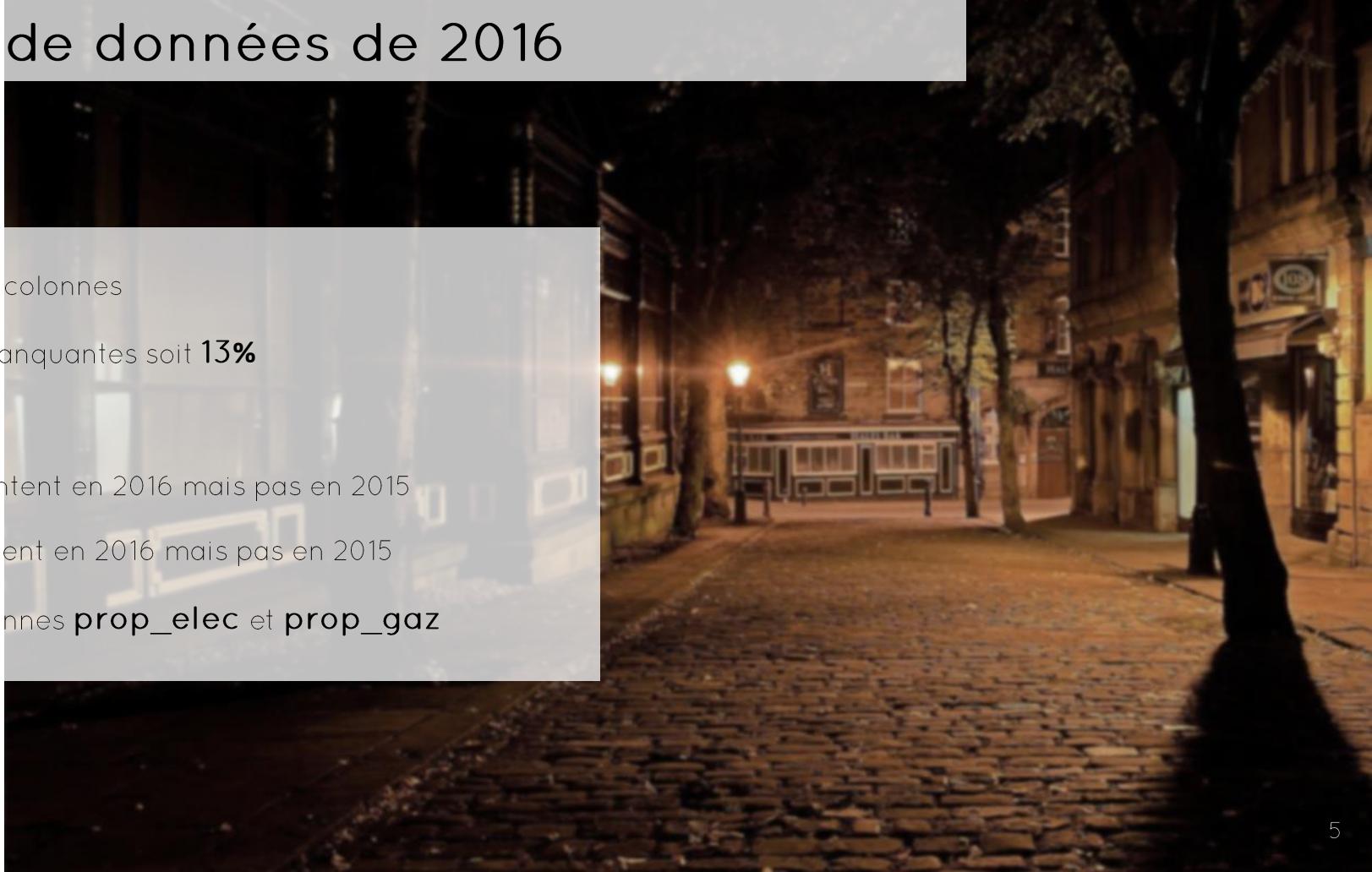
19952 données manquantes soit **13%**

Pas de doublons

9 colonnes présentent en 2016 mais pas en 2015

**92 lignes** présentent en 2016 mais pas en 2015

Création des colonnes **prop\_elec** et **prop\_gaz**



# c, regroupement des données

Concaténation des données de **2016** et des **56** lignes seulement présentes en 2015.

Liste des colonnes retenues:

PrimaryPropertyType

BuildingType

YearBuilt

NumberofBuildings

NumberofFloors

PropertyGFATotal

PropertyGFAParking

LargestPropertyUseTypeGFA

prop\_elec

prop\_gaz

Outlier

SiteEnergyUse(kBtu)

GHGEmissions(MetricTonsCO2e)

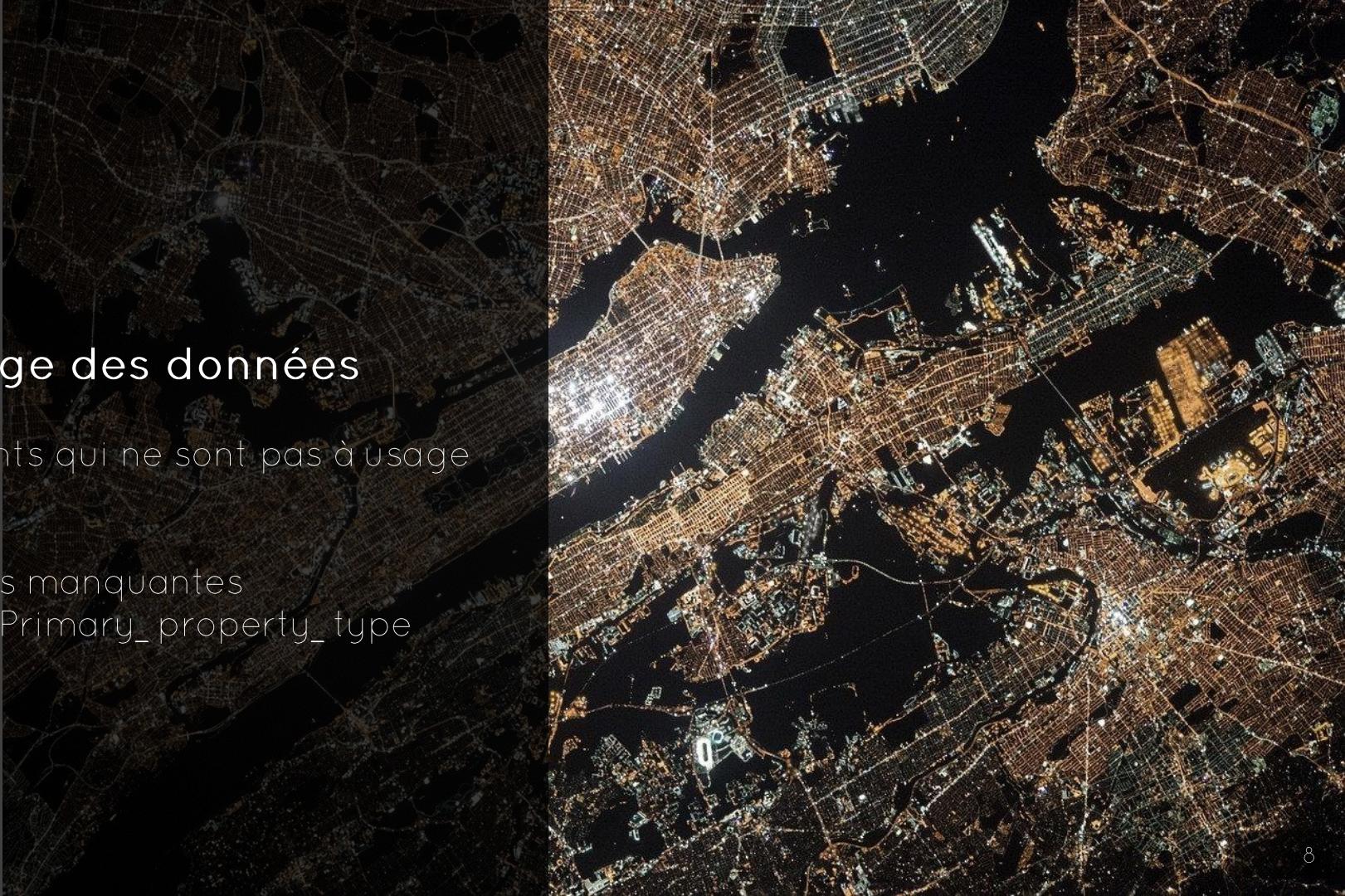




Finally we arrive at a dataset of **3432** lines and  
**13** columns  
With 3495 missing values  
**13.24%**

## 2, nettoyage des données

- a, les bâtiments qui ne sont pas à usage d'habitation
- b, les outliers
- c, les données manquantes
- d, la colonne Primary\_property\_type



# a, les bâtiments qui ne sont pas à usage d'habitation



```
df.BuildingType.unique()
```

```
array(['NonResidential', 'Nonresidential COS', 'Multifamily MR (5-9)',  
       'SPS-District K-12', 'Campus', 'Multifamily LR (1-4)',  
       'Multifamily HR (10+)', 'Nonresidential WA'], dtype=object)
```

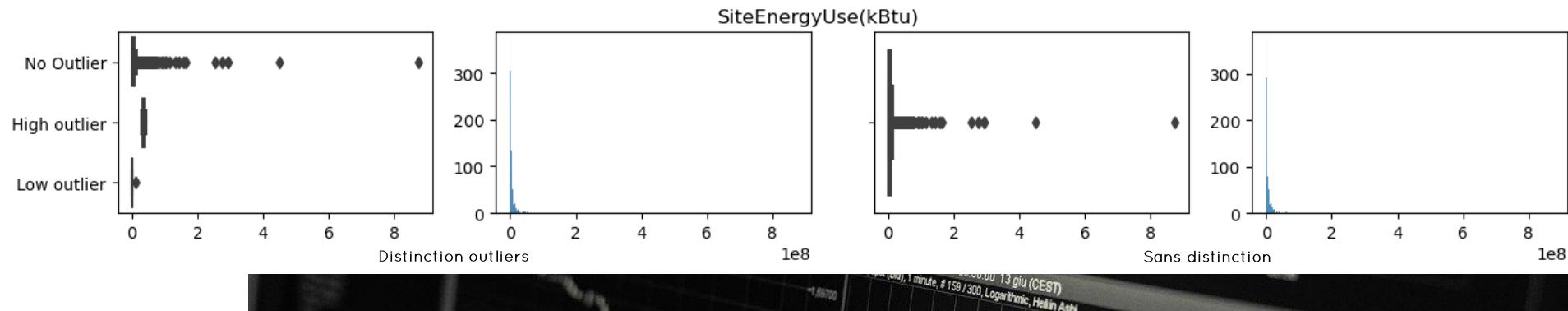
1734 lignes concernent des habitations à usage

d'habitation soit **50%** des données

Nous les supprimons

Nous arrivons à un jeu de donné de **1698** lignes et 12 colonnes Avec 1717 données manquantes, soit **12.13 %**

# b, les outliers



	SiteEnergyUse (kBtu)	count	mean	std	min	25%	50%	75%	max
Outlier									
High outlier	2.0	3.426835e+07	1.008498e+07	2.713719e+07	3.070277e+07				
Low outlier	17.0	8.288096e+05	2.675096e+06	1.680890e+04	1.008417e+05				
No Outlier	1676.0	8.523142e+06	3.037979e+07	0.000000e+00	1.245740e+06				

Les données estampillées 'Outlier' sont clairement des données extrêmes, néanmoins leur impact est relativement réduit. Il y a 2 outliers high et 17 outliers low.

Nous décidons quand même de les supprimer.

# c, les données manquantes et négatives

Après la suppression de la colonne outlier, il reste **38 données manquantes** réparties sur 19 lignes, soit **0,25%**.

Vu leur faible quantité, nous les **supprimons**.

**1 seul ligne** contient une donnée négative. Elle concerne les émissions gaz à effet de serre.

Même si cela est techniquement possible, nous écartons ce bâtiment.

```
# for c in df.columns:  
#     if df[c].dtypes != 'object':  
#         name = c  
#         valeur = df[df[c] < 0].shape[0]  
#         if valeur > 0:  
#             print(name, valeur)  
#             print(df[df[c] < 0].index)
```

```
TotalGHGEmissions 1  
Int64Index([1588], dtype='int64')
```

# c, la colonne Primary\_property\_type

```
df.PrimaryPropertyType.unique()
```

```
array(['Hotel', 'Other', 'Mixed Use Property', 'K-12 School',
       'University', 'Small- and Mid-Sized Office',
       'Self-Storage Facility', 'Warehouse', 'Large Office',
       'Senior Care Community', 'Medical Office', 'Retail Store',
       'Hospital', 'Residence Hall', 'Distribution Center',
       'Worship Facility', 'Supermarket / Grocery Store', 'Laboratory',
       'Refrigerated Warehouse', 'Restaurant', 'Low-Rise Multifamily',
       'Office', 'Non-Refrigerated Warehouse', 'Restaurant\n'],
      dtype=object)
```

Suppression des lignes contenant **family**

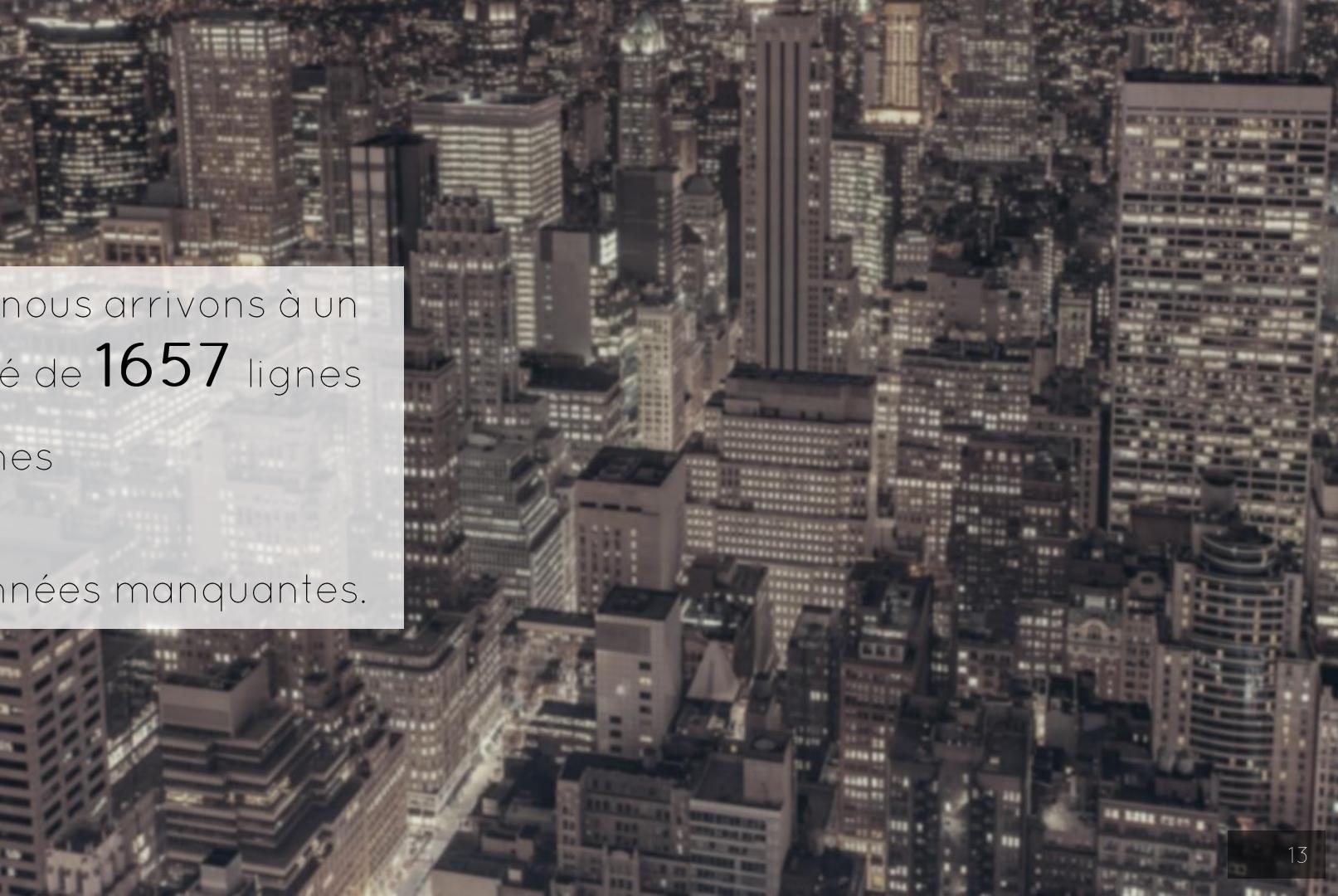
Correction des labels finissant par \n

Regroupement des **warehouse**

Passage de **24** valeur à **20**.

```
df1.PrimaryPropertyType.unique()
```

```
array(['Hotel', 'Other', 'Mixed Use Property', 'K-12 School',
       'University', 'Small- and Mid-Sized Office',
       'Self-Storage Facility', 'Warehouse', 'Large Office',
       'Senior Care Community', 'Medical Office', 'Retail Store',
       'Hospital', 'Residence Hall', 'Distribution Center',
       'Worship Facility', 'Supermarket / Grocery Store', 'Laboratory',
       'Restaurant', 'Office'], dtype=object)
```

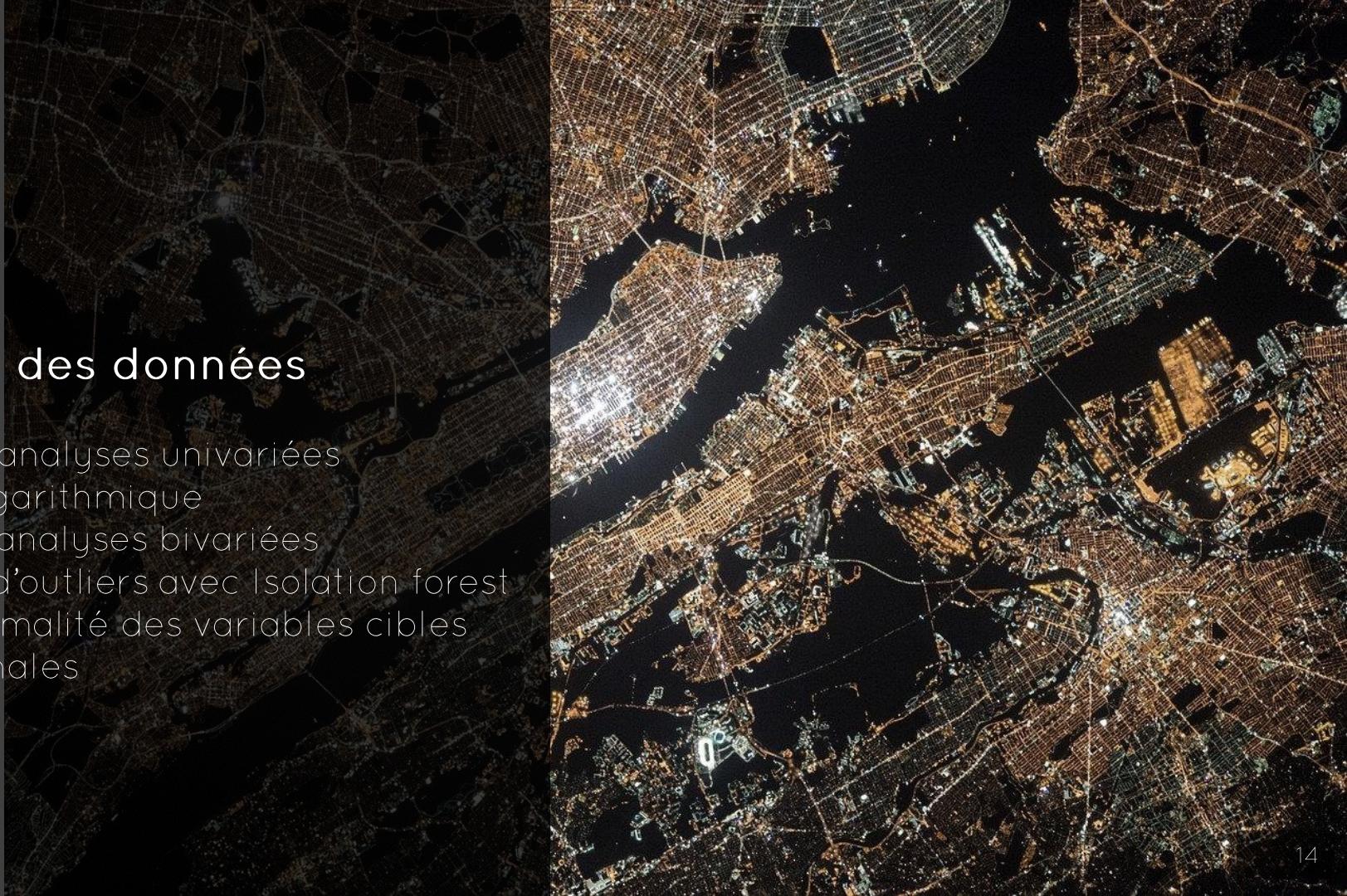


Finally we arrive at a dataset of **1657** lines  
and **11** columns

Average **0** missing values.

### 3, analyse des données

- a, premières analyses univariées
- b, analyse logarithmique
- c, premières analyses bivariées
- d, détection d'outliers avec Isolation forest
- e, test de normalité des variables cibles
- f, analyses finales

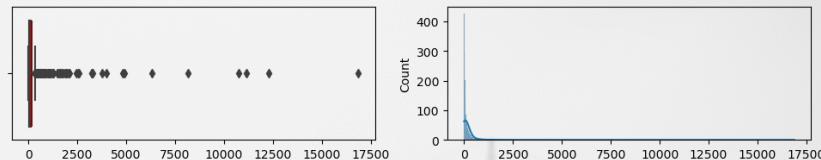


# a, première analyse univariées

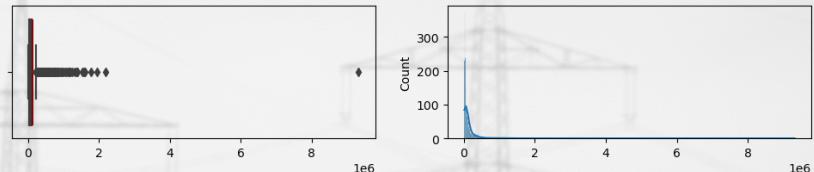
Malgré la suppression des outliers,  
les données sont encore très  
**disparates.**

Afin d'atténuer le problème, nous  
allons passer les données au  
**logarithme** et chercher si il y a  
**d'autres outliers** que ceux  
relevés par les équipes  
municipales

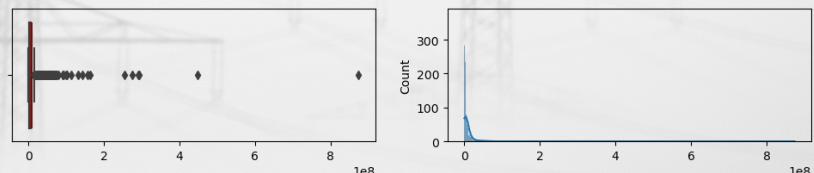
Analyse univariée de TotalGHGEmissions



Analyse univariée de PropertyGFTotal



Analyse univariée de SiteEnergyUse(kBtu)

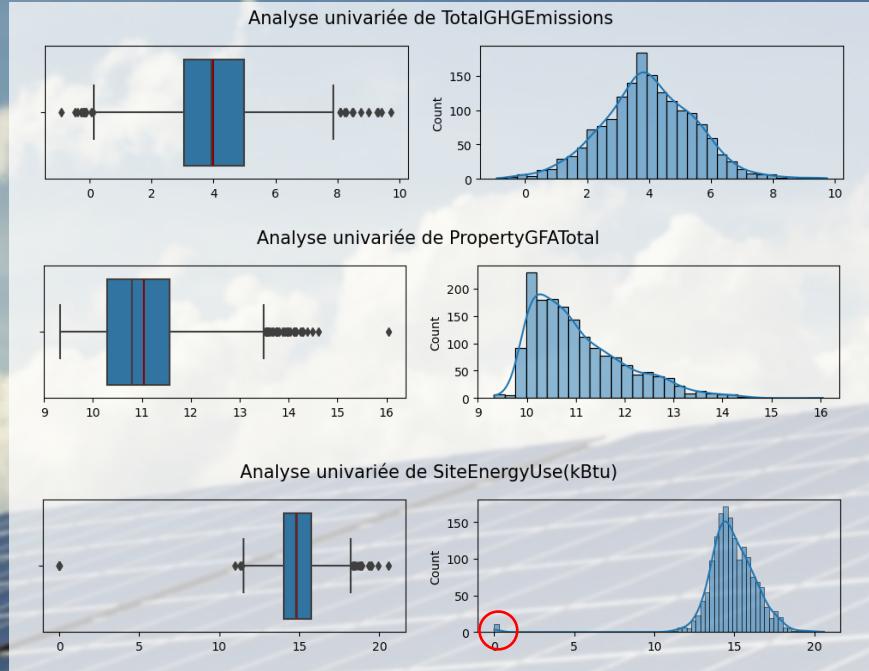


# b, analyse logarithmiques

Les analyses logarithmiques nous permettent de mieux voir la **distribution de données**.

On note que les **données cibles** pourraient être distribuées selon la **loi normale**.

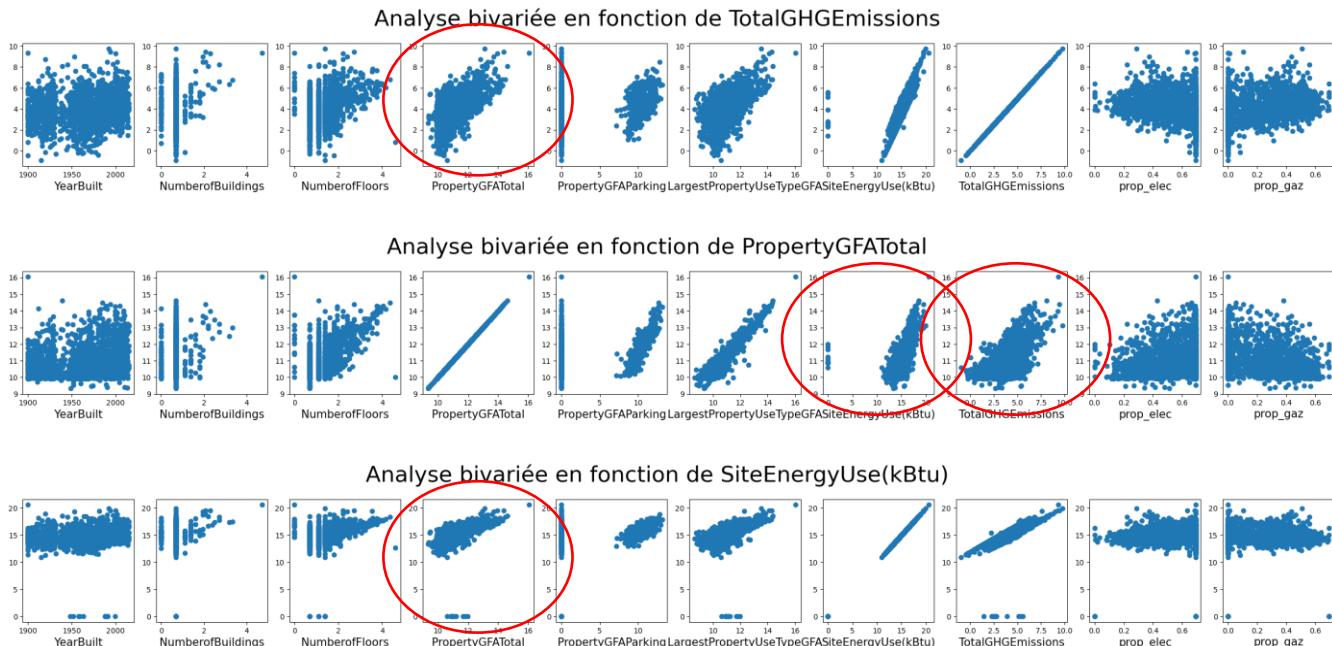
Néanmoins, certains graphiques font encore apparaître des **données extrêmes**.



# c, première analyses bivariées

Les analyses bivariées nous montrent des relations, + ou - franches entre les variables.

Notamment les variables **cibles** avec property GFA total



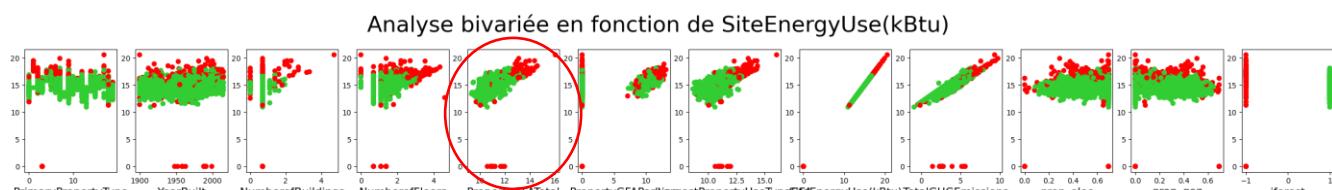
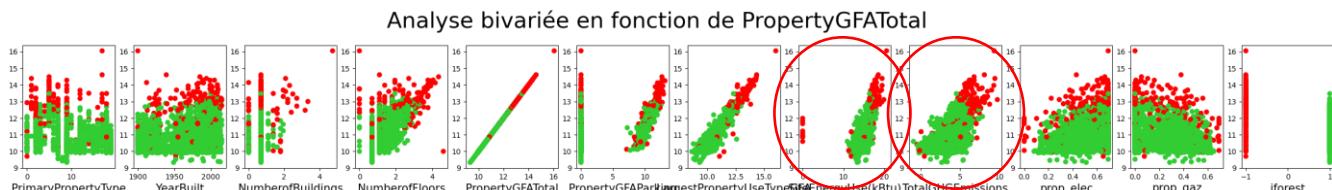
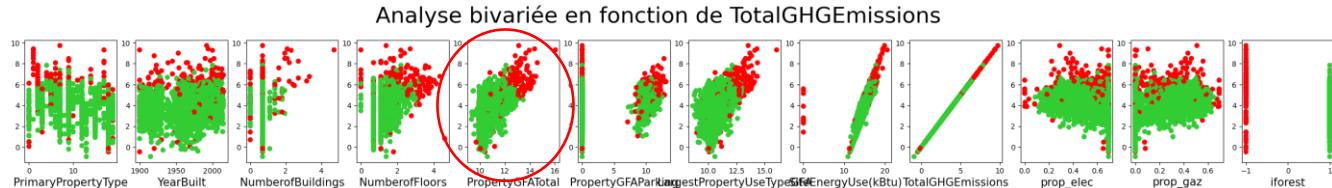
# d, détection d'outliers avec isolation forest

L'algorithme isolation forest trouve 187 valeurs aberrantes soit environ

**10 %** du jeu de données.

Nous les supprimons pour rendre notre modèle

**plus général.**

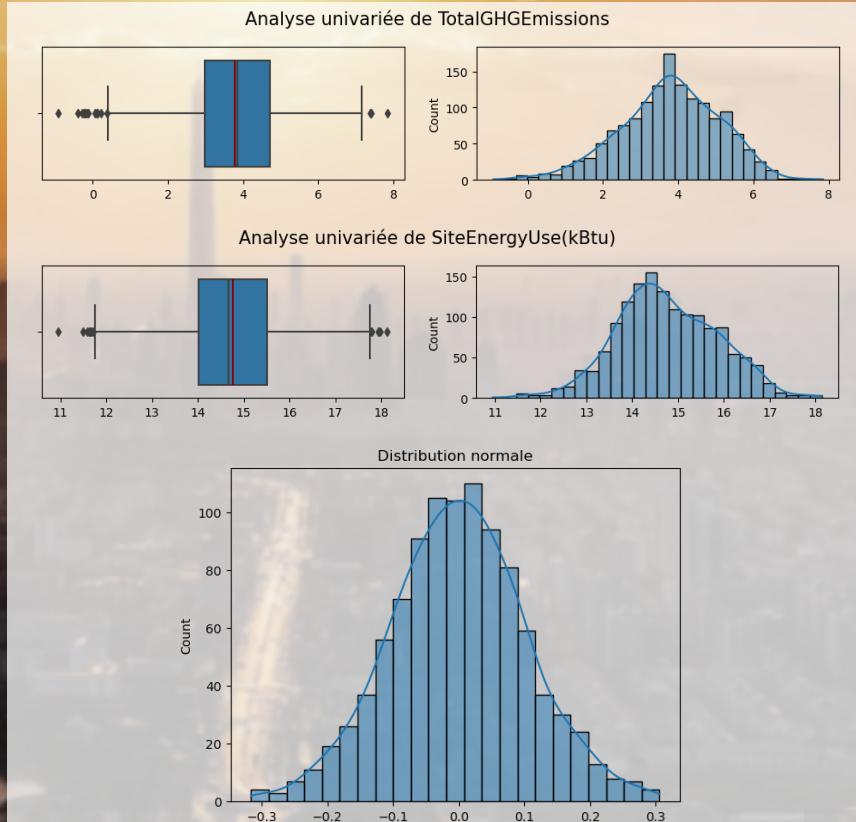


# e, test de normalité des variables cibles

Trois tests de normalité  
(D'Agostino/Pearson, Shapiro,  
Anderson)

**rejettent l'hypothèse**  
selon laquelle les courbes seraient  
distribuées normalement.

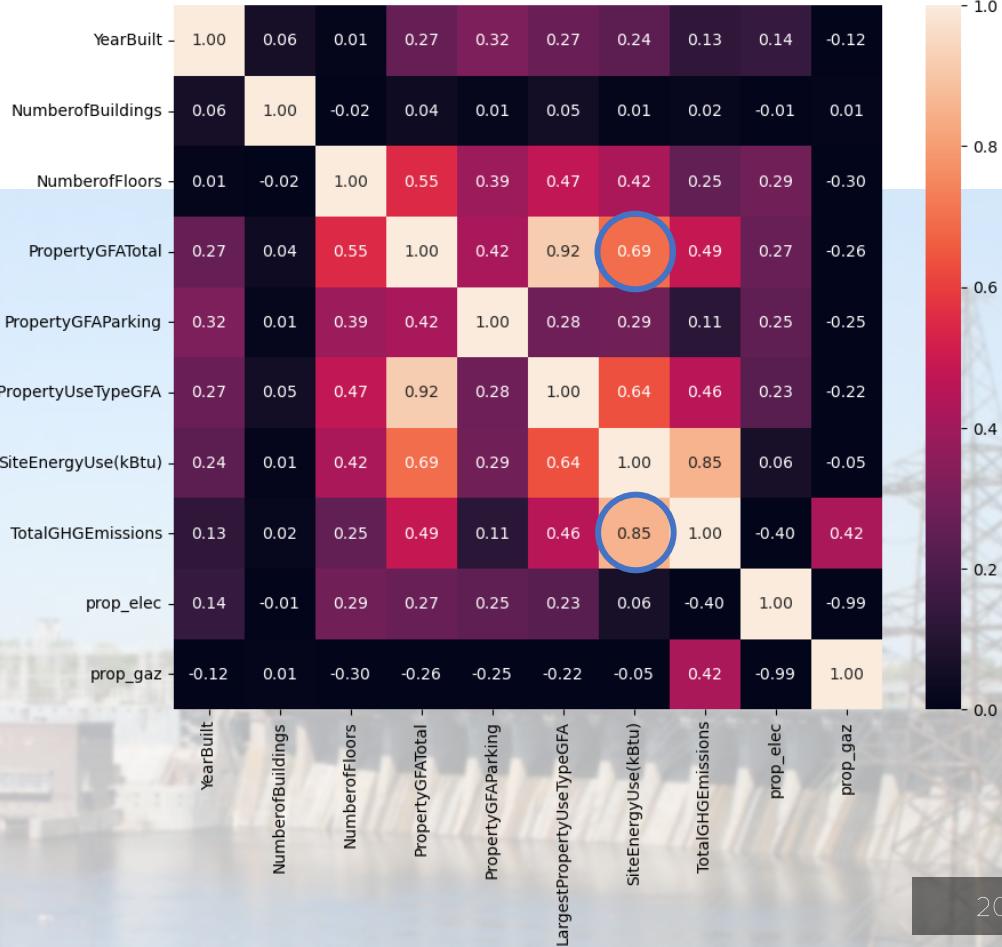
Graphiquement, on peut le voir car  
la courbe GHGE a un **skewed  
gauche** alors que la courbe  
énergie a un **skewed à droite**.



# f, analyses finales

La matrice de corrélation confirme notre impression graphique concernant la **taille de la propriété et la consommation d'énergie.**

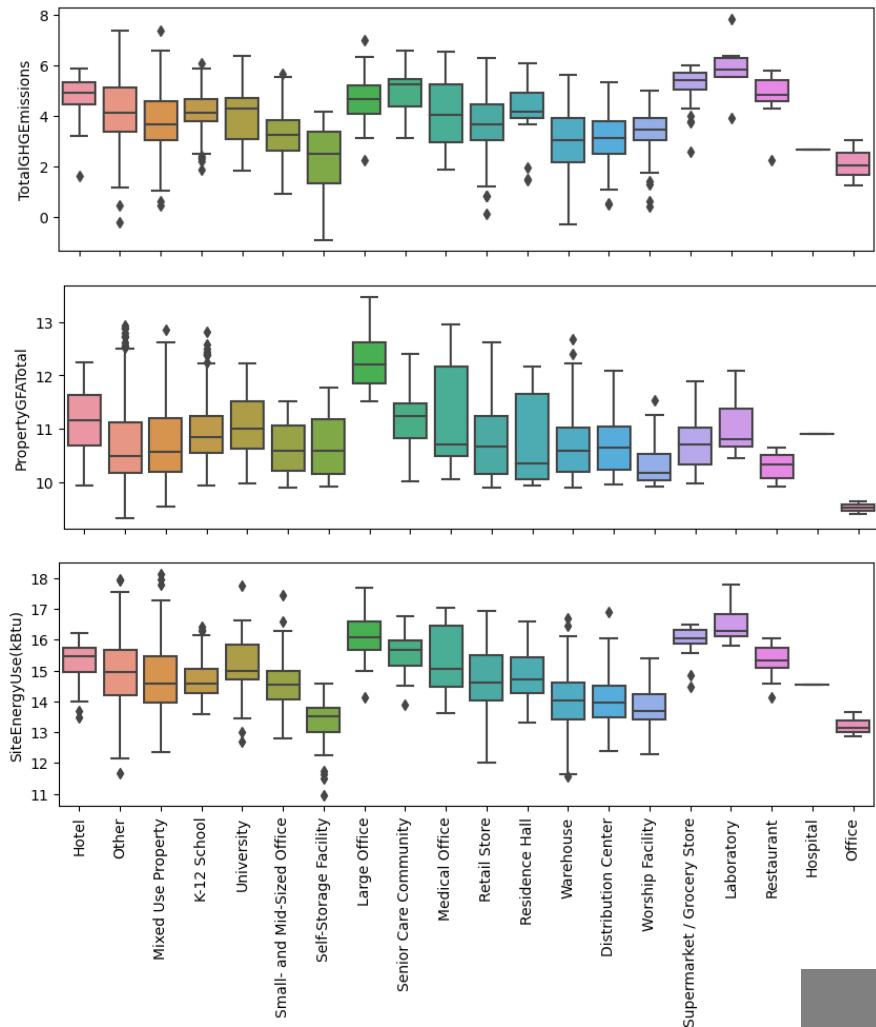
Elle confirme également le **lien fort entre les deux variables cibles.**



Concernant la seule variable qualitative du jeu de donnée (primary property type), son croisement avec les autres variables fait également ressortir des **différences de consommations entre types de bâtiments.**

ANOVA entre property type et energie obtient un R2 de 0,377.

OLS Regression Results		
Dep. Variable:	dfl[i]	R-squared:
Model:	OLS	Adj. R-squared:
Method:	Least Squares	F-statistic:
Date:	Mon, 20 Sep 2021	Prob (F-statistic):
Time:	12:41:16	Log-Likelihood:
No. Observations:	1478	AIC:
Df Residuals:	1458	BIC:
Df Model:	19	
Covariance Type:	nonrobust	



## 4, mise en place d'un modèle

- a, préparation des données
- b, choix des algorithmes
- c, méthode
- d, modèle de prédiction de la variable d'émission des gaz à effet de serre
- e, modèle de prédiction de la variable energie site



# a, préparation des données

## Normalisation des données

Méthode `StandardScaler()`  $z=(x-u) / s$   
Avec  $u$  moyenne de l'échantillon et  $s$  sa déviation standard.

## Labellisation de la variable

PrimaryPropertyType  
Méthode `OneHotEncoder()`

Transformation de la colonne en 20 colonnes binaires.

## Création d'un échantillon d'entraînement et de test

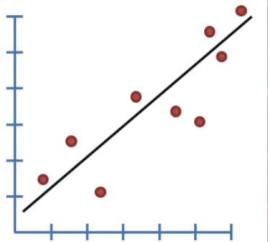
Méthode `train_test_split()`



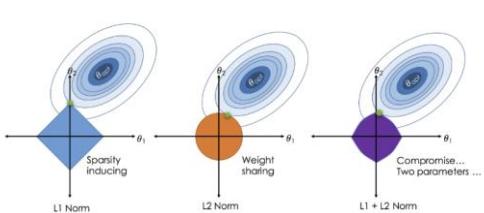
# b, choix des algorithmes

les linéaires

régression linéaire

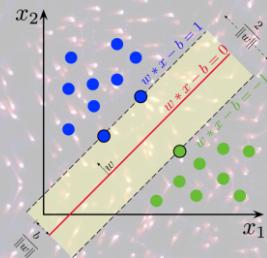


elastic Net

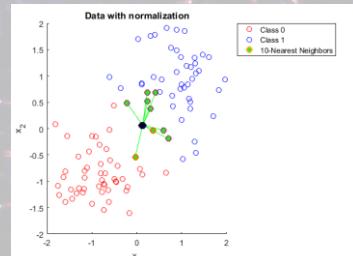


les non-linéaires

SVR

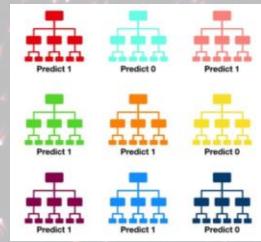


knn Regressor



les ensemblistes

random forest



gradient boost



# c, la méthode

## définition des paramètres

```
model_params = {
    'linear_regression' : {
        'model' : LinearRegression(),
        'params' : {}
    },
    'elastic_net' : {
        'model' : ElasticNet(),
        'params' : {
            'alpha' : [0.001, 0.01, 0.1, 1., 10],
            'l1_ratio' : np.linspace(0.1,1,5),
        }
    },
    'SVR' : {
        'model' : svm.SVR(kernel='rbf'),
        'params' : {
            'C' : [0.1, 1, 3, 5, 10],
            'gamma' : [0.1, 0.5, 1.0]
        }
    },
    'knn_regressor' : {
        'model' : neighbors.KNeighborsRegressor(),
        'params' : {
            'n_neighbors' : [i for i in range(2,15)]
        }
    },
    'random_forest' : {
        'model' : RandomForestRegressor(oob_score=True),
        'params' : {
            'n_estimators' : [50,100,1000],
            'min_samples_split' : [10, 20, 50, 100]
        }
    },
    'gradient_boost' : {
        'model' : GradientBoostingRegressor(),
        'params' : {
            'n_estimators' : [50,100,1000],
            'loss' : ['ls', 'lad'],
        }
    }
}
```

## définition des métriques à observer

$$\text{scoring} = \text{R2}$$

Coefficient de détermination

```
scores.append({
    'model' : model_name,
    'r2' : clf.best_score_,
    'best_params' : clf.best_params_,
    'RMSE' : np.sqrt(metrics.mean_squared_error(y_test, y_pred)),
    'medAE' : metrics.median_absolute_error(y_test, y_pred),
    'MAE_%' : metrics.mean_absolute_percentage_error(y_test, y_pred),
    'time' : clf.refit_time_
})
```

### RMSE

Erreur quadratique moyenne (racine carré).

### MEDAE

Erreur médiane absolue permet une désensibilisation aux données aberrantes.

### MAE %

Erreur moyenne absolue en %

### Temps

Temps nécessaire pour réentraîner le meilleur modèle sur l'ensemble du dataset.

## Validation croisée

méthode GridSearchCV()

5 validations croisées

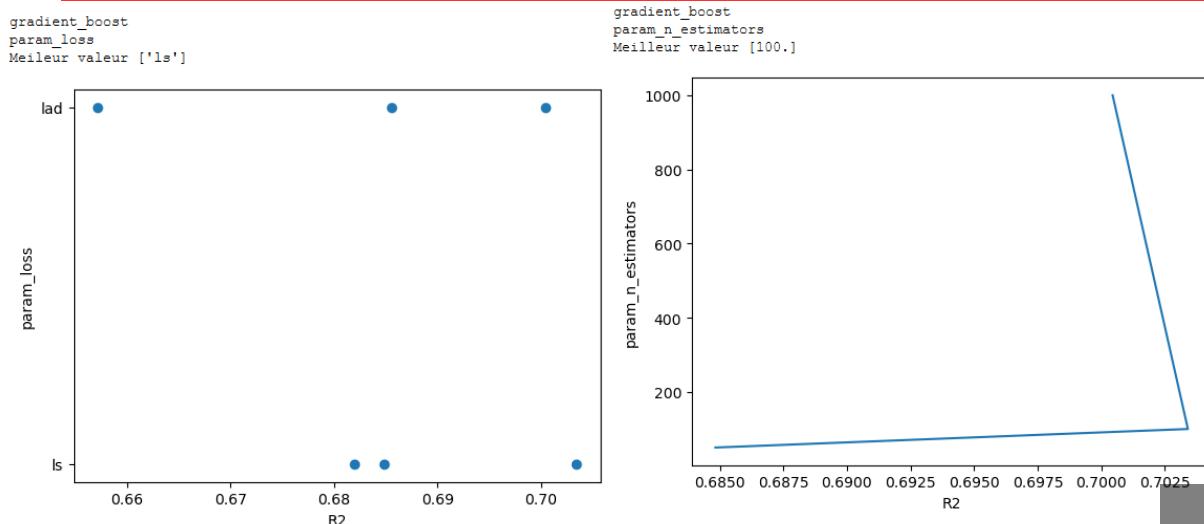
# d, modèle de prédiction de la variable GHGE

## 1, les résultats

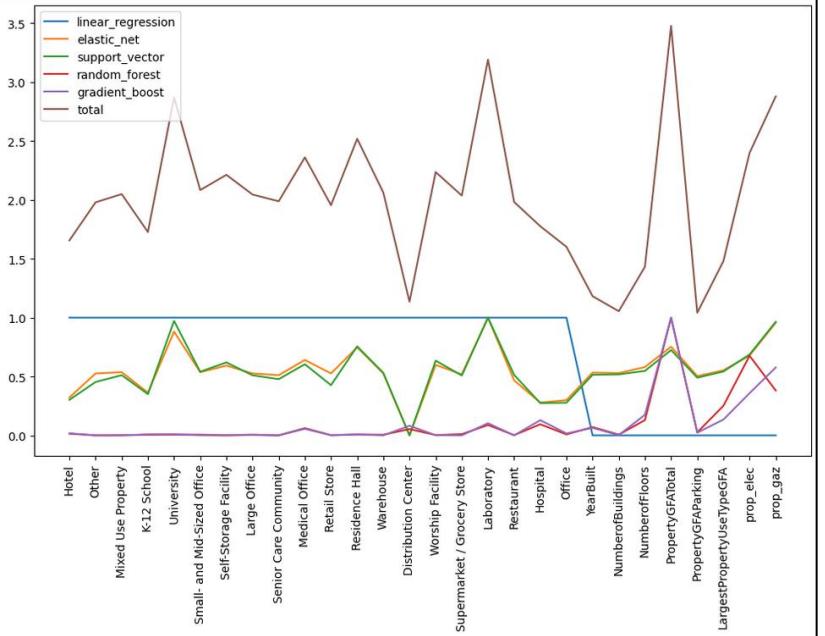
L'algorithme du **gradient boost** est celui qui nous donne les meilleurs résultats.

Cependant, l'**elastic net** fait juste un peu moins bien mais en beaucoup moins de temps.

	model	r2	best_params	RMSE	medAE	MAE_%	time
0	linear_regression	-4.512309e+22	{}	0.705943	0.375393	0.184435	0.001000
1	elastic_net	6.964264e-01	{'alpha': 0.001, 'l1_ratio': 0.55}	0.704390	0.384879	0.184933	0.004968
2	SVR	6.962448e-01	{'C': 3, 'gamma': 0.1}	0.711183	0.344540	0.181743	0.059918
3	knn_regressor	6.070910e-01	{'n_neighbors': 12}	0.818261	0.449705	0.220056	0.000000
4	random_forest	6.541690e-01	{'min_samples_split': 20, 'n_estimators': 1000}	0.714782	0.427087	0.191157	2.871004
5	gradient_boost	7.034064e-01	{'loss': 'ls', 'n_estimators': 100}	0.680427	0.347664	0.174683	0.120913

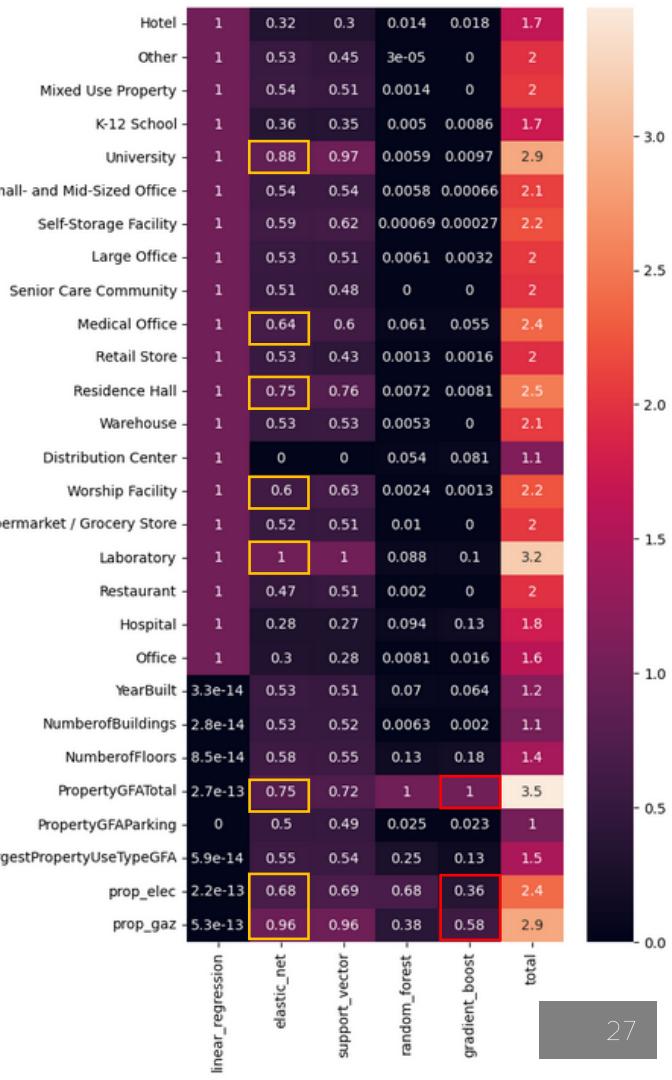


## 2, le poids des variables



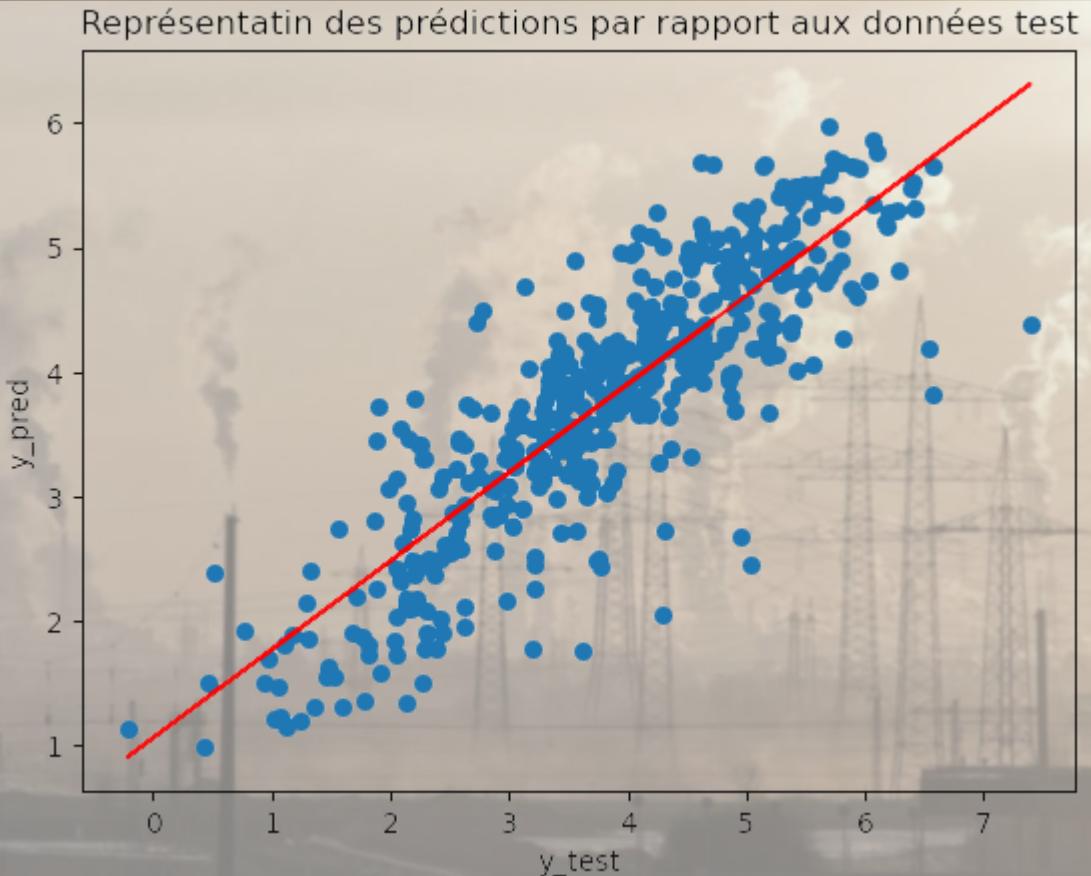
Le gradient boost se focalise sur la taille de la propriété et le type d'énergie consommée.

L'elastic net regarde les mêmes informations mais est également sensible à certains types de propriété.



R2 0,70

On voit une relation linéaire entre les prédictions du modèle et les données test.



# e, modèle de prédiction de la variable Energie site

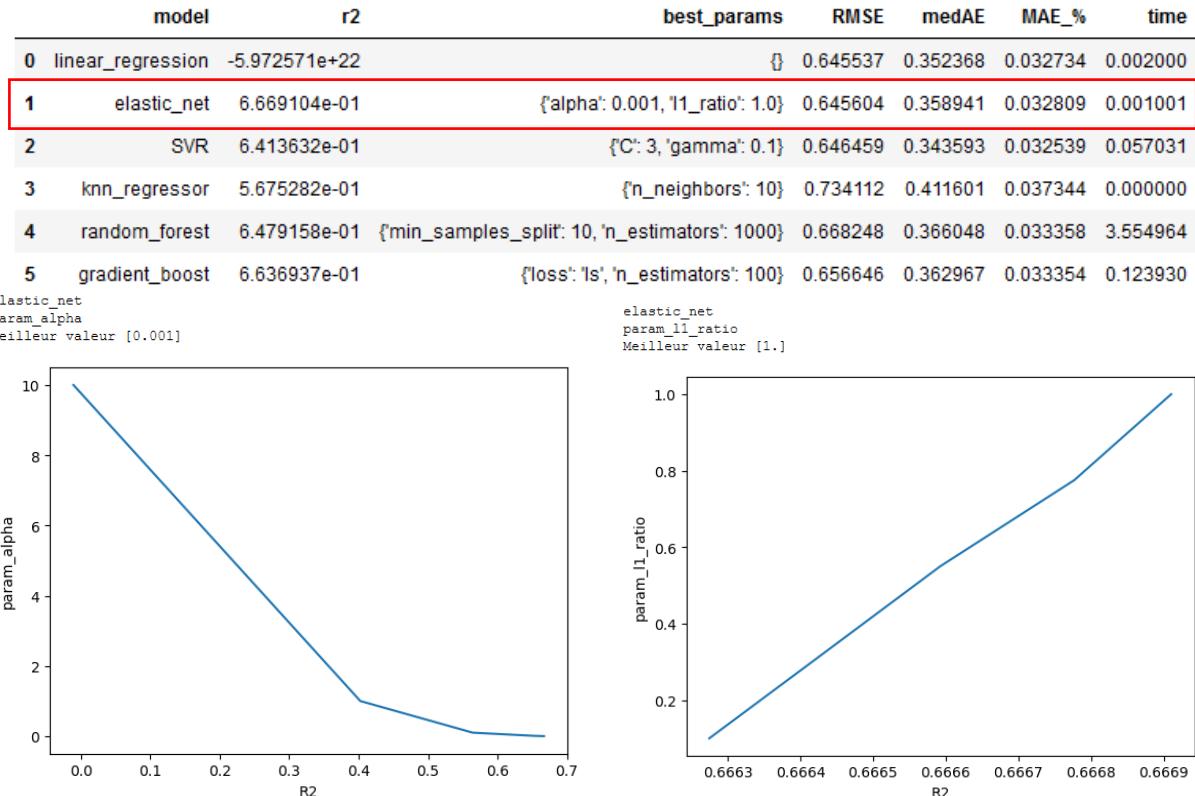
1, les résultats

Cette fois-ci c'est l'algorithme de l'elastic net qui affiche les meilleures performances.

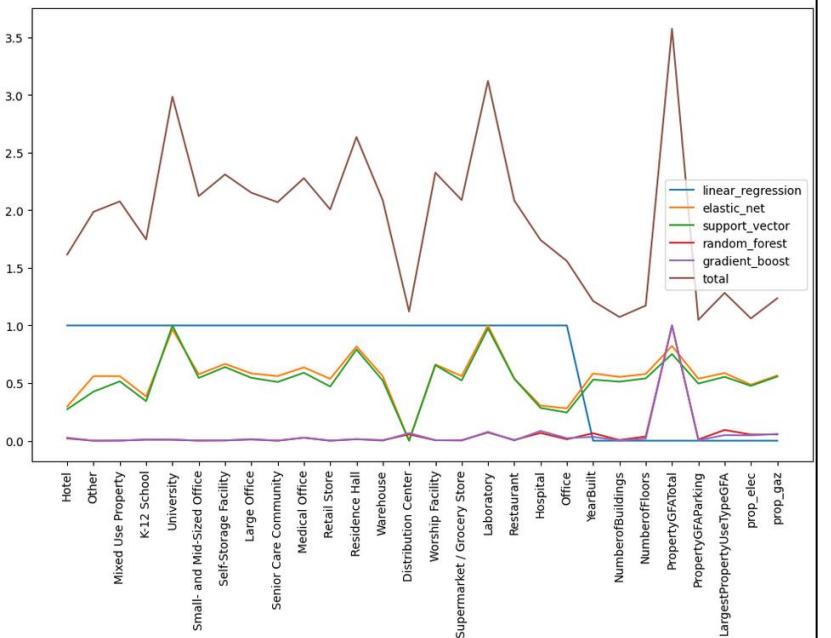
Les paramètres retenus sont:

alpha=0,001, cette faible pénalité nous rapproche d'une régression linéaire classique

L1\_ratio=1, pénalité L1, donc l'elastic net se comporte ici comme une régression lasso (en triant les variables).



## 2, le poids des variables



Pour déterminer la consommation d'énergie du site, le modèle a accordé plus d'**importance** au **type de propriété** par rapport à celui déterminant les émissions de gaz à effet de serre.

La **taille** de la propriété est toujours déterminante.

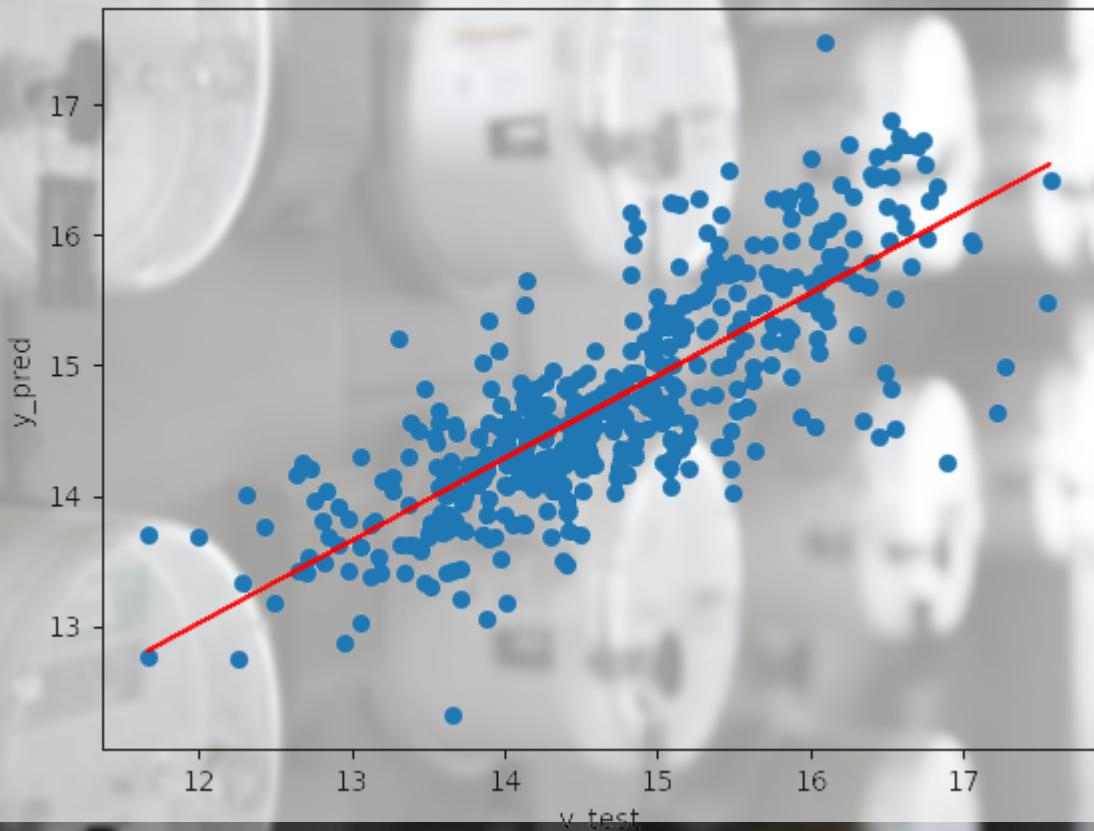
	linear_regression	elastic_net	support_vector	random_forest	gradient_boost	total	
Hotel	1	0.3	0.27	0.02	0.026	1.6	- 3.5
Other	1	0.56	0.43	0	0	2	- 2.8
Mixed Use Property	1	0.56	0.51	0.0014	0	2.1	- 2.5
K-12 School	1	0.38	0.34	0.0086	0.01	1.7	- 2.3
University	1	0.97	1	0.0078	0.01	3	- 2.0
Small- and Mid-Sized Office	1	0.58	0.54	0.0014	0	2.1	- 2.0
Self-Storage Facility	1	0.67	0.64	0.0019	0.0029	2.3	- 2.0
Large Office	1	0.58	0.55	0.013	0.0095	2.2	- 2.0
Senior Care Community	1	0.56	0.51	7.1e-05	0	2.1	- 2.0
Medical Office	1	0.64	0.59	0.026	0.027	2.3	- 2.0
Retail Store	1	0.54	0.47	0.0002	0	2	- 2.0
Residence Hall	1	0.82	0.79	0.013	0.013	2.6	- 2.0
Warehouse	1	0.56	0.52	0.0045	0.00014	2.1	- 2.0
Distribution Center	1	0	0	0.055	0.066	1.1	- 2.0
Worship Facility	1	0.66	0.66	0.0038	0.0048	2.3	- 2.0
Supermarket / Grocery Store	1	0.56	0.52	0.0044	0.00069	2.1	- 2.0
Laboratory	1	1	0.97	0.071	0.077	3.1	- 2.0
Restaurant	1	0.54	0.54	0.0061	0.0021	2.1	- 2.0
Hospital	1	0.3	0.28	0.067	0.085	1.7	- 2.0
Office	1	0.28	0.24	0.012	0.022	1.6	- 2.0
YearBuilt	1e-12	0.58	0.53	0.065	0.034	1.2	- 2.0
NumberofBuildings	7.3e-13	0.55	0.51	0.0058	0.0017	1.1	- 2.0
NumberofFloors	9.8e-13	0.58	0.54	0.036	0.017	1.2	- 2.0
PropertyGFATotal	3.3e-12	0.82	0.75	1	1	3.6	- 2.0
PropertyGFAParking	5.6e-13	0.54	0.5	0.01	0.0041	1	- 2.0
LargestPropertyUseTypeGFA	1.2e-12	0.59	0.55	0.093	0.049	1.3	- 2.0
prop_elec	0	0.49	0.48	0.053	0.047	1.1	- 2.0
prop_gaz	8e-13	0.57	0.56	0.055	0.059	1.2	- 2.0

R2 0,66

Ici aussi on voit la relation entre les prédictions et les données test.

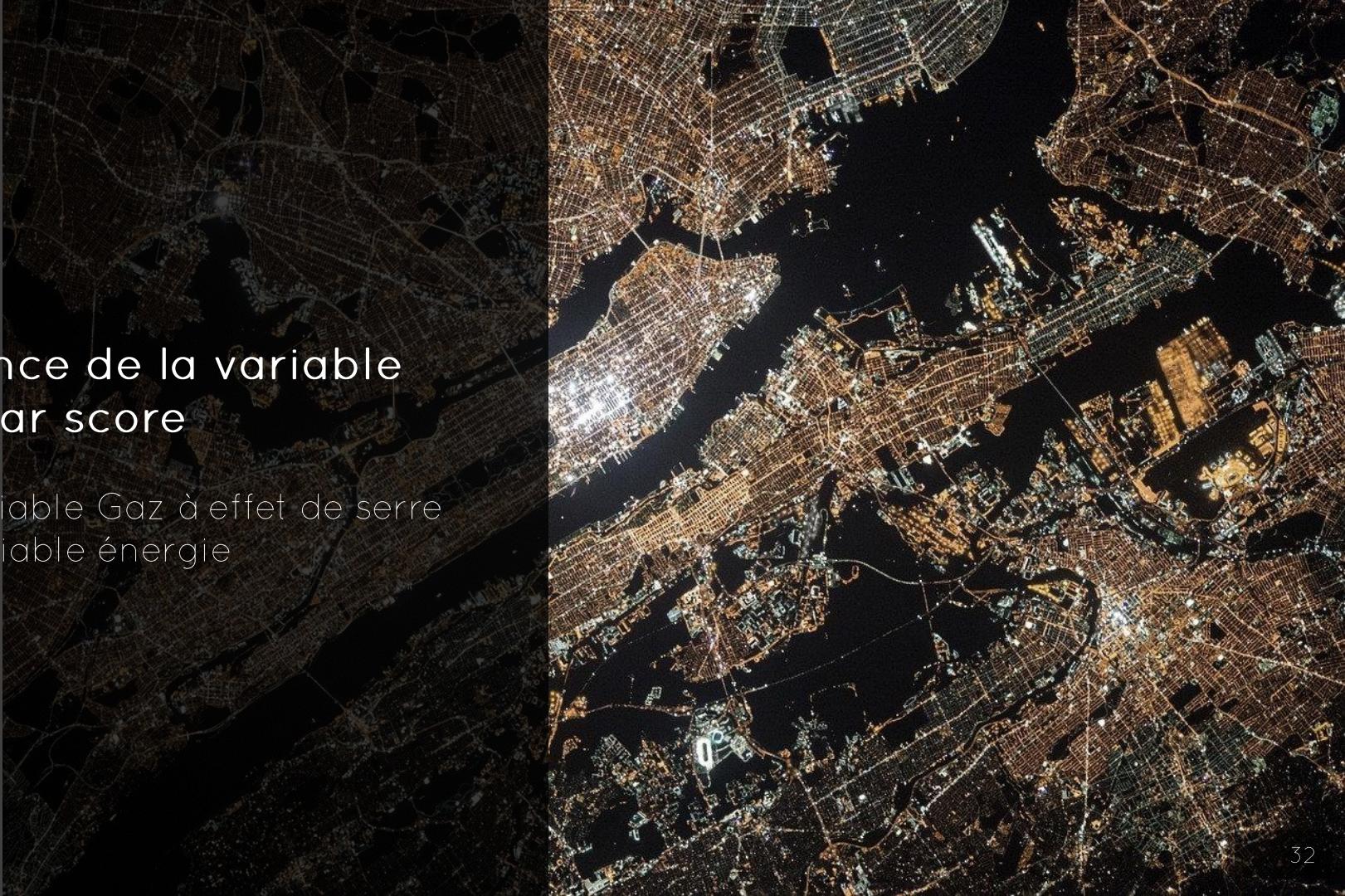
Néanmoins, comme les métriques nous le laissaient deviner, le nuage de point est plus diffus.

Représentation des prédictions par rapport aux données test



## 5, pertinence de la variable Energie star score

a, pour la variable Gaz à effet de serre  
b, pour la variable énergie



# a, pour la variable gaz à effet de serre

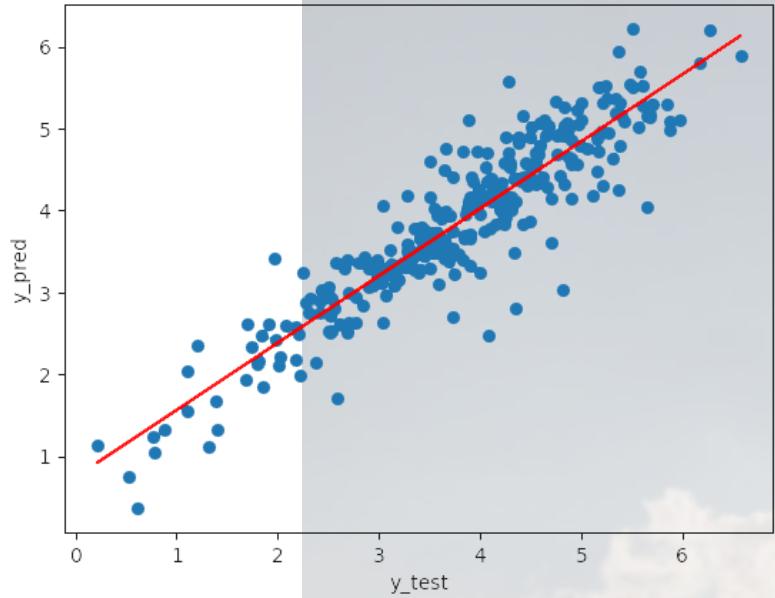
Comme dans le 1<sup>er</sup> modèle, le **gradient boost** est celui qui nous donne le meilleur coefficient de détermination.

	model	r2	best_params	RMSE	medAE	MAE_%	time
0	linear_regression	0.824912	{}	0.472391	0.189782	0.112407	0.002001
1	elastic_net	0.825813	{"alpha": 0.001, "l1_ratio": 1.0}	0.473179	0.189979	0.113150	0.004004
2	SVR	0.820300	{"C": 1, "gamma": 0.1}	0.468296	0.208631	0.114867	0.026999
3	knn_regressor	0.726958	{"n_neighbors": 3}	0.610423	0.337521	0.147277	0.000000
4	random_forest	0.761950	{"min_samples_split": 10, "n_estimators": 1000}	0.535025	0.281363	0.141674	2.312200
5	gradient_boost	0.826517	{"loss": "ls", "n_estimators": 100}	0.470566	0.262454	0.123376	0.093002

L'ajout de la variable Energie Star score permet des **gains significatifs** de précision du modèle, et ce **pour toutes les valeurs** (medAE - 23%)

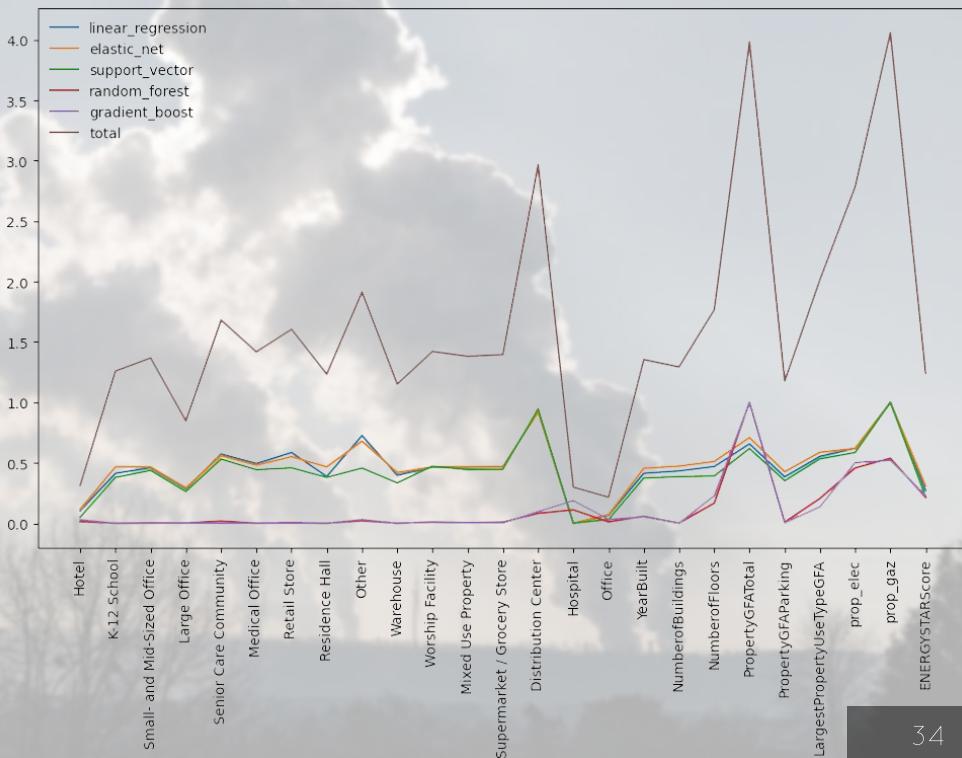
	SANS ENERGIE STAR	AVEC ENERGIE STAR	DIFFÉRENCE EN VALEUR	DIFFÉRENCE EN %
R2	0,70	0,82	0,12	17%
RMSE	0,68	0,47	0,21	-30%
MEDAE	0,34	0,26	0,08	-23%

## Représentation des prédictions par rapport aux données test



R2 0,82

Bien que le modèle n'accorde pas un poids très conséquent à la variable Energie star (- de 0,3), sa prise en compte influe **positivement** sur les prédictions.



# b, pour la variable énergie

1, les résultats

Comme précédemment,  
l'elastic net obtient le  
meilleur score et avec des  
paramètres identiques.

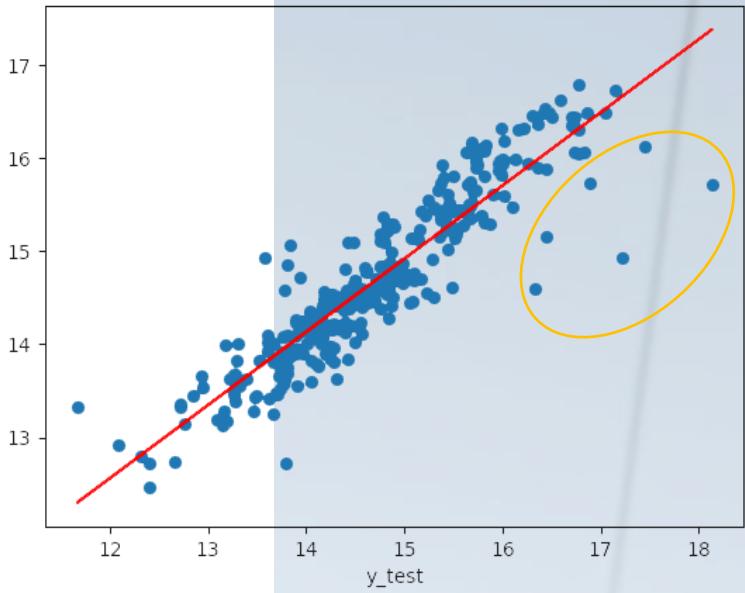
	model	r2	best_params	RMSE	medAE	MAE_%	time
0	linear_regression	0.843780	{}	0.433508	0.185241	0.018522	0.000000
1	elastic_net	0.845678	{"alpha": 0.001, "l1_ratio": 1.0}	0.437154	0.191777	0.018647	0.000998
2	SVR	0.831337	{"C": 3, "gamma": 0.1}	0.457432	0.200423	0.020440	0.027969
3	knn_regressor	0.722512	{"n_neighbors": 5}	0.572331	0.255127	0.026002	0.000000
4	random_forest	0.811258	{"min_samples_split": 10, "n_estimators": 50}	0.502792	0.238407	0.022800	0.172102
5	gradient_boost	0.843365	{"loss": "ls", "n_estimators": 100}	0.441247	0.218004	0.020276	0.112031

Ici, l'apport de la variable  
amène une amélioration  
conséquente du modèle.

	SANS ENERGIE STAR	AVEC ENERGIE STAR	DIFFÉRENCE EN VALEUR	DIFFÉRENCE EN %
R2	0,67	0,84	0,17	<b>25%</b>
RMSE	0,64	0,43	0,21	<b>-33%</b>
MEDAE	0,36	0,19	0,17	<b>-47%</b>

## Représentation des prédictions par rapport aux données test

y\_pred

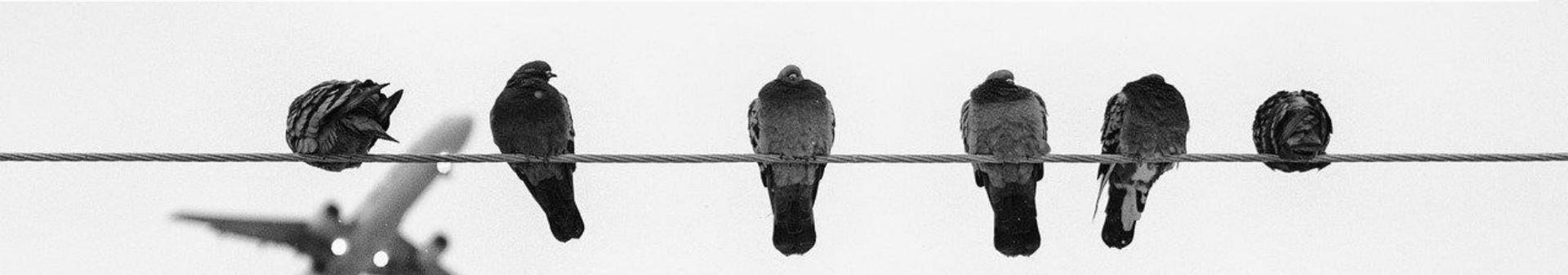


R2 0,84

On note quelques points qui mettent en difficulté le modèle mais autrement les prédictions sont bonnes.



# conclusion



Le modèle retenu est

**l'elastic net**

qui a très bien performé pour les deux variables.

Même si **l'Energie star** score est fastidieux, le fait de l'inclure dans le modèle **augmente largement la précision** des prédictions, donc il semble important d'en tenir compte.