

# **Anticipez les besoins en consommation électrique de bâtiments**

Soutenance du projet n°3 : Parcours « Ingénieur Machine Learning »

Luke Duthoit

# Plan de la présentation :

- Présentation de la problématique, de son interprétation et des pistes de recherche envisagées.
- Présentation du cleaning effectué, du feature engineering et de l'exploration.
- Présentation des différentes pistes de modélisation effectuées.
- Présentation du modèle final sélectionné ainsi que des améliorations effectuées.

# La problématique officielle du projet :

- *Data set* : bâtiments de la ville de Seattle (USA), sur 2015-2016.
- Paramètres : données de constructions, données énergétiques.
- Prédire la consommation électrique des bâtiments et leurs émissions de CO<sub>2</sub>.
- Évaluer la pertinence de la prise en compte de l'*EnergyStarSCORE*.

# Interprétation :

Prédire des grandeurs numériques continues : PB de régression.

- il faut jusqu'à 1 algorithme par étiquette, qui minimise l'erreur de prédiction et le sur-apprentissage ;
- les appliquer à *EnergyStarSCORE* et comparer l'erreur commise à celle sans prise en compte de ce paramètre.

# Pistes de recherche envisagées :

- Faire un premier tri parmi les variables afin de fournir un *dataset* réduit aux algorithmes (*i.e.* : trouver quelles variables ne sont pas pertinentes).
- Formater les données de telles sortes à pouvoir être lues par un algorithme de régression.
- Tester plusieurs algorithmes (via la validation croisée et l'optimisation de leurs hyper-paramètres).
- Retenir les meilleurs, et voir quelles sont leurs limites.

# Plan de la présentation :

- Présentation de la problématique, de son interprétation et des pistes de recherche envisagées.
- Présentation du *cleaning* effectué, du *feature engineering* et de l'exploration.
- Présentation des différentes pistes de modélisation effectuées.
- Présentation du modèle final sélectionné ainsi que des améliorations effectuées.

# *Cleaning effectué :* **fusionner les bases de données 2015 et 2016**

- Deux *data set* sur deux années successives : autant les regrouper en une seule base de donnée pour en faciliter l'exploitation.
- ! : mêmes informations ⇒ même format.

	TotalGHGEmissions	GHGEmissionsIntensity
count	3367.000000	3367.000000
mean	119.723971	1.175916
std	538.832227	1.821452
min	-0.800000	-0.020000
25%	9.495000	0.210000
50%	33.920000	0.610000
75%	93.940000	1.370000
max	16870.980000	34.090000

2016

	GHGEmissions(MetricTonsCO2e)	GHGEmissionsIntensity(kgCO2e/ft2)
count	3330.000000	3330.000000
mean	110.094102	0.985339
std	409.450179	1.637172
min	0.000000	0.000000
25%	9.265000	0.080000
50%	32.740000	0.460000
75%	88.642500	1.180000
max	11824.890000	31.380000

2015

# Réduction de dimension : quelles étiquettes conserver ?

- Au total, 13 étiquettes possibles, mais de très fortes corrélations linéaires entre certaines :
  - exprimées dans différentes unités (kBtu  $\propto$  kWh  $\propto$  therms);
  - liées par relations physique (*Site[...]* et *Source[...]*).

- On s'est donc concentré sur celles avec un coeff.<0,9.

- Encore 7 étiquettes, dont :
  - *SiteEnergyUse(kBtu)* ;
  - *GHGEmissions(Metric[...])*.

**NB : Par manque de temps, seules ces 2 étiquettes auront fait l'objet de prédiction par ML.**

	SiteEUI(kBtu/sf)	EUIWN(kBtu/sf)	SourceEUI(kBtu/sf)	EUIWN(kBtu/sf)	EnergyUse(kBtu)	EnergyUseWN(kBtu)	SteamUse(kBtu)	Electricity(kWh)	Electricity(kBtu)	NaturalGas(therms)	NaturalGas(kBtu)	GHGEmissions(MetricTonsCO2e)	GHGEmissionsIntensity(kgCO2e/ft2)
SiteEUI(kBtu/sf)	1	0.99	0.95	0.95	0.35	0.42	0.12	0.33	0.33	0.29	0.29	0.31	0.73
EUIWN(kBtu/sf)	0.99	1	0.93	0.94	0.32	0.41	0.11	0.31	0.31	0.29	0.29	0.31	0.75
SourceEUI(kBtu/sf)	0.95	0.93	1	1	0.35	0.42	0.098	0.38	0.38	0.2	0.2	0.26	0.52
EUIWN(kBtu/sf)	0.95	0.94	1	1	0.33	0.41	0.089	0.36	0.36	0.2	0.2	0.25	0.53
SiteEnergyUse(kBtu)	0.35	0.32	0.35	0.33	1	0.8	0.59	0.95	0.95	0.55	0.55	0.87	0.31
SiteEnergyUseWN(kBtu)	0.42	0.41	0.42	0.41	0.8	1	0.52	0.7	0.7	0.69	0.69	0.87	0.39
SteamUse(kBtu)	0.12	0.11	0.098	0.089	0.59	0.52	1	0.5	0.5	0.036	0.036	0.71	0.2
Electricity(kWh)	0.33	0.31	0.38	0.36	0.95	0.7	0.5	1	1	0.33	0.33	0.69	0.18
Electricity(kBtu)	0.33	0.31	0.38	0.36	0.95	0.7	0.5	1	1	0.33	0.33	0.69	0.18
NaturalGas(therms)	0.29	0.29	0.2	0.2	0.55	0.69	0.036	0.33	0.33	1	1	0.71	0.47
NaturalGas(kBtu)	0.29	0.29	0.2	0.2	0.55	0.69	0.036	0.33	0.33	1	1	0.71	0.47
GHGEmissions(MetricTonsCO2e)	0.31	0.31	0.26	0.25	0.87	0.87	0.71	0.69	0.69	0.71	0.71	1	0.45
GHGEmissionsIntensity(kgCO2e/ft2)	0.73	0.75	0.52	0.53	0.31	0.39	0.2	0.18	0.18	0.47	0.47	0.45	1

# Réduction de dimension : quelles caractéristiques conserver ?

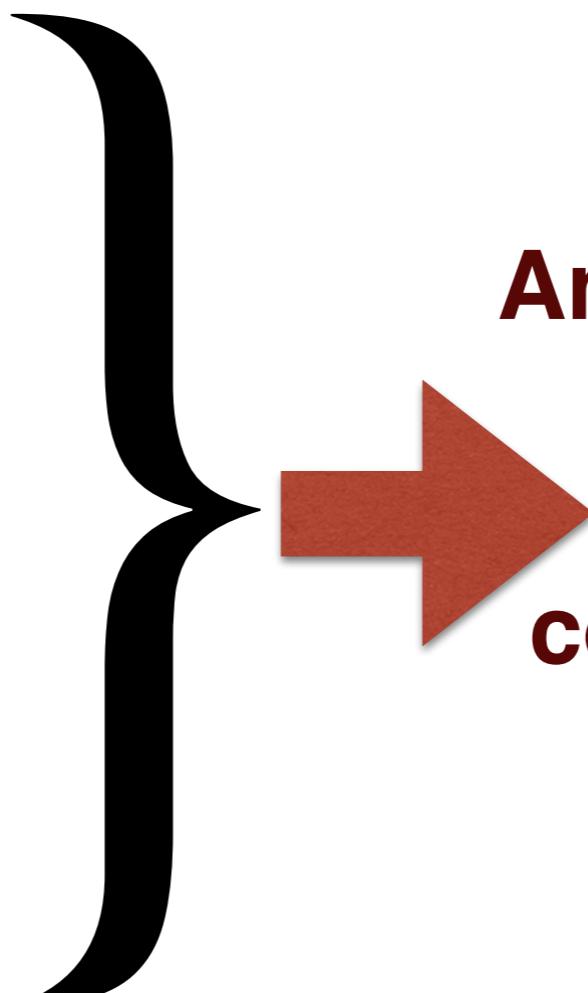
D'après l'interprétation de l'énoncé, peuvent être mis en entrée de l'algorithme

- des paramètres quantitatifs tels que :

- surface ;
- nbr. étages ;
- nbr. immeubles ;
- année de construction ;
- localisation GPS ;

- des paramètres qualitatifs tels que :

- type d'usage ;
- mention de travaux récents ;
- quartier.



**Analyse préliminaire pour anticiper la pertinence de conserver tous ces paramètres.**

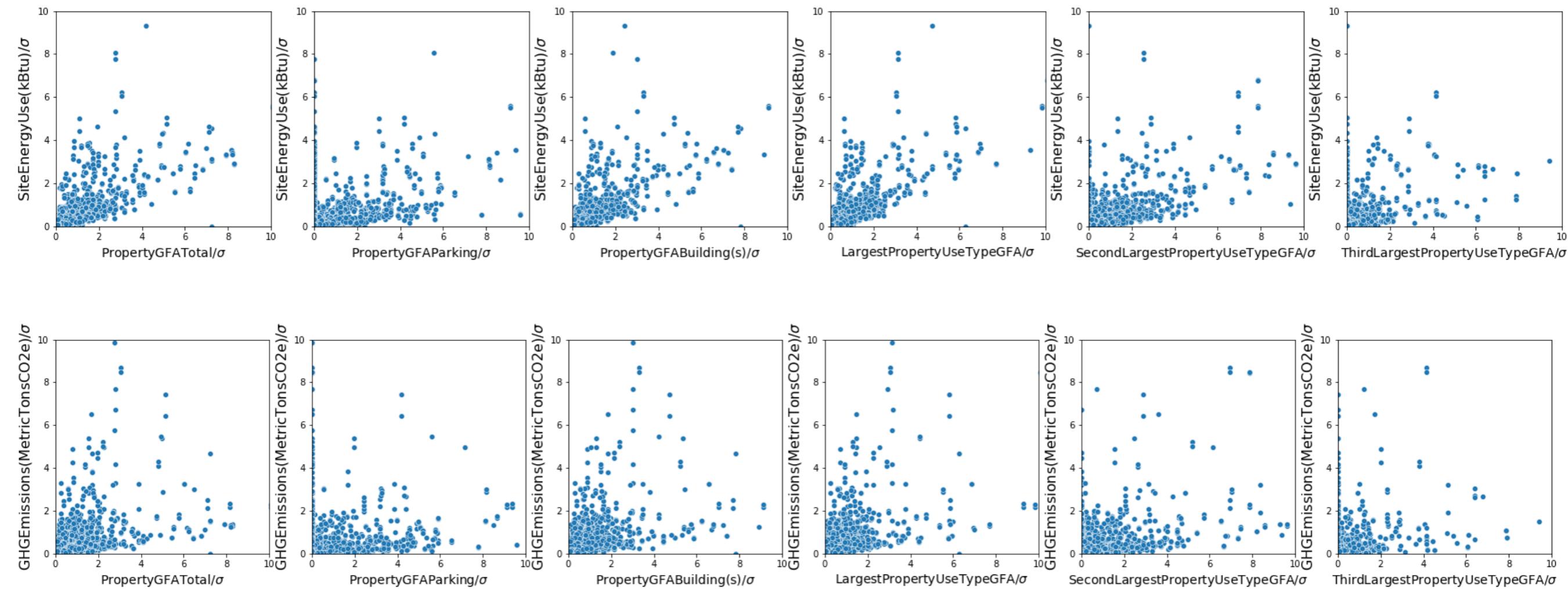
# Réduction de dimension : quelles caractéristiques conserver ?

Au final, 14 paramètres sont retenus comme *features* :

- 8 paramètres quantitatifs :
  - 6 surfaciques (*PropertyGFATotal* ; *PropertyGFAParking* ; *PropertyGFABuilding(s)* ; *LargestPropertyUseTypeGFA* ; *2ndLargestPropertyUseTypeGFA* ; *3rdLargestPropertyUseTypeGFA* ) ;
  - 2 localisation GPS ( *Latitude*, *Longitude* );
- 6 paramètres qualitatifs :
  - 5 type d'usage ( *BuildingType* ; *PrimaryPropertyType* ; *LargestPropertyUseType* ; *2ndLargestPropertyUseTypeGFA* ; *3rdLargestPropertyUseType* );
  - 1 quartier (*Neighborhood*). 10

# Ex. : Paramètres surfaciiques : *PropertyGFA[...] et [...]UseTypeGFA*

Analyse bi-variée : certains paramètres semblent avoir une influence sur la distribution des certaines étiquettes.



# Ex. : Paramètres surfaciques : *PropertyGFA[...] et [...]UseTypeGFA*

- Assez bonnes corrélations linéaire, du fait de relations hiérarchiques :

- $Total = Parking + Building(s)$  ;

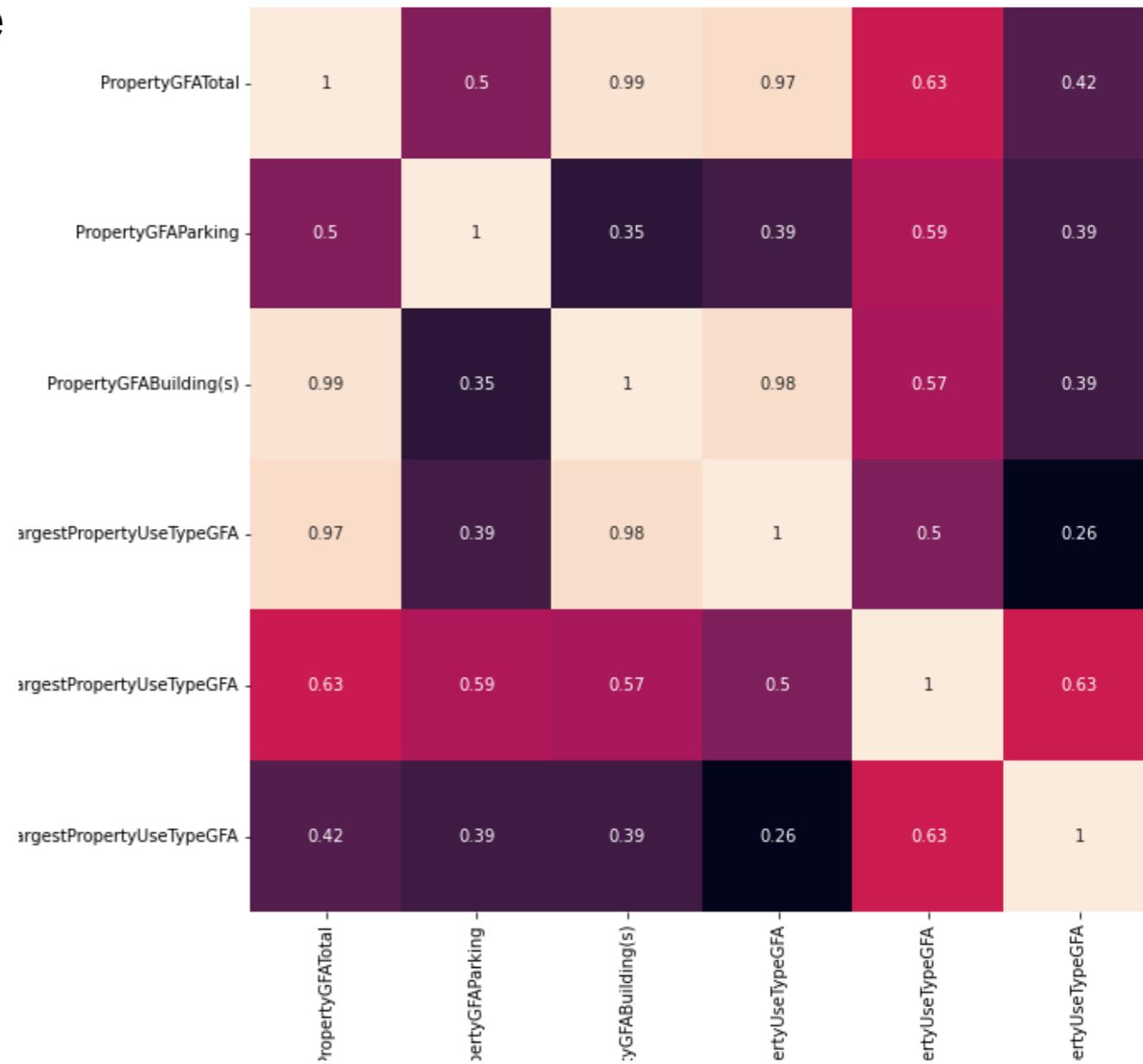
- $Total \geq First \geq Second \geq Third$ .

- ⚠ On les conserve **tous** (au cas où ces relations apportent de la robustesse aux algorithmes).

- 💡 On se sert de ces relations pour :

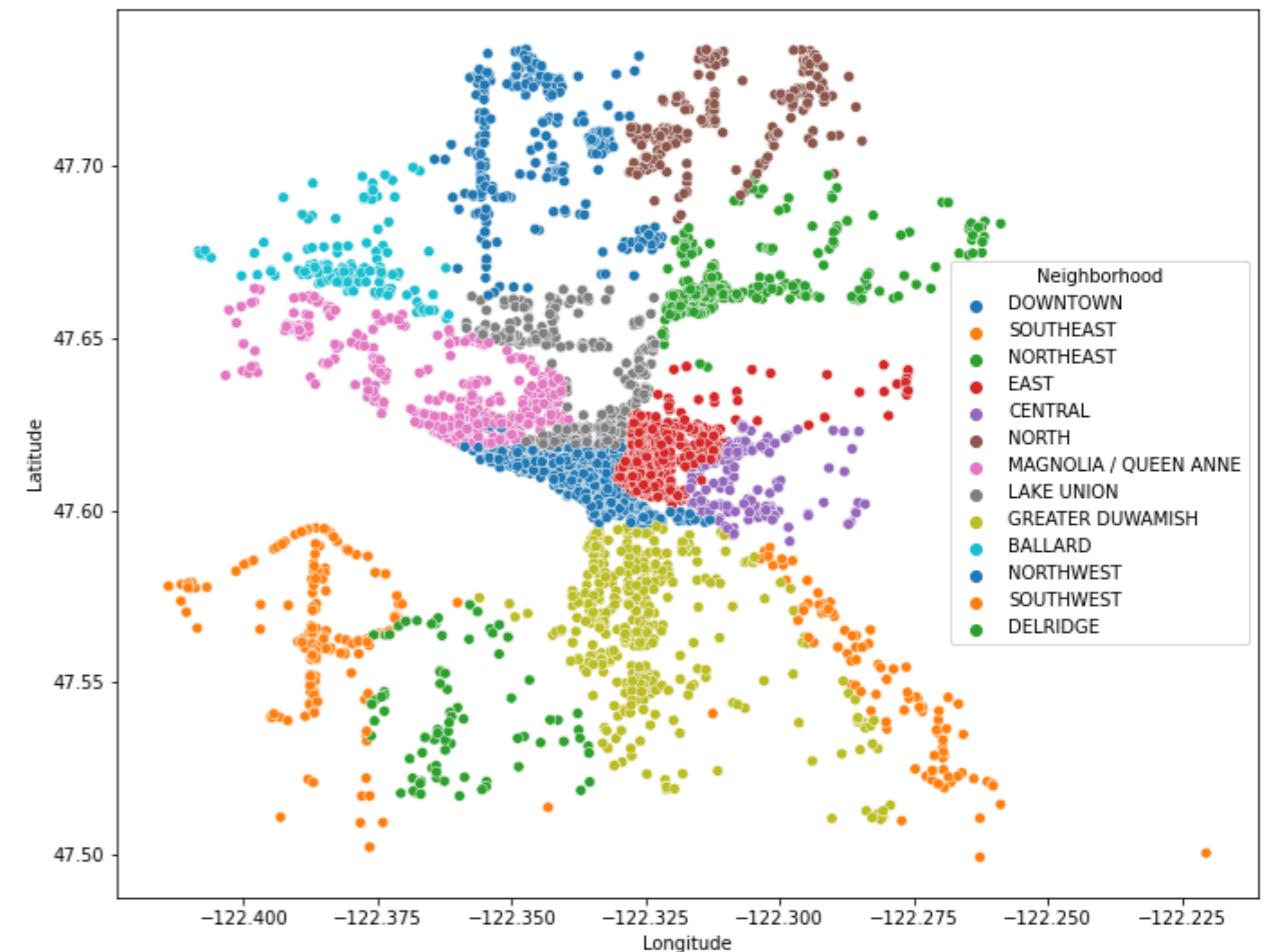
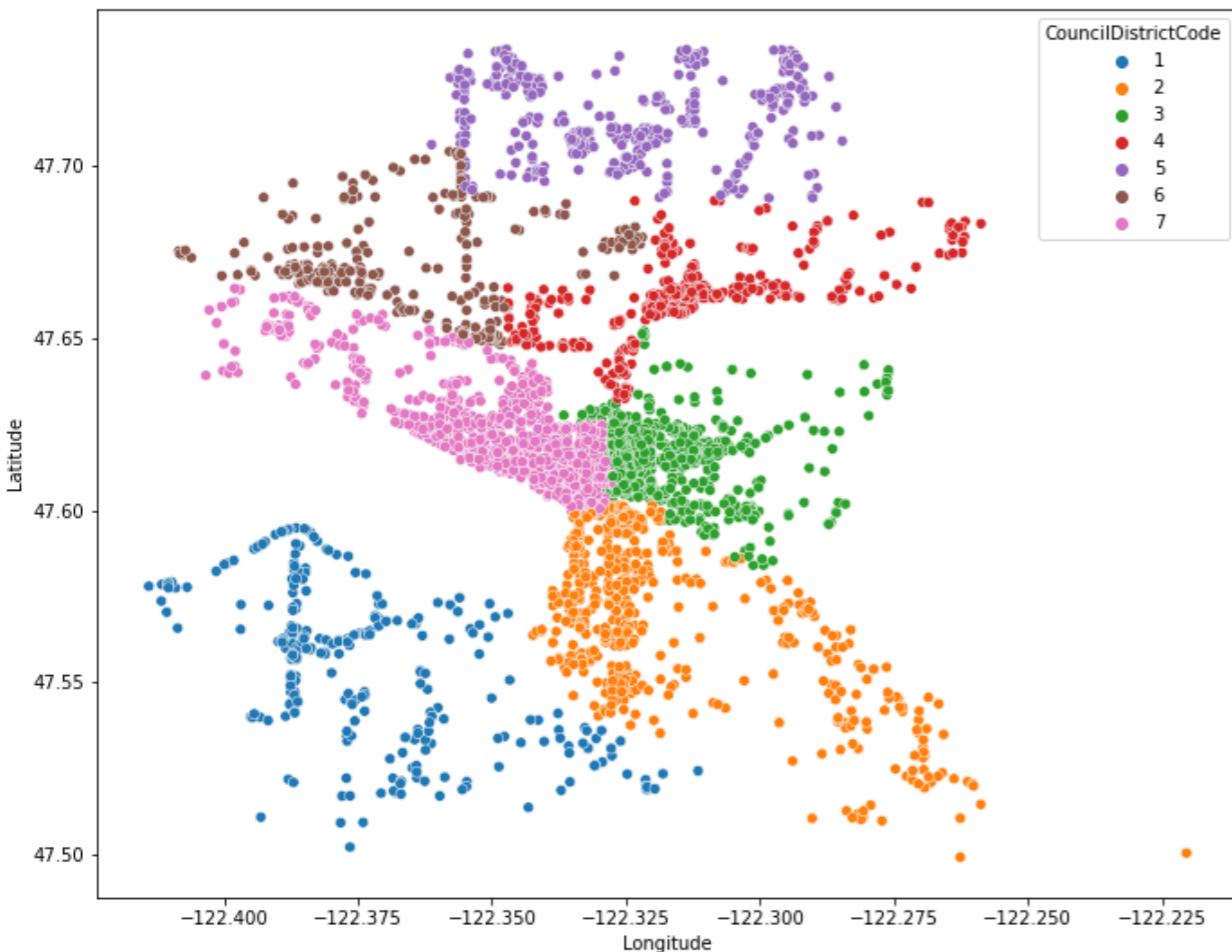
- supprimer valeurs aberrantes ;

- imputer un grand nombre de NaN (remplacées par 0 pour *Second*/*Third*[...])



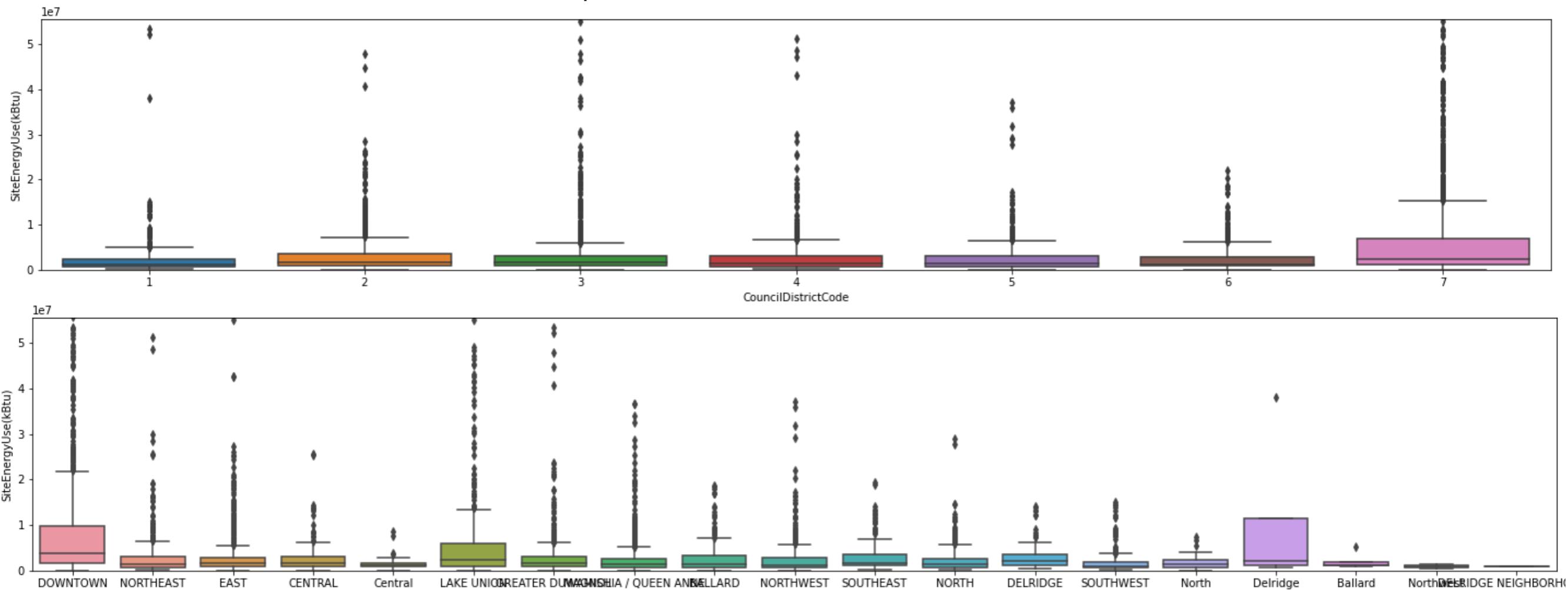
# Ex. : Position dans l'espace.

- Quatre paramètres apporte cette information :
  - les coordonnées GPS ;
  - le découpage administratif (avec deux niveaux de détail de description).



# Ex. : Position dans l'espace.

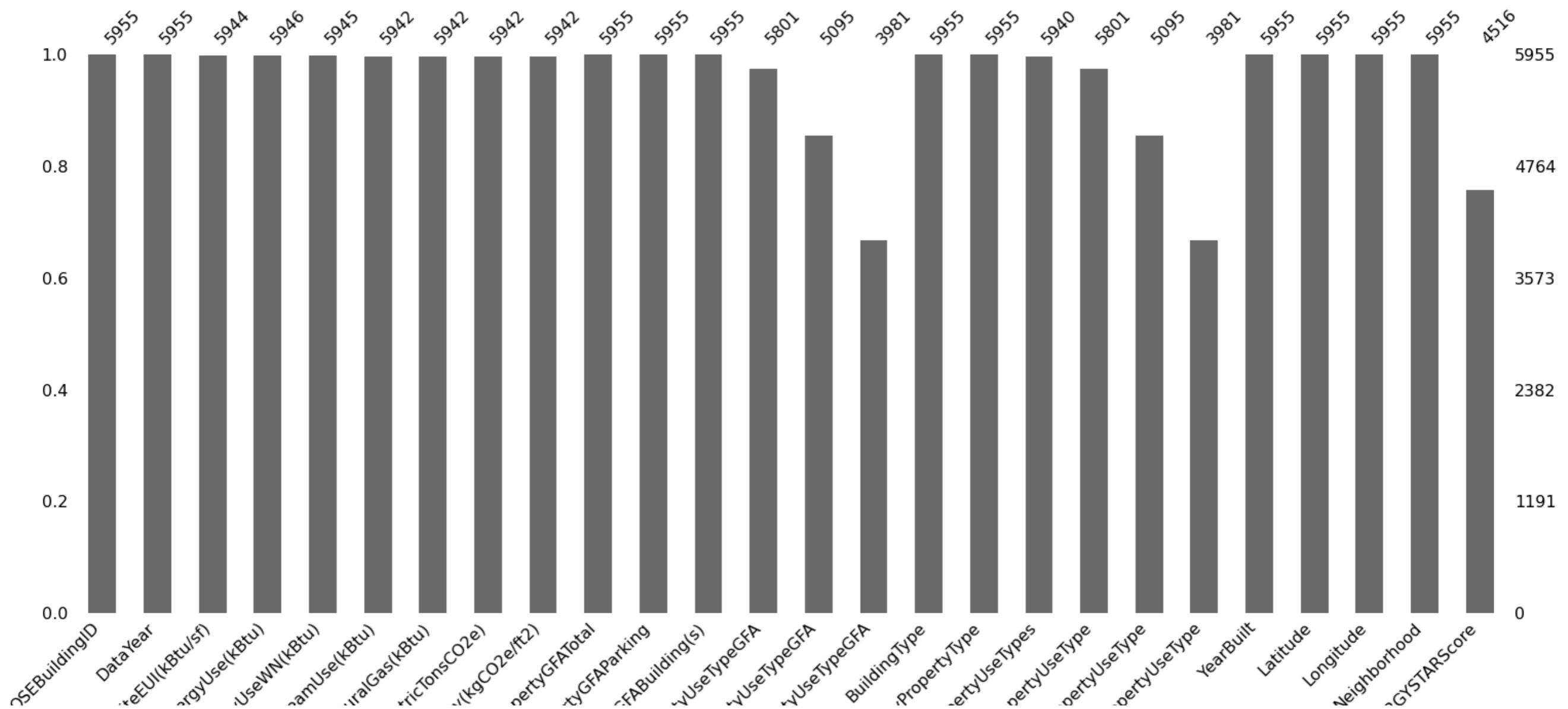
- Analyse bi-variée : Seul *Neighborhood* permet de faire quelques différences entre distributions particularisées.  
⇒ On conserve seulement ce paramètre, et on délaisse *CouncilDistrictCode*.



- ⚠ On conserve également *Latitude* et *Longitude* pour se donner une alternative qualitative à *Neighborhood*.

# Réduire le nombre de NaN : imputation sous contrôle...

- On utilise les relations hiérarchiques (surfaces, type d'usage) pour remplacer des milliers de NaN par des zéros (absence de 2nd et/ou 3ème usage du bâtiment).



# ... avant la réduction de dimension.

- Pas d'autres moyens de faire de l'imputation « prudente » sans corrompre le jeu de données : on supprime les lignes du data set contenant les NaN.
- Conséquences pour les étiquettes et caractéristiques :
  - % de modification des propriétés statistiques des distributions (étiquettes et *features* quantitatives) ;

```
SiteEnergyUse(kBtu) : Delta(moyenne) = 6.20% ; Delta(écart-type) = -2.34% ; Delta(médiane) = 6.40%
GHGEmissions(MetricTonsCO2e) : Delta(moyenne) = 7.21% ; Delta(écart-type) = 14.34% ; Delta(médiane) = 2.70%
```

```
PropertyGFATotal : Delta(moyenne) = 5.16% ; Delta(écart-type) = -7.68% ; Delta(médiane) = 6.65%
PropertyGFAParking : Delta(moyenne) = 16.68% ; Delta(écart-type) = 3.02% ; Médiane = 0
```
  - les catégories disparues et % d'éléments disparus par catégories (*features* qualitatives).

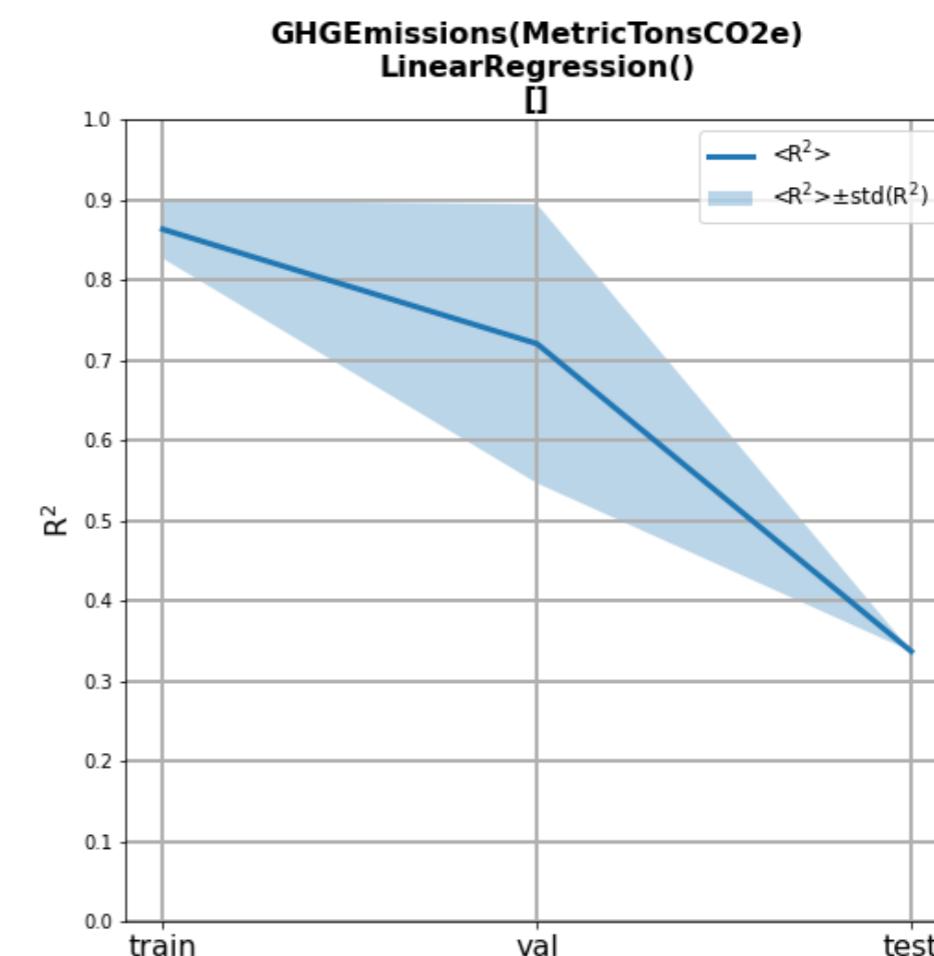
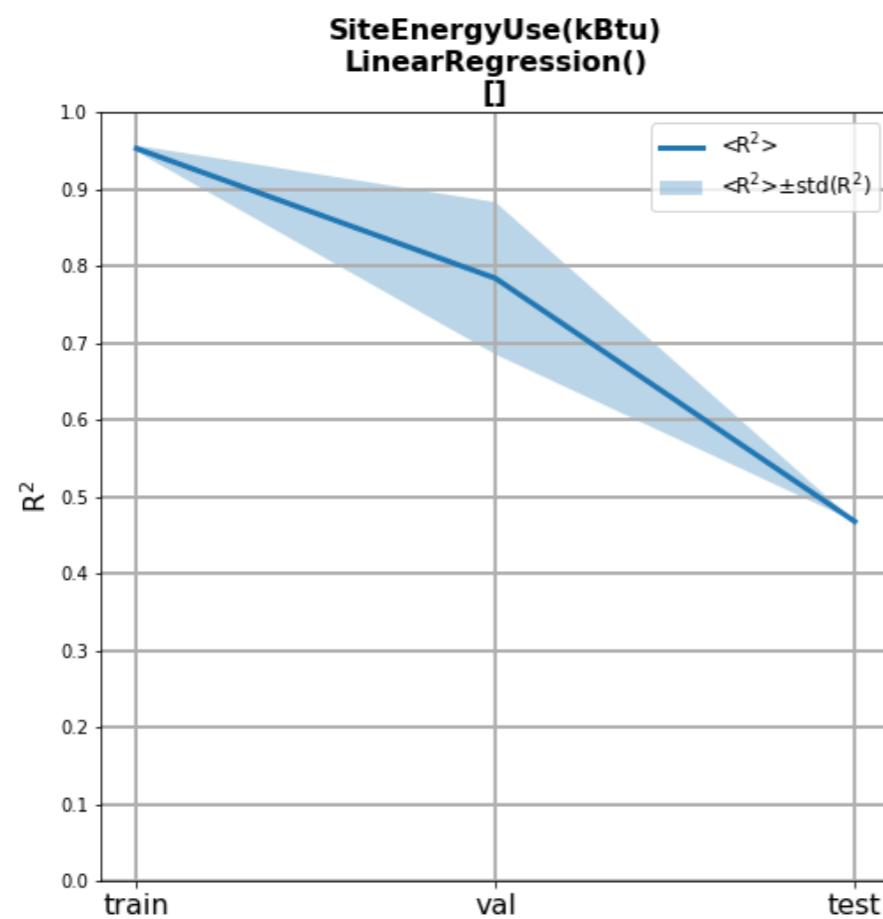
```
BuildingType : Categories supprimees : 0.00% ; <Elemnts en - / categorie restante> : 29.66% +- 9.57%
PrimaryPropertyType : Categories supprimees : 3.70% ; <Elemnts en - / categorie restante> : 33.18% +- 17.23%
```
  - On a considéré ces pertes comme acceptables.

# Séparation du *data set* en jeux d'entraînement et de test.

- ! Certains bâtiments sont présents sur 2 années (mêmes caractéristiques)  $\Rightarrow$  les 2 éléments correspondant doivent rester dans le même jeu.
- À partir de (*OSEBuildingID* ; *DataYear*) on sépare les bâtiments selon s'ils sont sur une seule année ou deux.
- On génère les jeux d'entraînement [*train set*] et de test [*test set*] à partir de ces deux précédentes catégories.

# Features engineering : numériser les features qualitatives.

- **Features qualitatives** : on crée autant de nouveaux paramètres binaires qu'il y'a catégories ≠ via le *One Hot Encoding* (6 qualitatives → 190 quantitatives !!!)
- Permet un 1er modèle simple de référence : une régression linéaire classique sur les features numériques initiales et nos 190 nouveaux paramètres.
- Sur-apprentissage pour chacune de nos étiquettes.

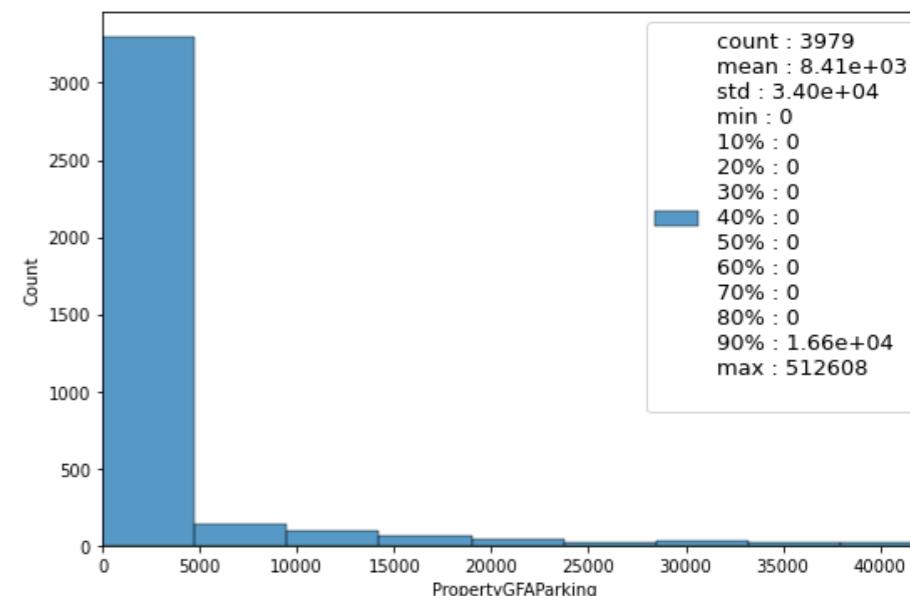
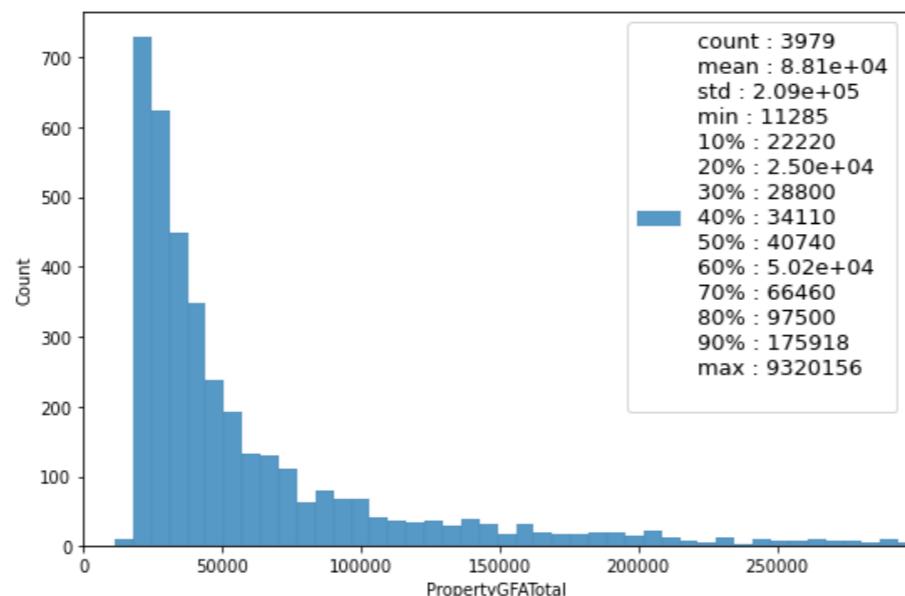


# Features engineering : adimensionner les features quantitatives et les targets.

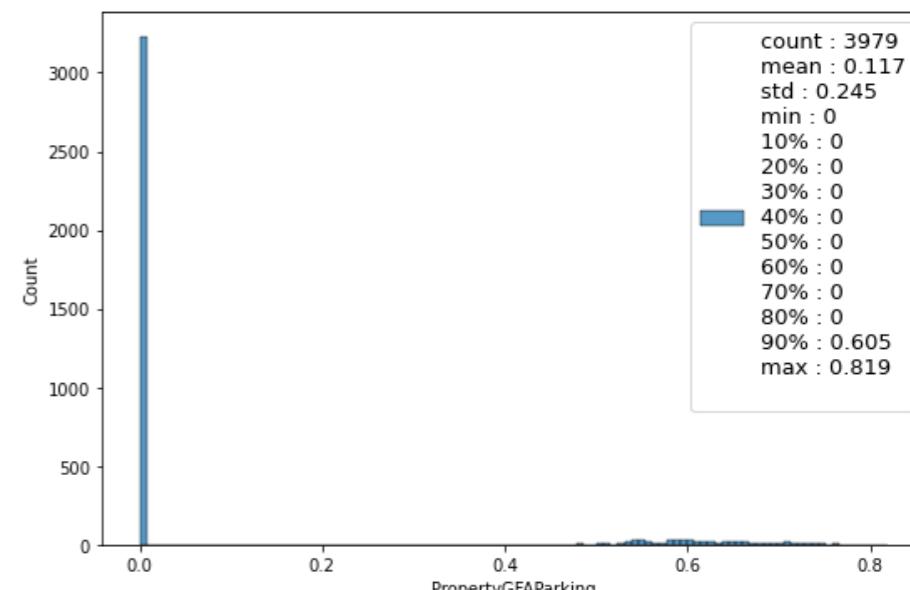
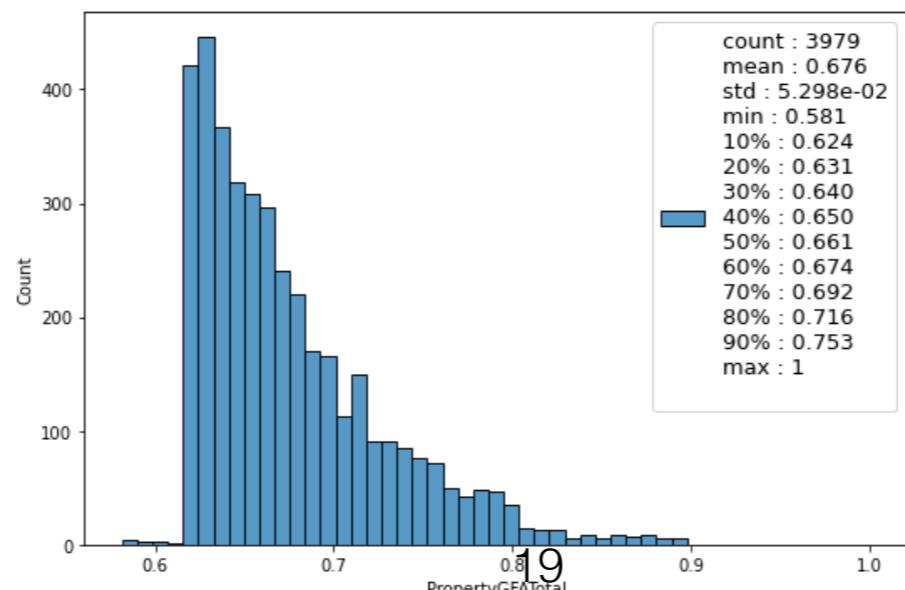
- 💡 **Features quantitatives** : Doivent être ramenées à des valeurs de l'ordre de 0 et/ou 1 pour être comparées aux nouvelles *features* binaires.

- Paramètres surfaciques :**

passage par  $\log(x+1)$   
puis divisé par  
 $\max(\text{PropertyGFATotal})$  afin de conserver  
hiérarchie des surfaces.



- Coordonnées GPS :**  
StandardScaler().



- Targets** : passage par  $\log(x+1)$ .

# Plan de la présentation :

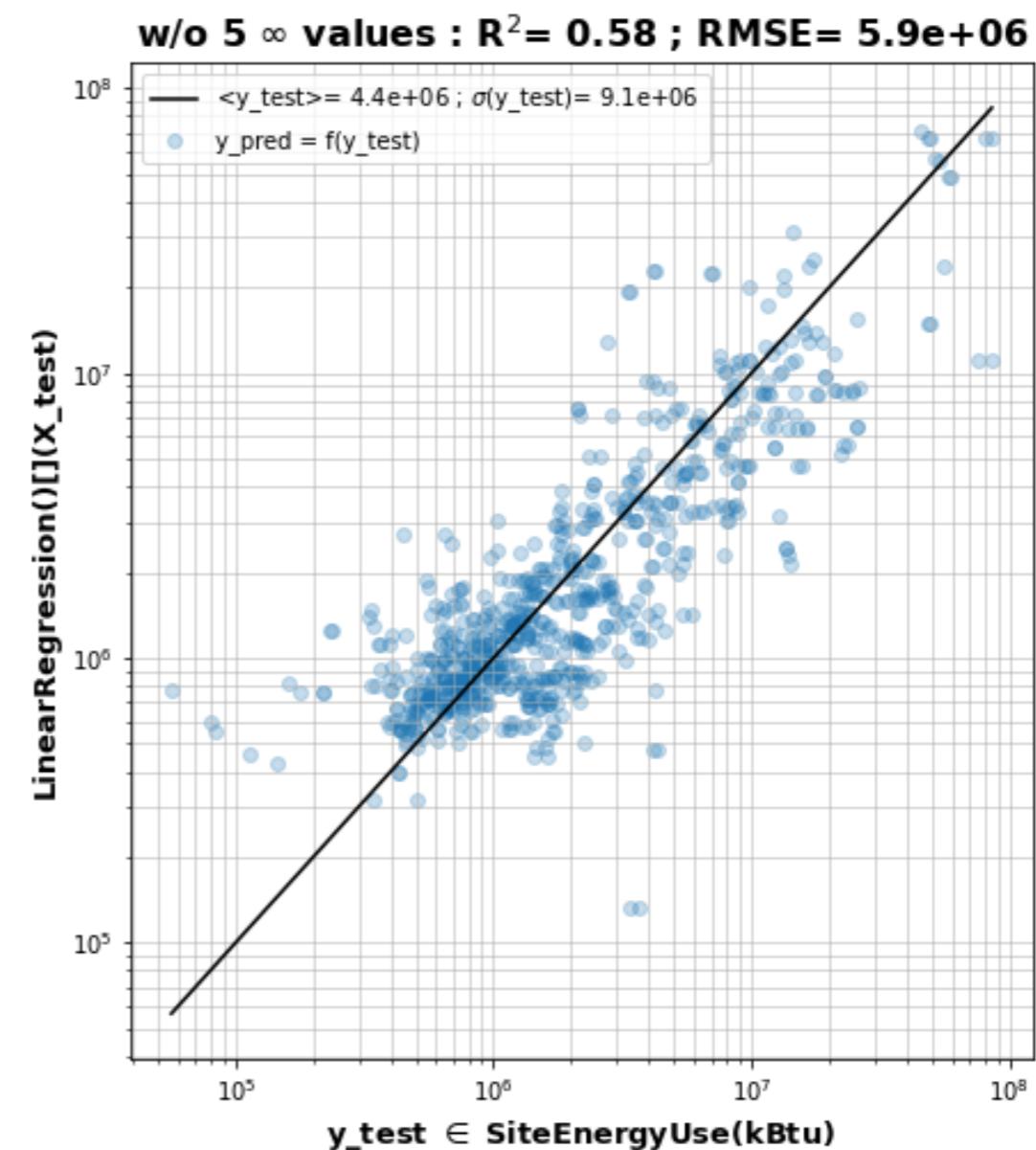
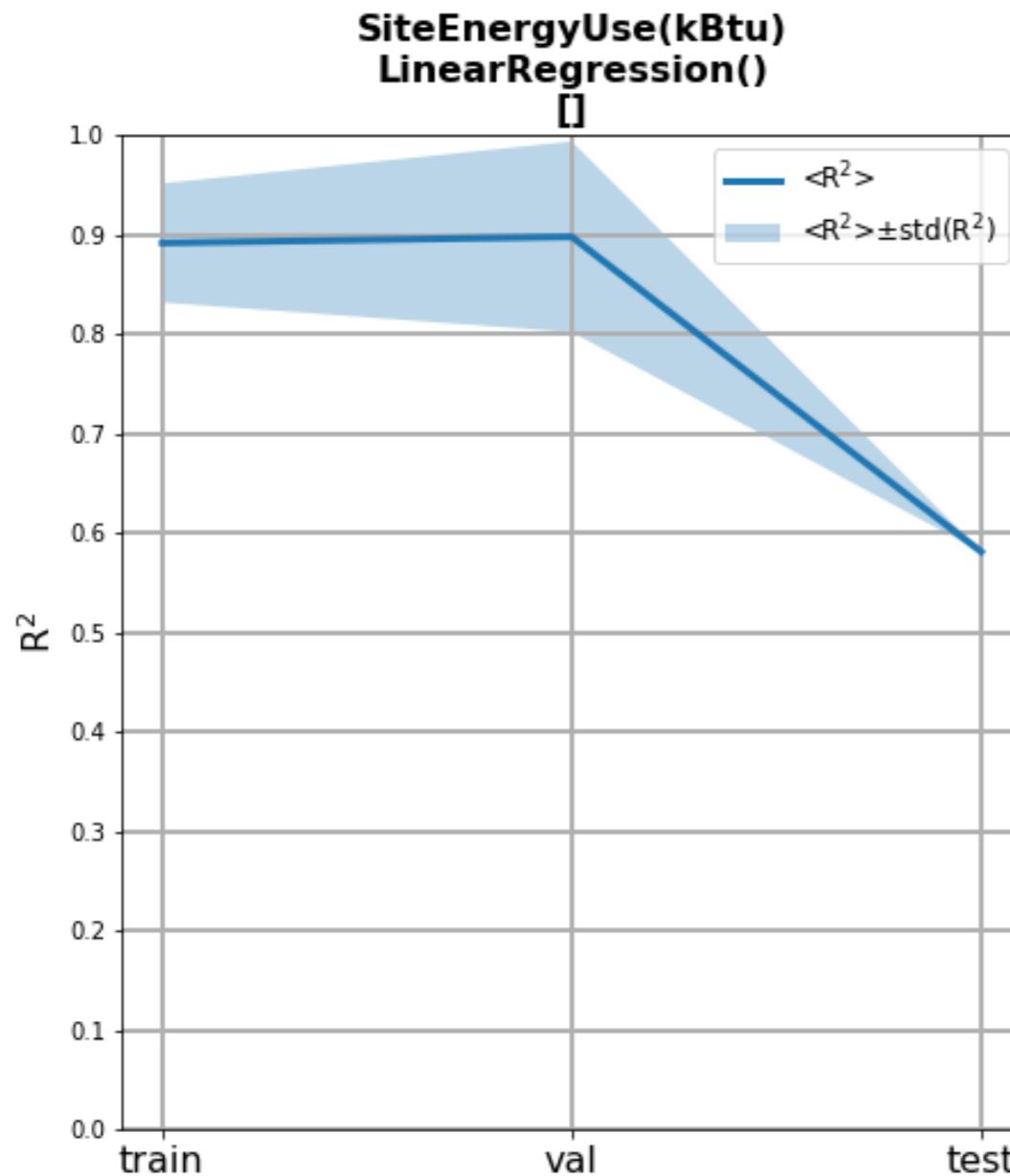
- Présentation de la problématique, de son interprétation et des pistes de recherche envisagées.
- Présentation du cleaning effectué, du feature engineering et de l'exploration.
- Présentation des différentes pistes de modélisation effectuées.
- Présentation du modèle final sélectionné ainsi que des améliorations effectuées.

# Méthodologie :

- Choix d'un algorithme/d'une série d'algorithme similaires.
- Optimisation des hyper-paramètres avec *GridSearchCV()* sur *train set* avec étiquettes adimensionnées.
- Validation croisée et calcul du score sur *test set* avec étiquettes originales.

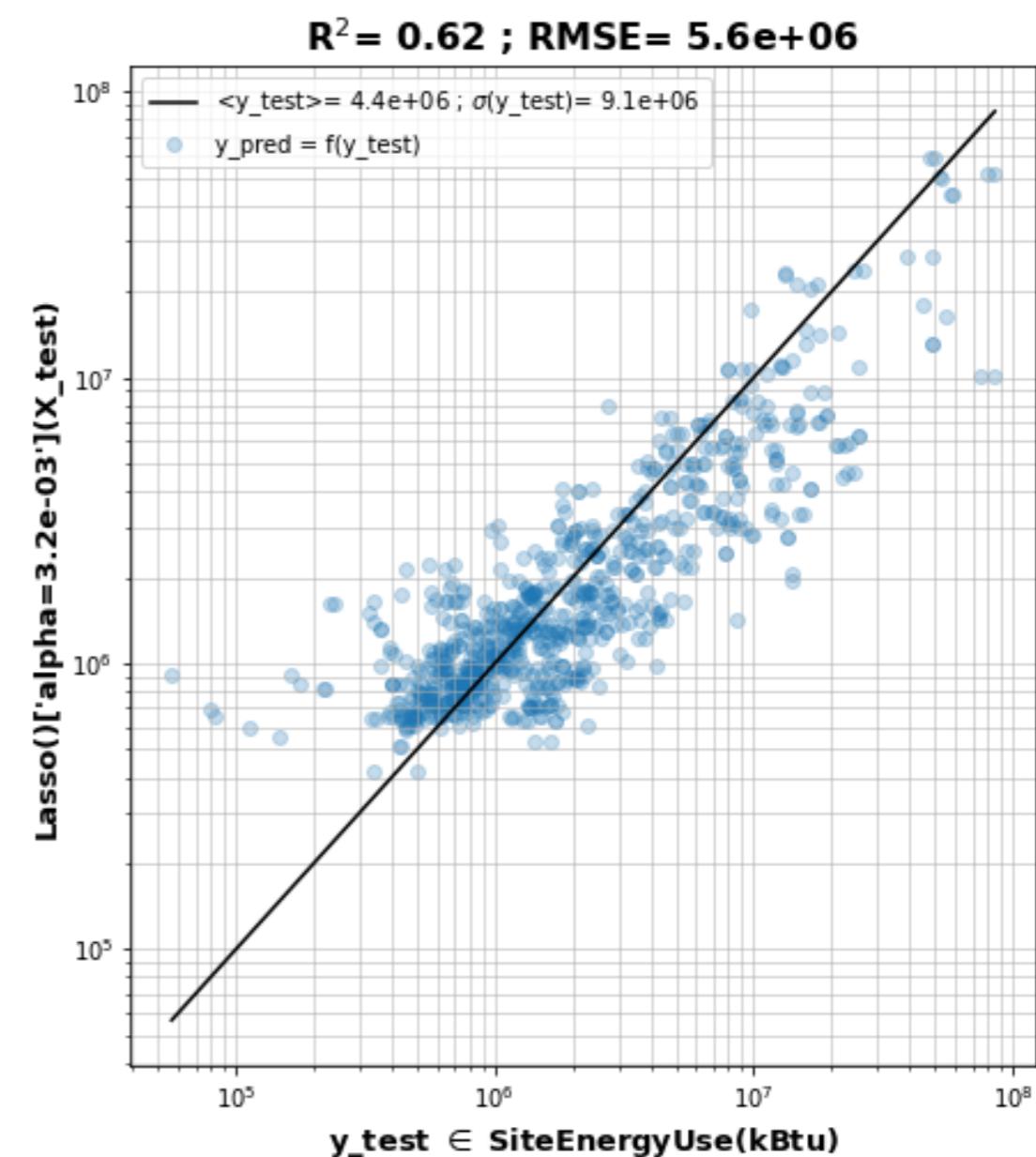
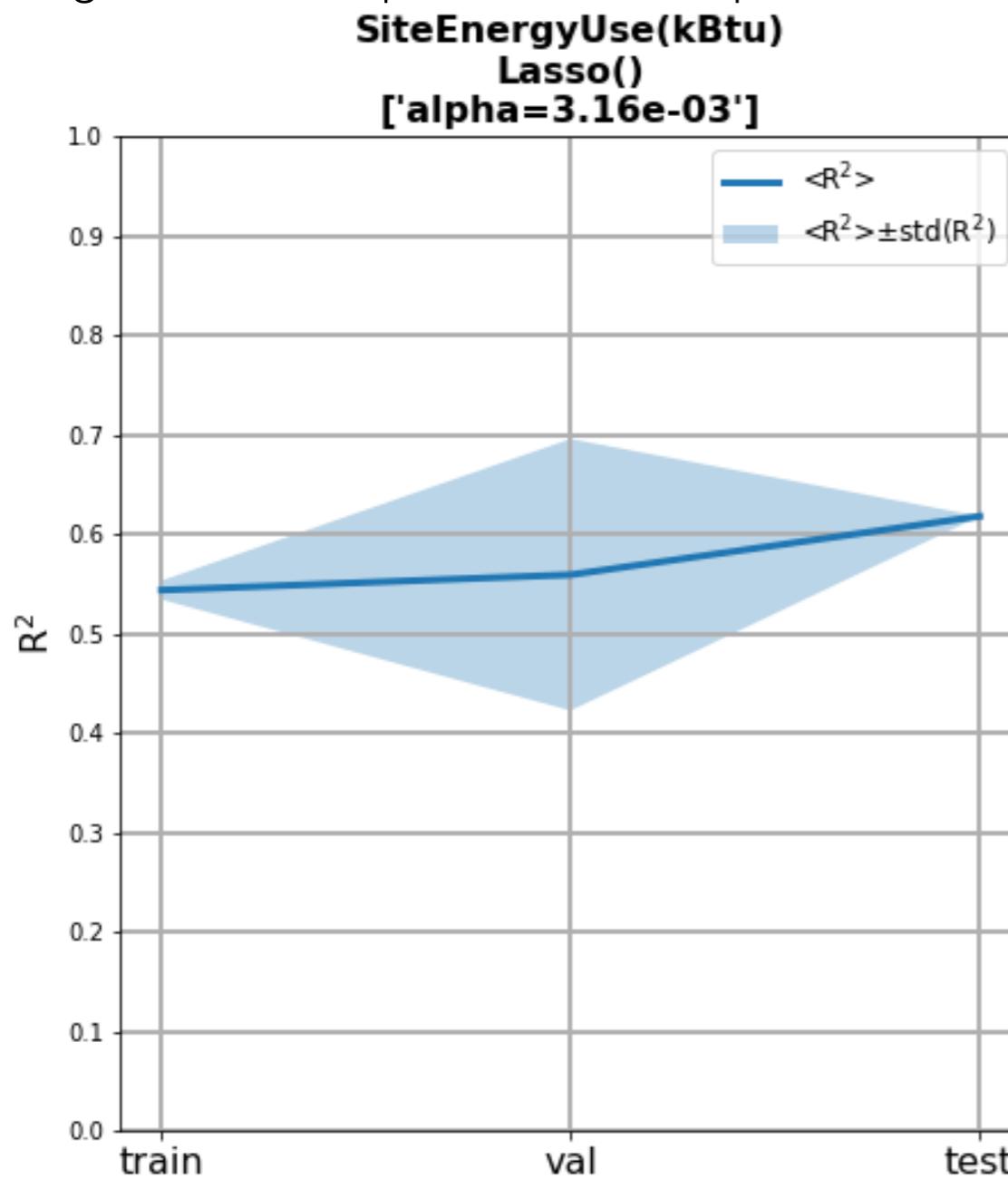
# 1ère piste : régression linéaire sur données adimensionnées.

- Poignée de prédictions sont si élevées que la reconversion en grandeur physique donnent des valeurs  $\infty$ .
- Hormis celles-ci : meilleures performances avec données adimensionnées, mais encore trop de sur-apprentissage :  $R^2(\text{train}) \sim R^2(\text{val}) \gg R^2(\text{test})$ .



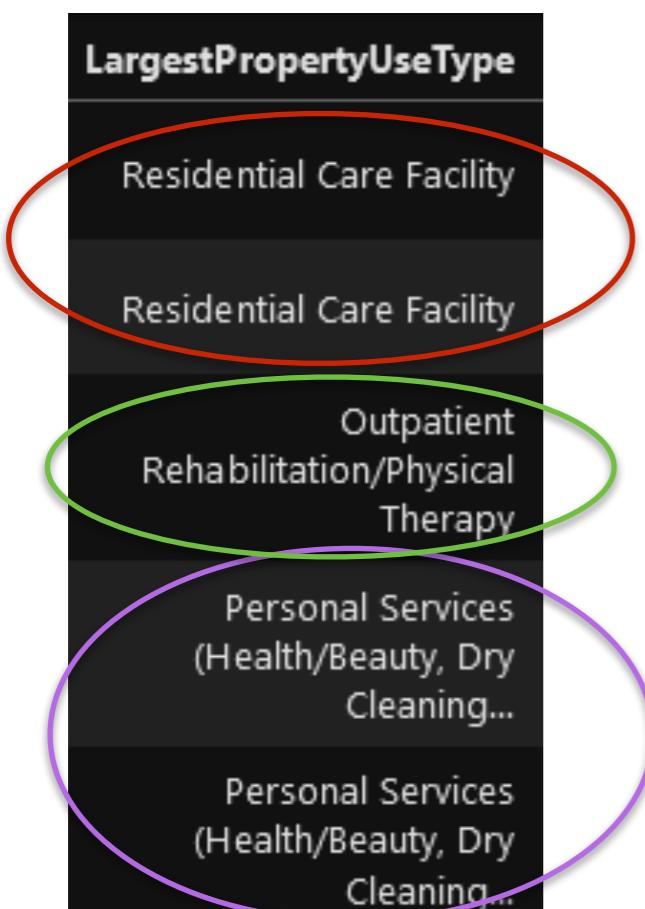
# 1ère piste : régression linéaire sur données adimensionnées.

- Singularités dues à la hauteur des coefficients, car la régularisation règle ce problème (en + de meilleures performances).
- Ci-dessous : exemple de meilleures performances pour la régression Lasso avec paramètre de régularisation  $\alpha$  préalablement optimisé



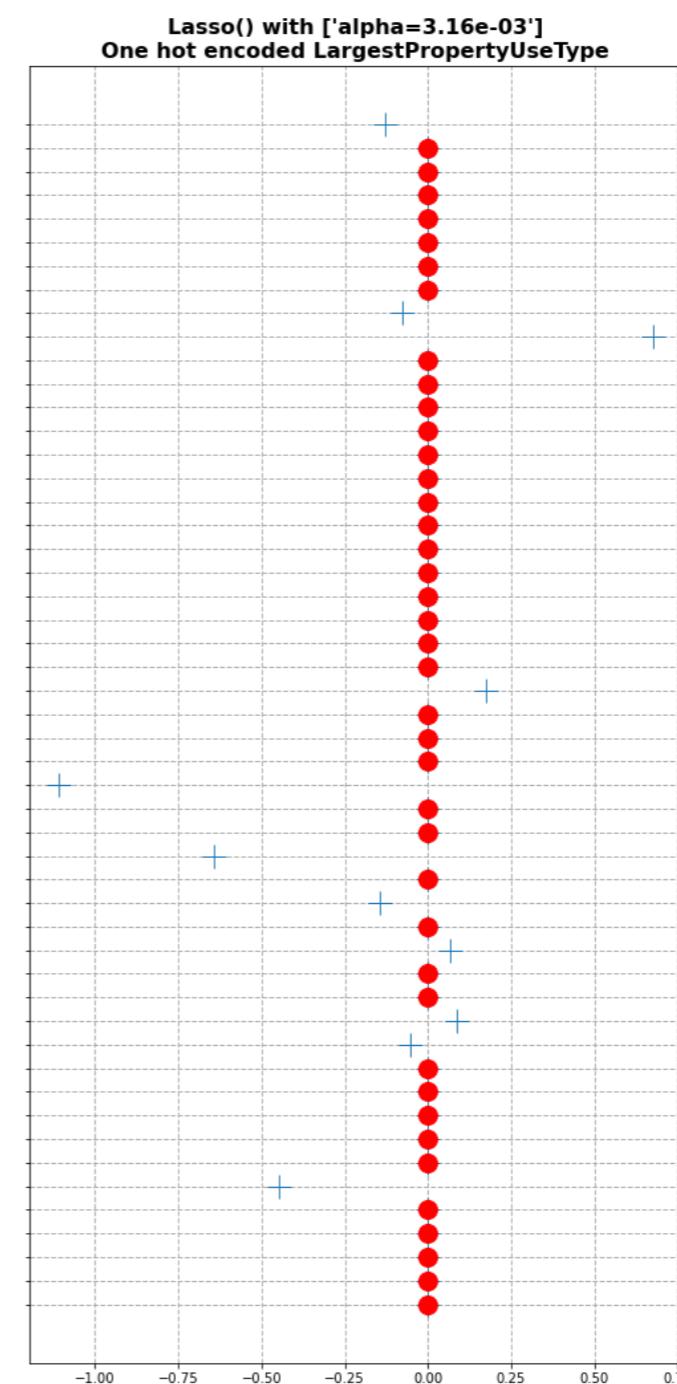
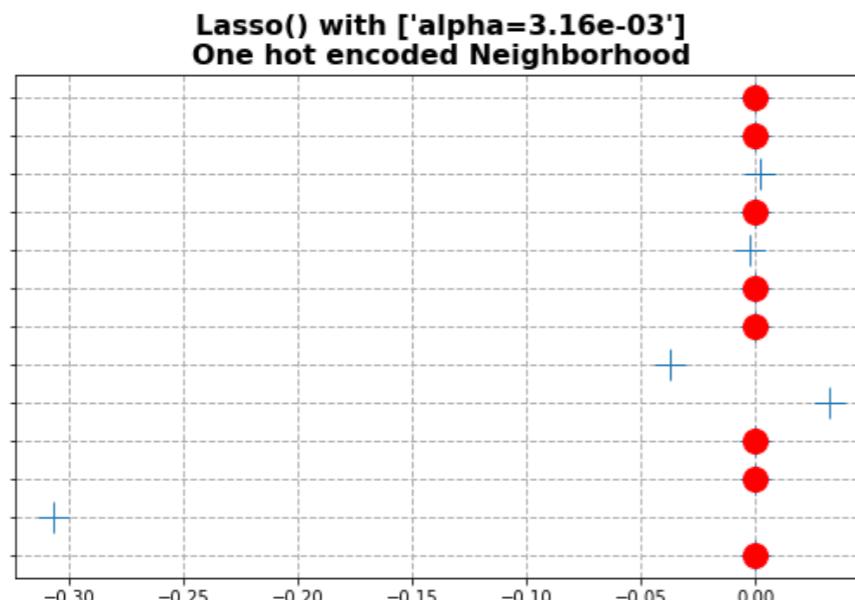
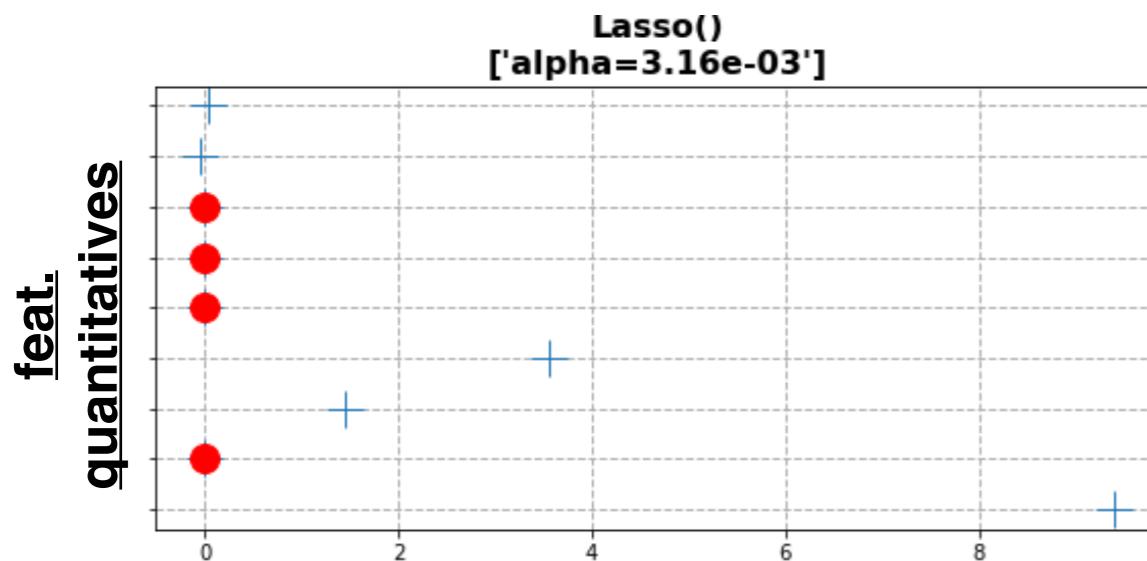
# 1ère piste : régression linéaire sur données adimensionnées.

- PB : difficultés à expliquer pourquoi ces éléments conduisent à des hautes valeurs:
  - *feat.* quantitatives : valeurs dans la moitié basse (GFA) ou moyenne (GPS) des distributions ;
  - *feat.* qualitatives : aucune catégorie associée à un coeff. linéaire très différent des autres.



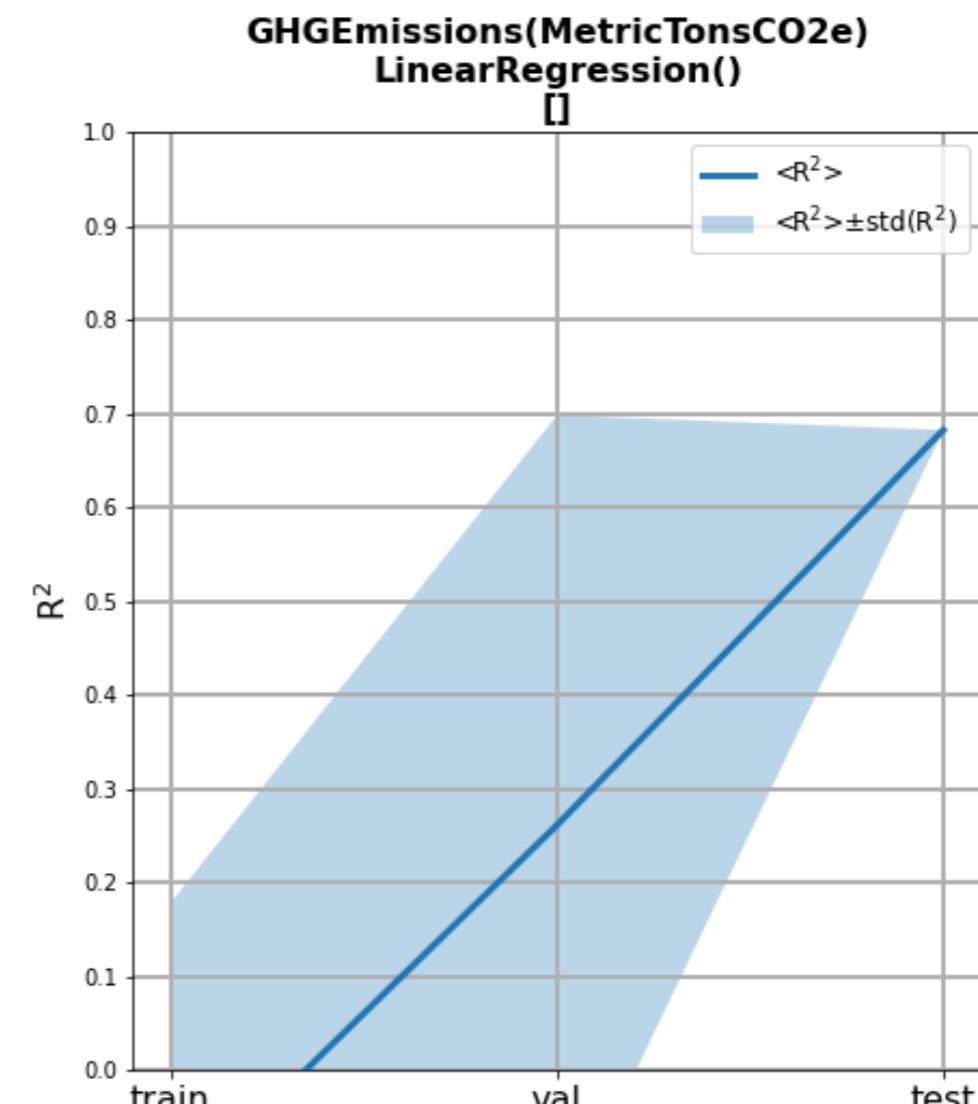
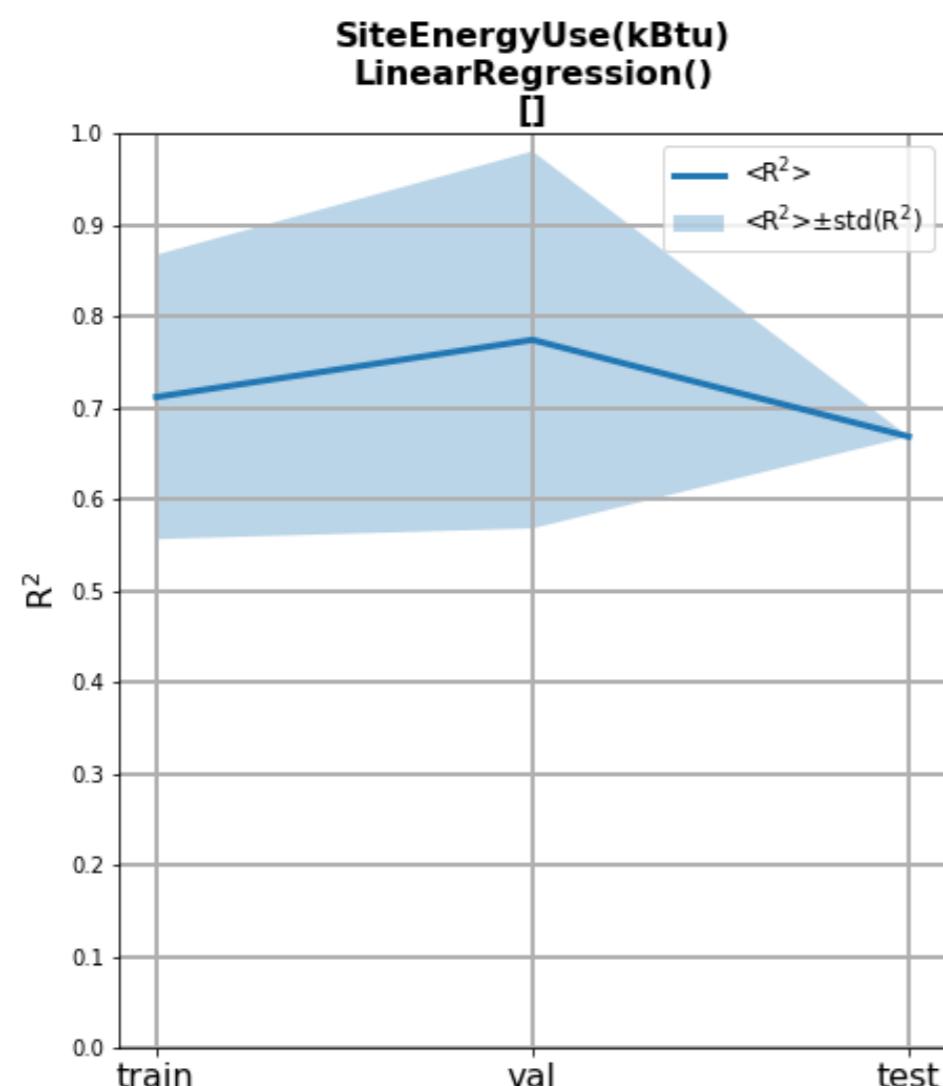
# 2nde piste : régression linéaire sur données adimensionnées, - de *features*.

- Coefficients linéaires de la régression Lasso souvent nuls (en rouge sur les graphes ci-dessous)  $\Rightarrow$  on essaie la régression linéaire sans les *features* correspondants.



# 2nde piste : régression linéaire sur données adimensionnées, - de *features*.

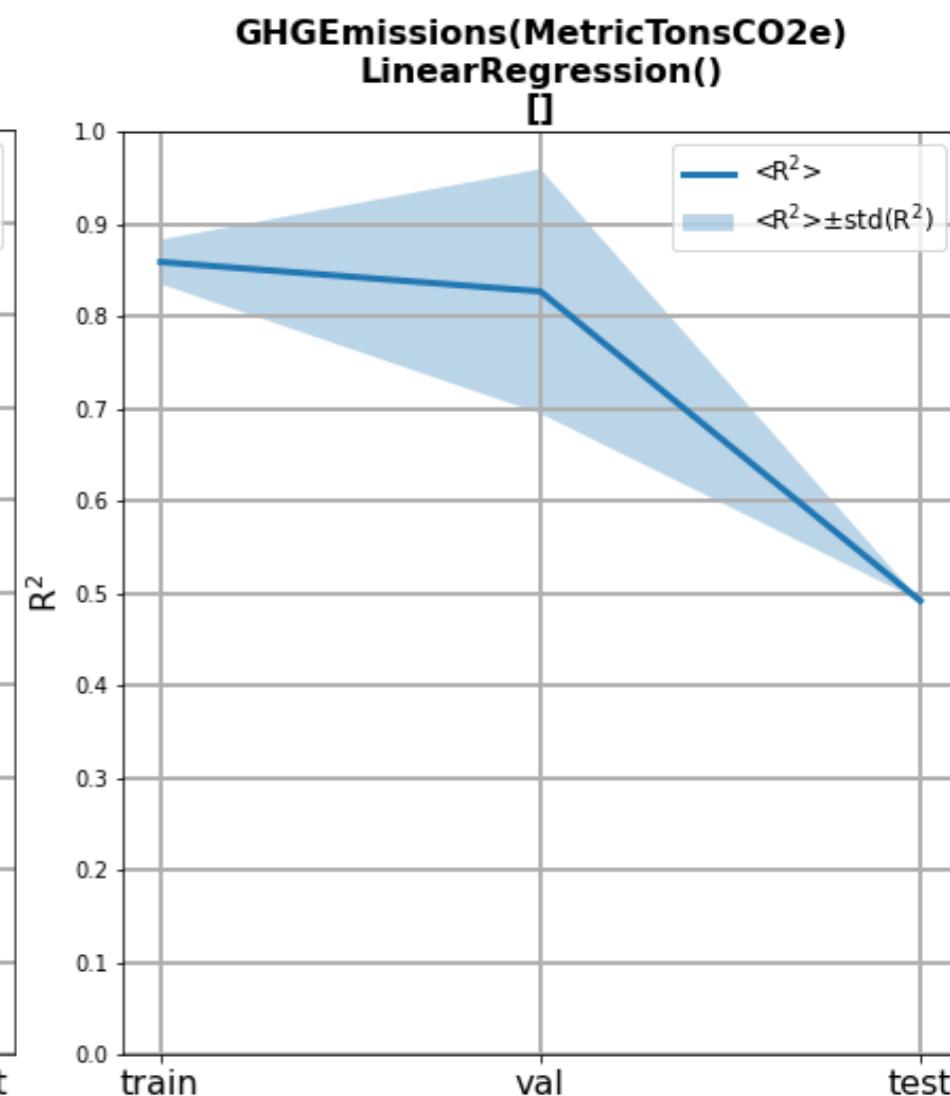
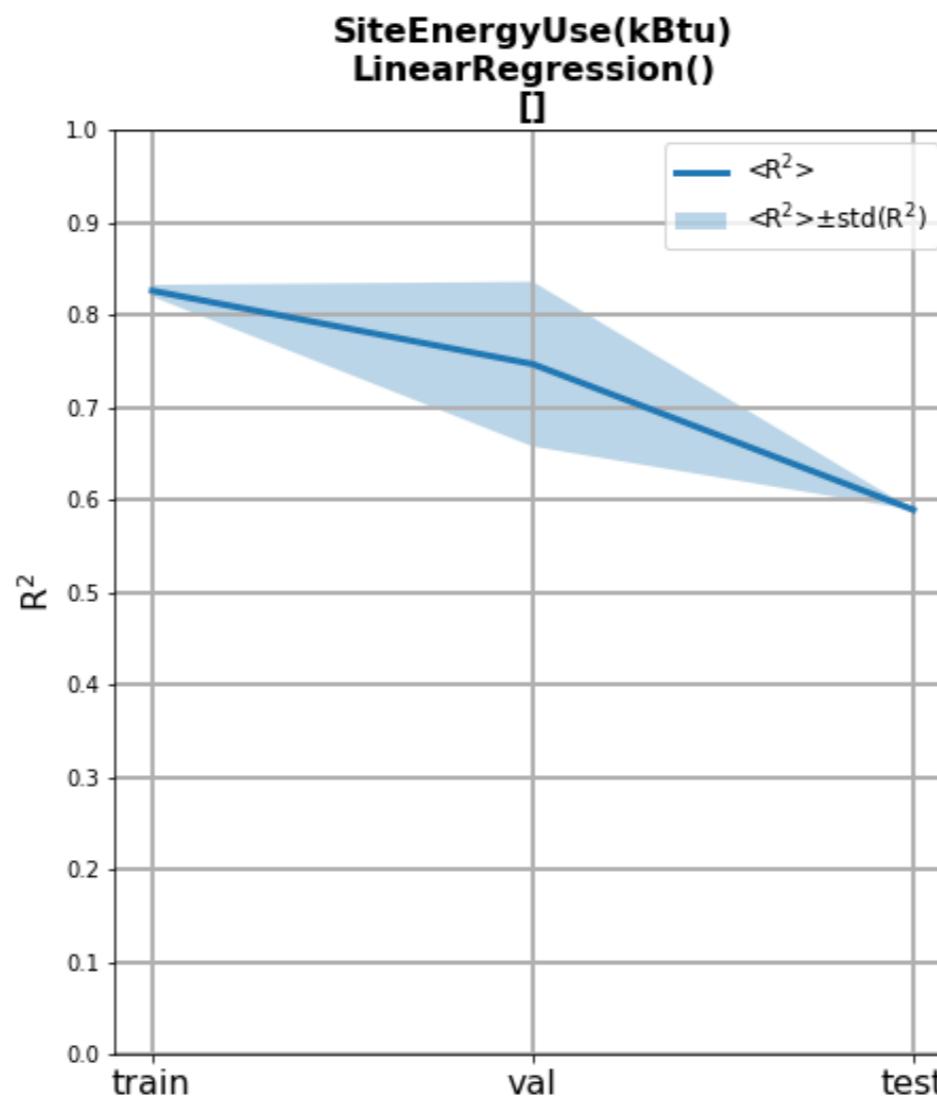
- Réduction efficace du sur-apprentissage pour *SiteEnergyUse*.
- *GHGEmissions* : Effondrement de l'apprentissage, le bon  $R^2(\text{test})$  n'est dû qu'au hasard de la séparation train-test.
- Disparition des prédictions démesurées.



# 2nde piste : régression linéaire sur données adimensionnées, - de *features*.

- En sélectionnant « manuellement » qlqs. features :  $\approx$  mêmes performances qu'avec les 199, encore améliorées avec régularisation (cf Notebook).
- Ci dessous : performances avec *PropertyTotalGFA* + param. issus du *one hot encoding* de *LargestPropertyUseType*  $\sim 1/3$  du data set adimensionné.

- ⇒ Remet en question :
  - notre sélection de *features* ;
  - notre *feature engineering*.

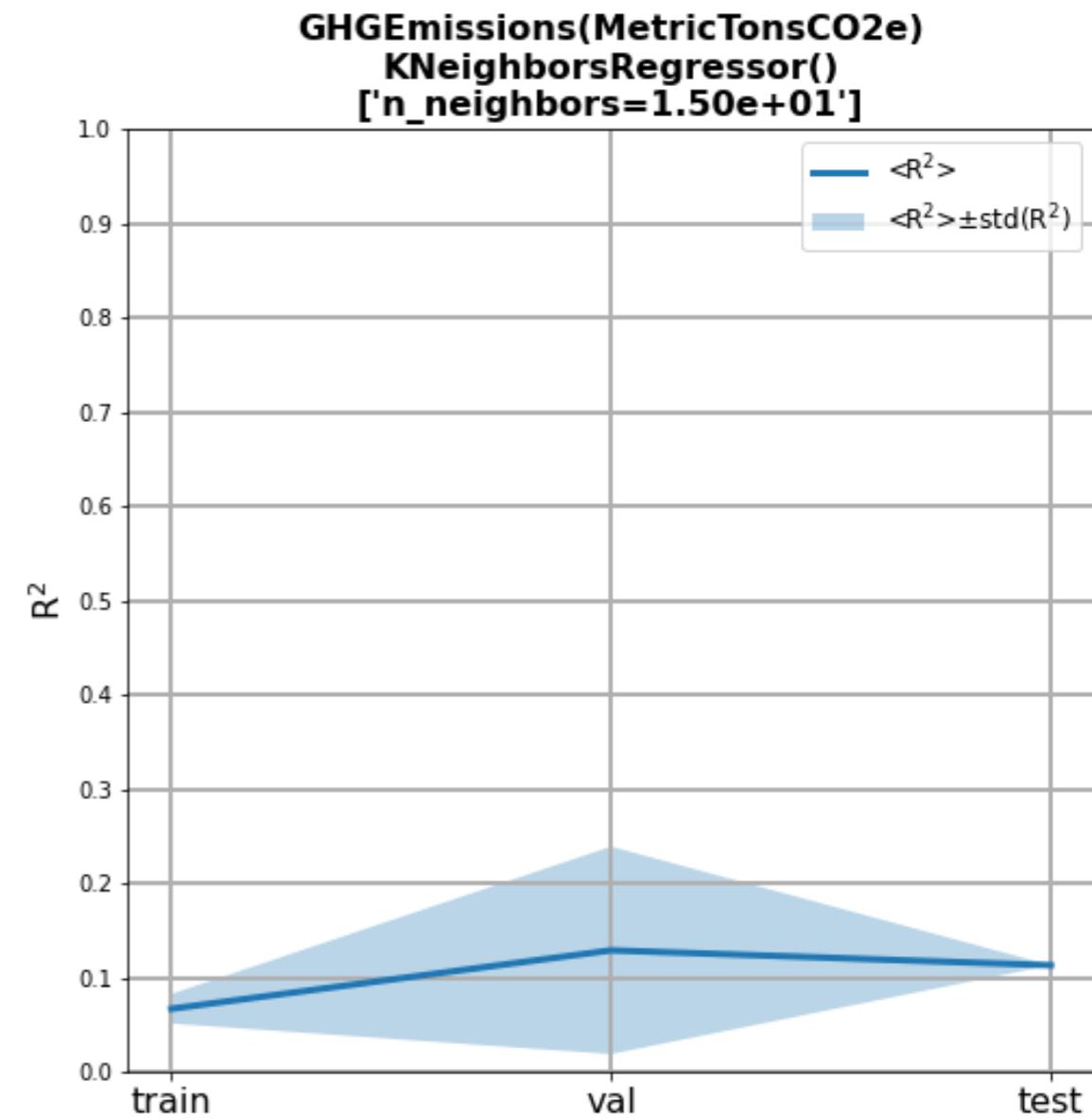
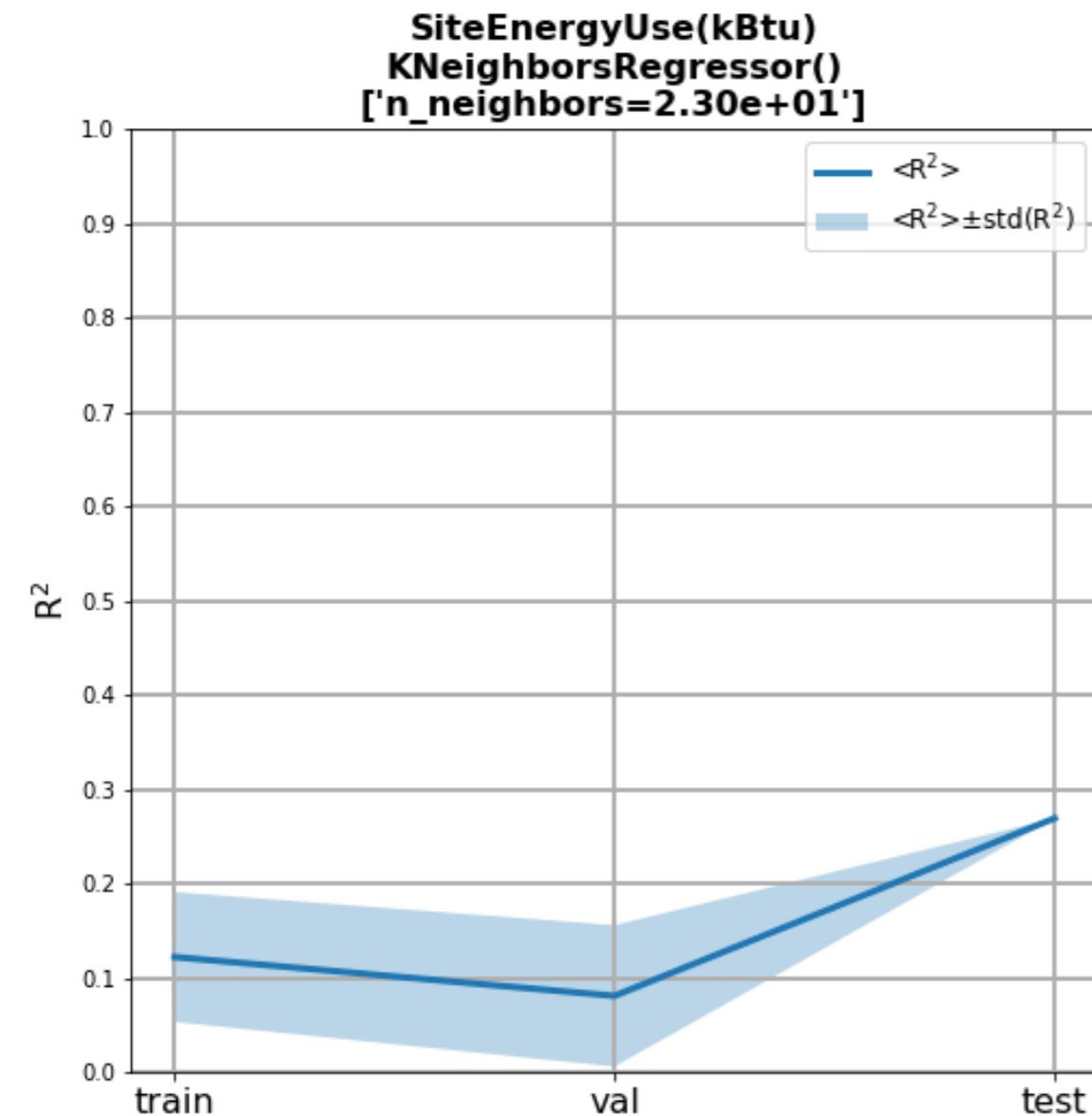


# 3ème piste : non linéarité avec KNNRegressor()

- Paramètres à optimiser :
  - nbr. de voisins ;
  - pondération des voisins.
- ⚠ Si pondération  $\propto$  proximité des voisins, risque de sur-apprentissage total :
  - car des éléments sont présents en 2015 et 2016 (mêmes valeurs pour les *features*) ;
  - car les features one hot encodées ne permettent pas de nuances dans la mesure de distances (même catégorie que l'élément courant  $\Rightarrow$  proximité totale).
- Pas de pondération, seul le nombre de voisins est à optimiser.

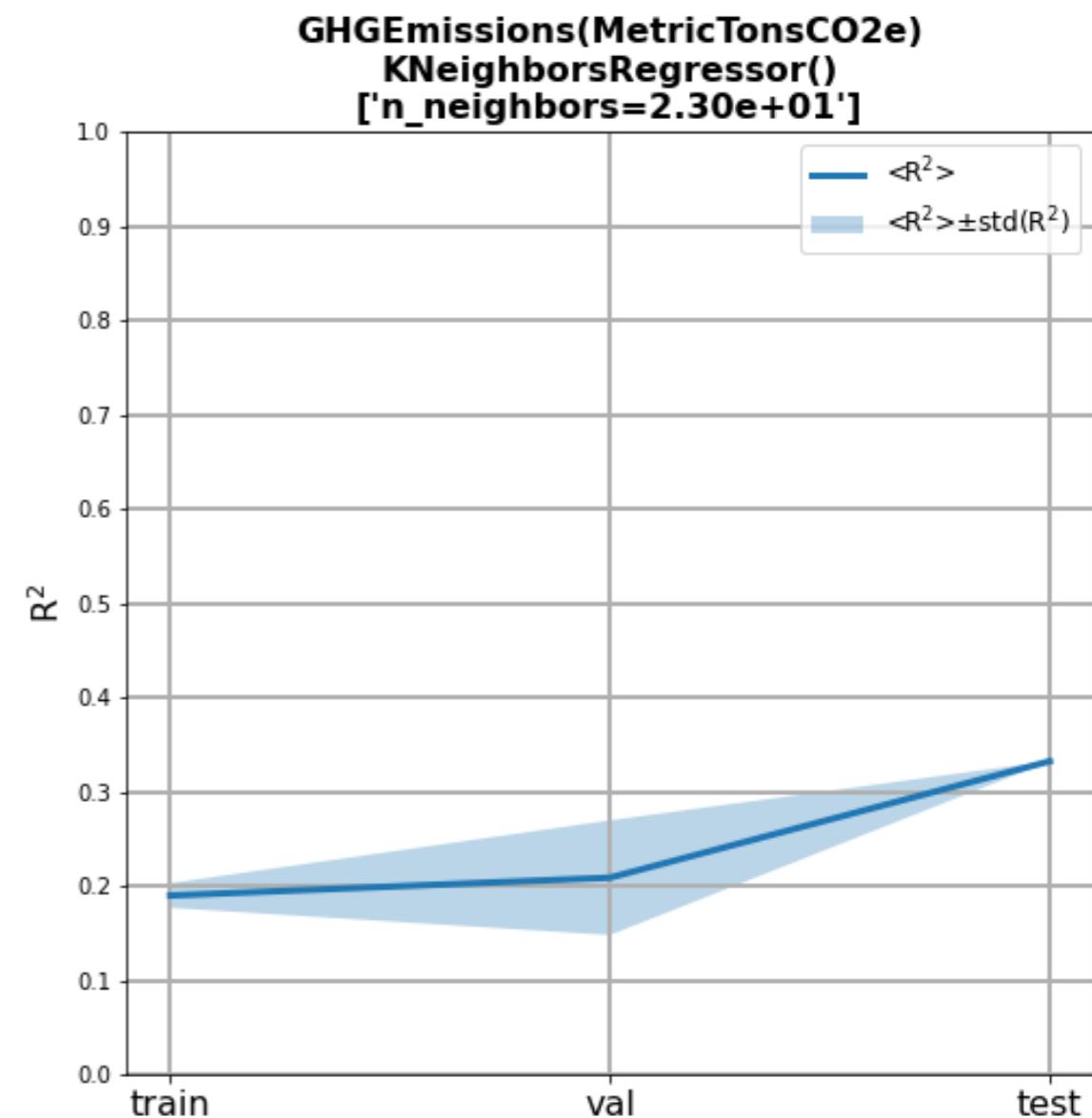
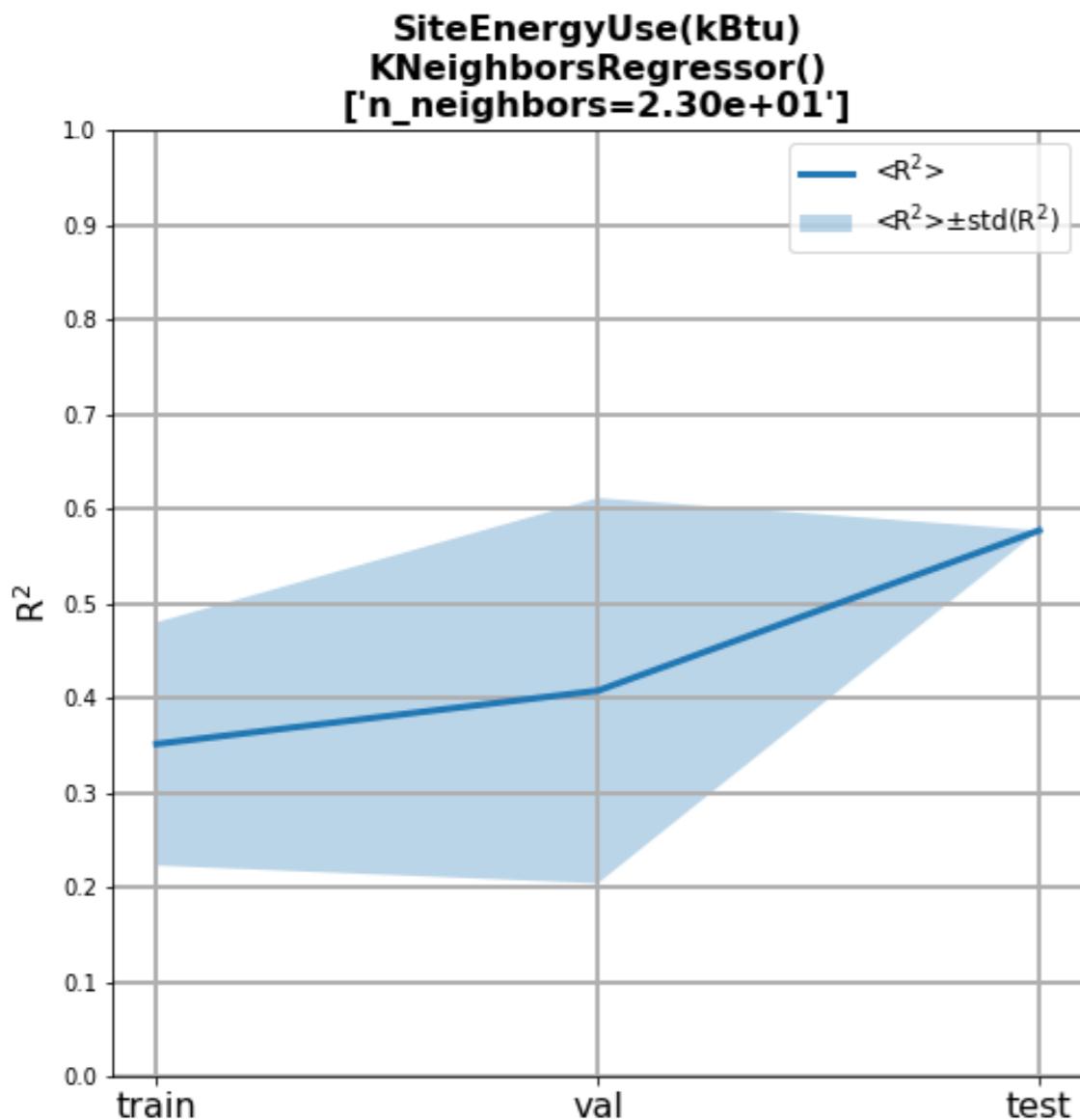
# 3ème piste : non linéarité avec KNNRegressor()

- Performances très inférieures à celles de la régression linéaire.



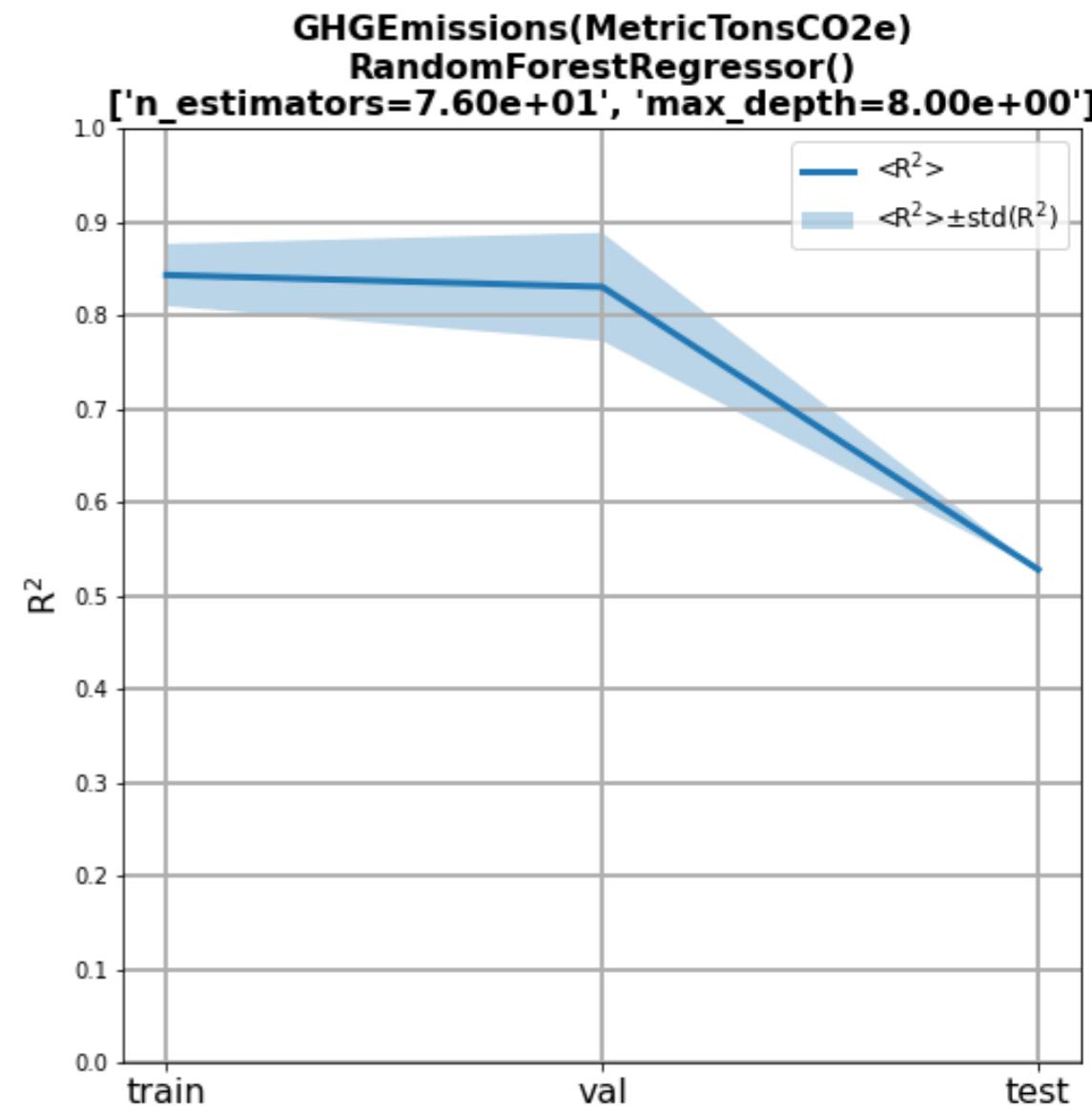
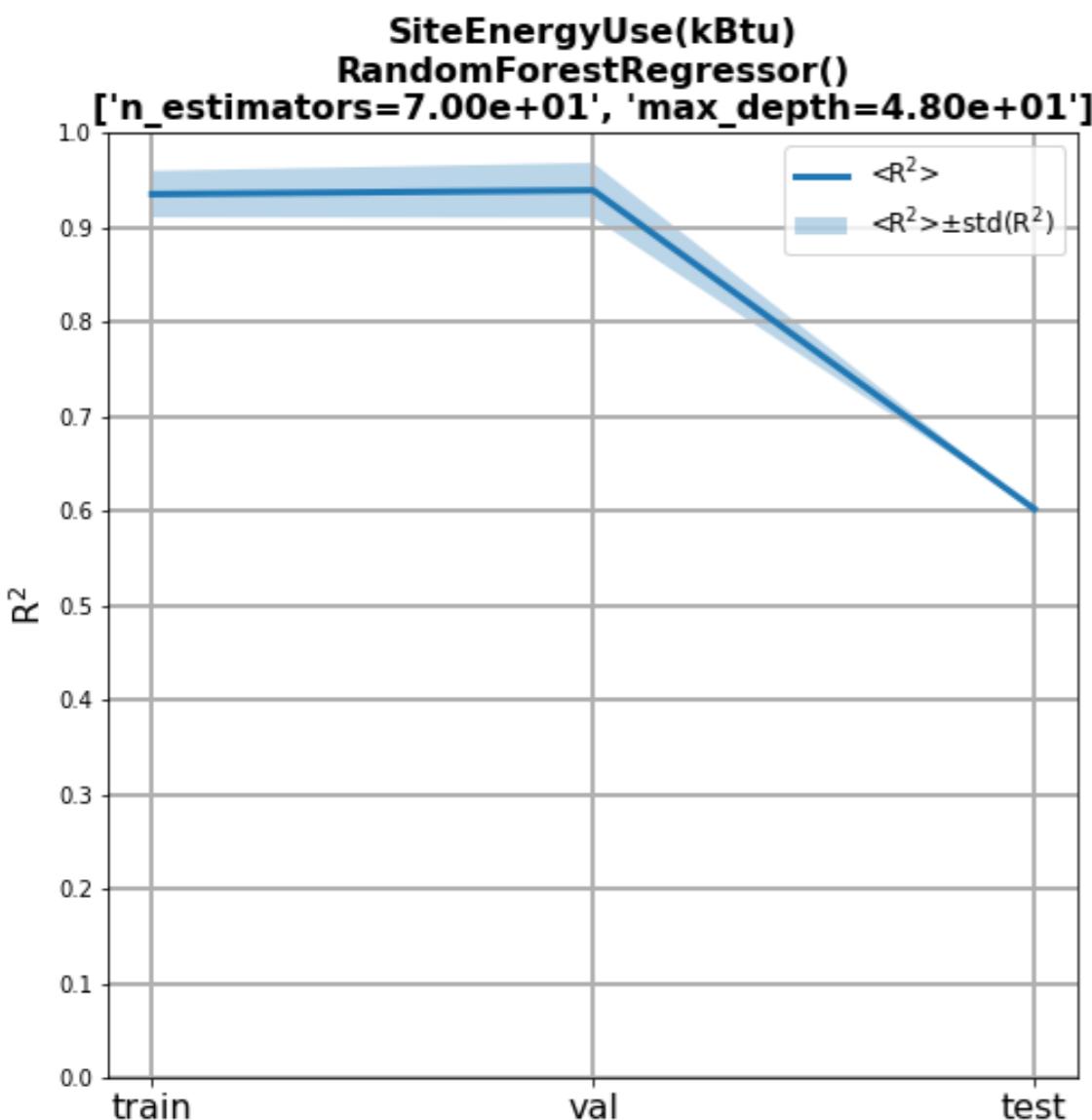
# 3ème piste : non linéarité avec KNNRegressor()

- Là aussi, augmentation nette des performances en cas de réduction du nombre de *features*.
- Ci-dessous : performances obtenues avec seulement qqs. *features quantitatives* (cf Notebook).



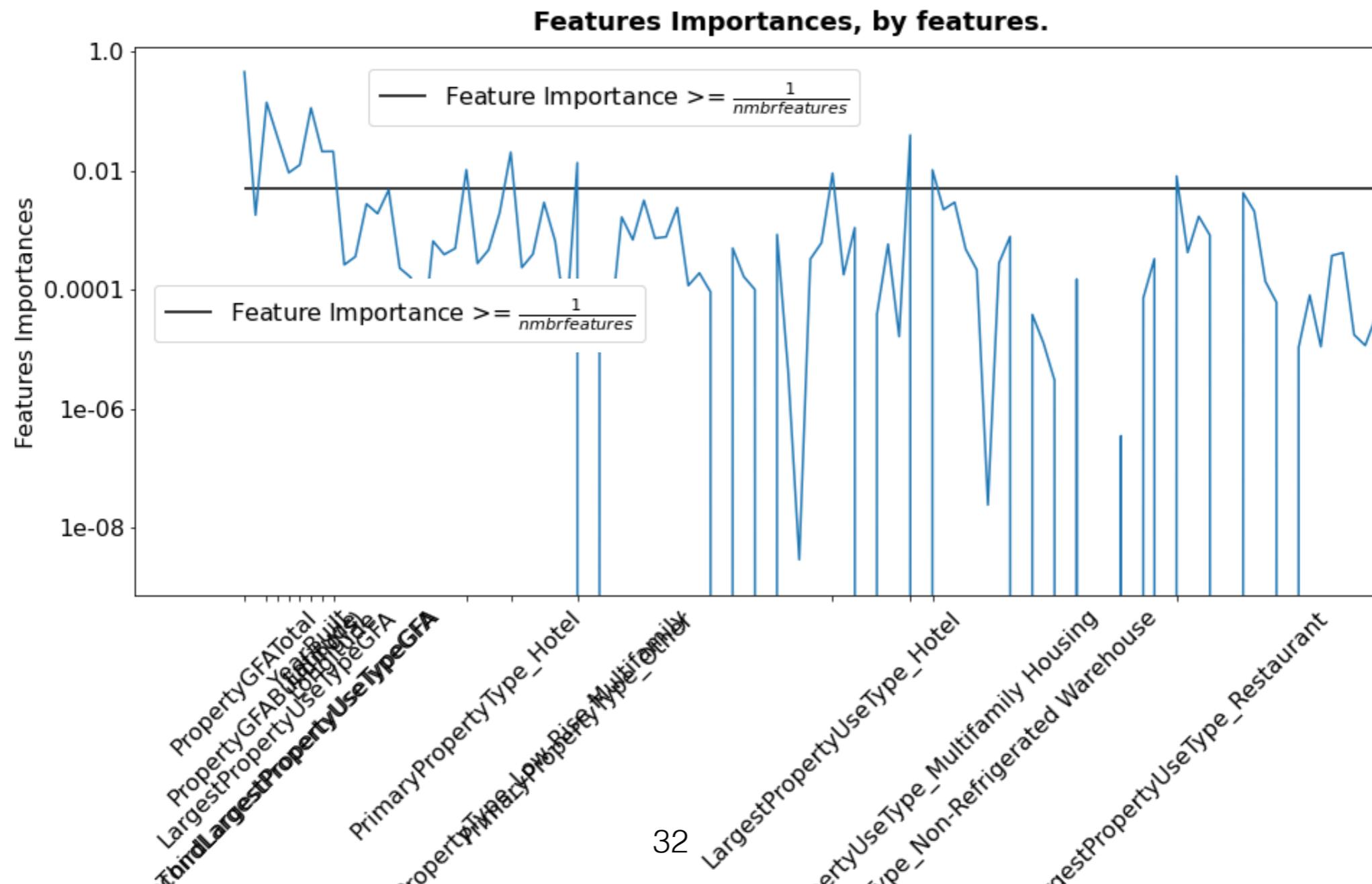
# 4ème piste : non linéarité avec forêt aléatoire.

- Optimisation du nombre d'estimateurs et de leur profondeur maximale.
- Sur-apprentissage important.



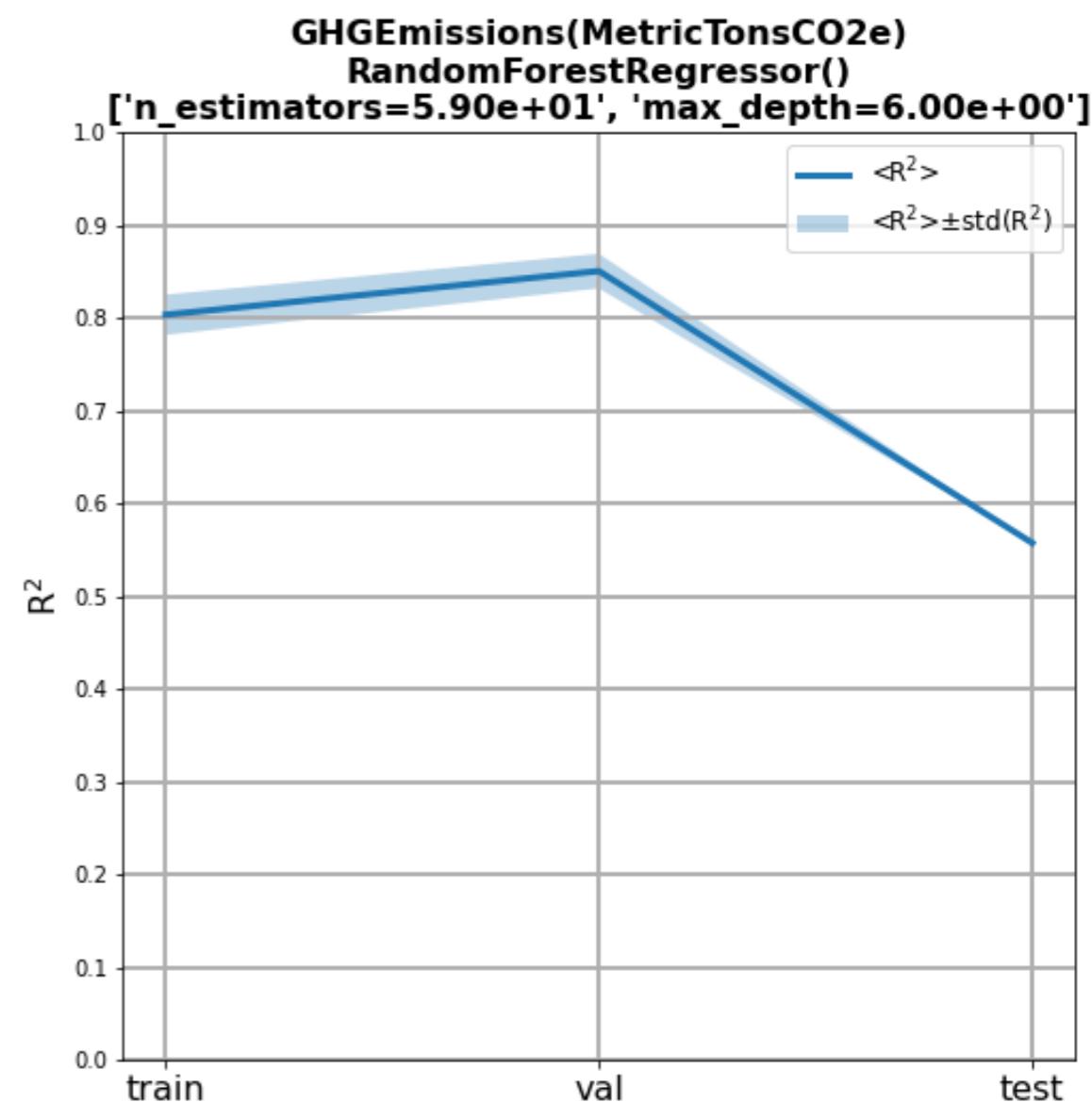
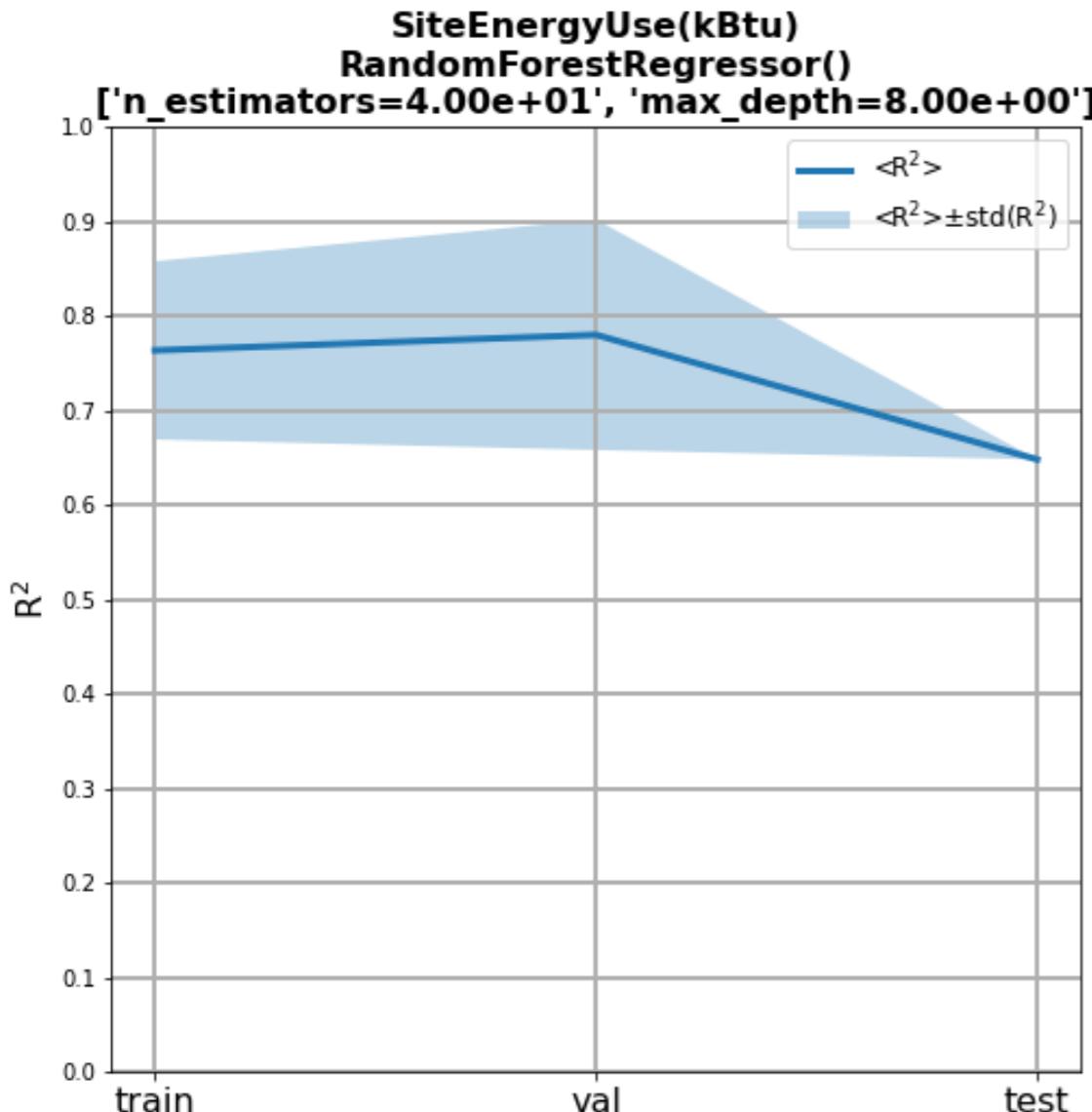
# 4ème piste : non linéarité avec forêt aléatoire.

- L'algorithme optimisé révèle que seules une dizaine de paramètres ont une importance  $\geq 1/\text{Nbr}(features)$ .
- Importance écrasée par  $PropertyGFATotal$ .



# 4ème piste : non linéarité avec forêt aléatoire.

- On re-entraîne/re-optimise l'algorithme sur ces seuls *features*, qui offrent encore de la redondance (corrélation linéaire entre *features* quantitatives).
- Amélioration des performances seulement dans le cas de *SiteEnergyUse*.

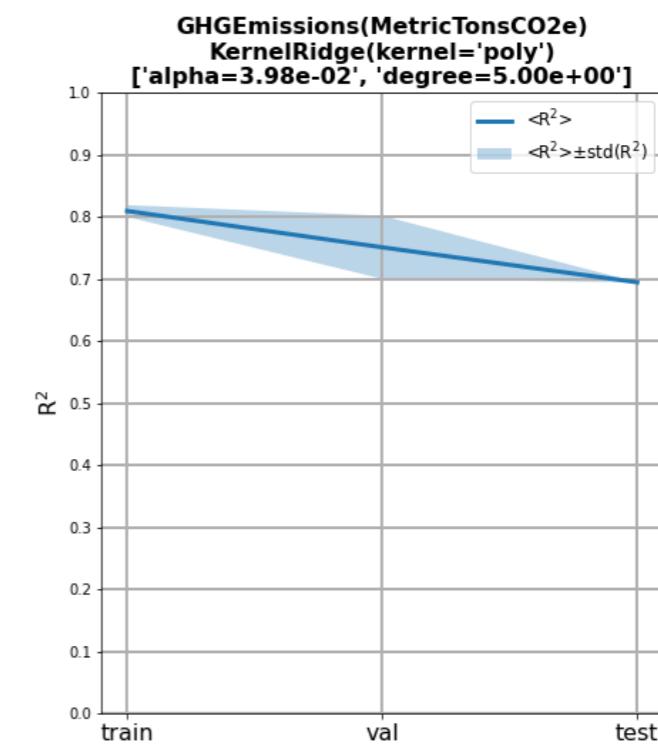
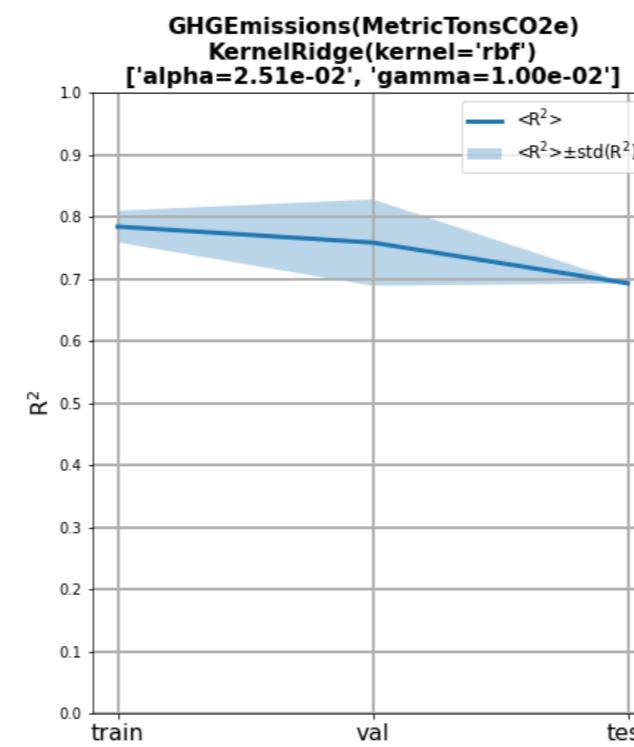
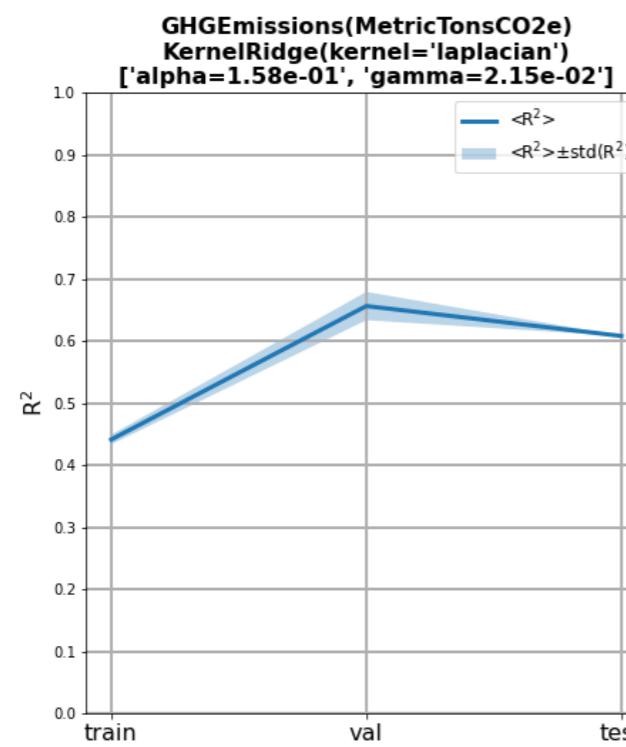
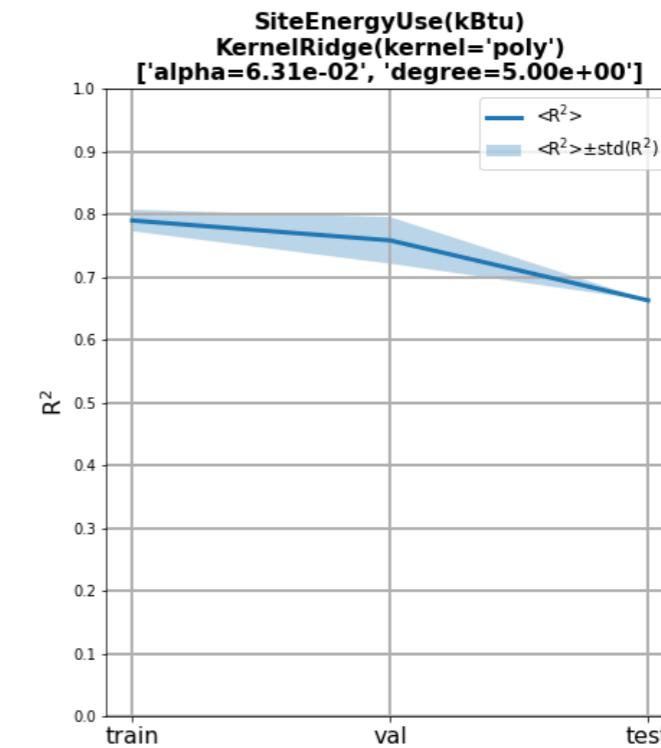
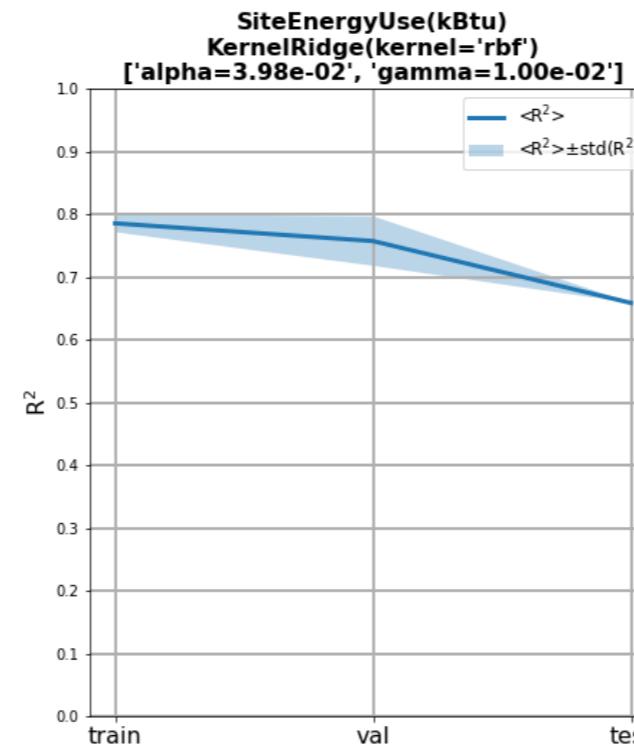
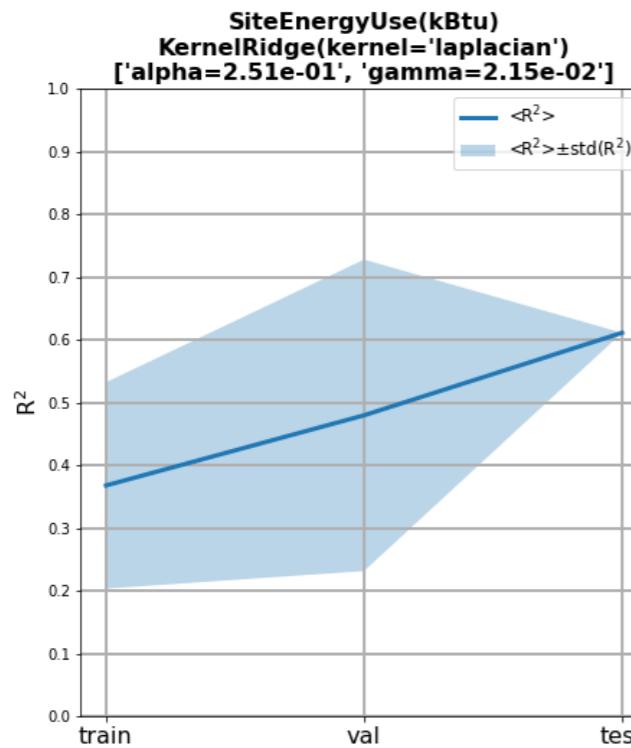


# 5ème piste : non linéarité avec noyaux.

- Mis à l'essai de noyaux
  - lapalcien (paramètre à optimiser :  $\gamma$ )
  - gaussien (paramètre à optimiser :  $\gamma$ )
  - polynomial (paramètre à optimiser : degré polynomial)
- Régularisation ridge simultanée grâce à *KernelRidge* (paramètre à optimiser :  $a$ )

# 5ème piste : non linéarité avec noyaux.

- Les 2 derniers produisent les meilleures performances.



# Plan de la présentation :

- Présentation de la problématique, de son interprétation et des pistes de recherche envisagées.
- Présentation du cleaning effectué, du feature engineering et de l'exploration.
- Présentation des différentes pistes de modélisation effectuées.
- Présentation du modèle final sélectionné ainsi que des améliorations effectuées.

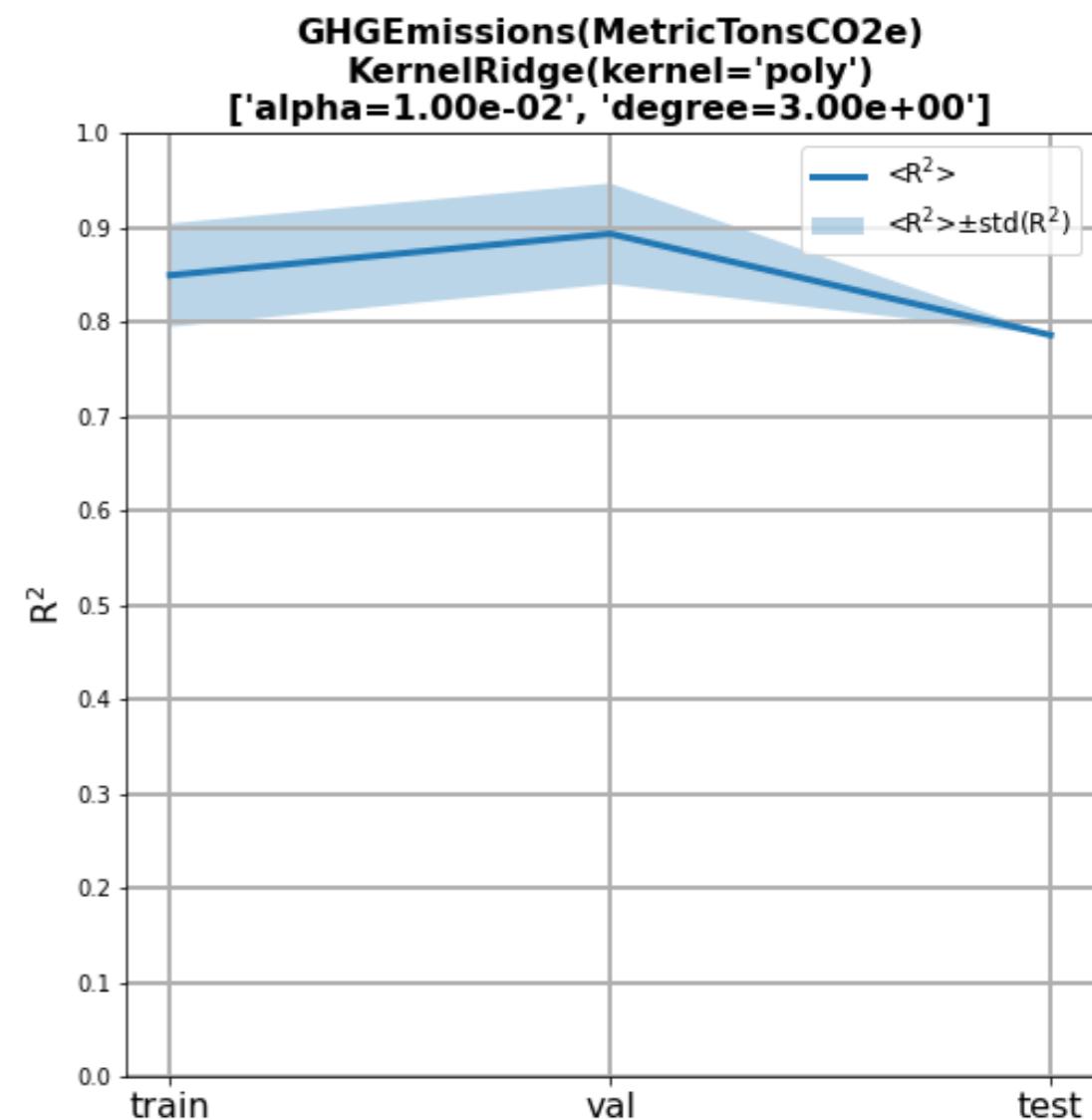
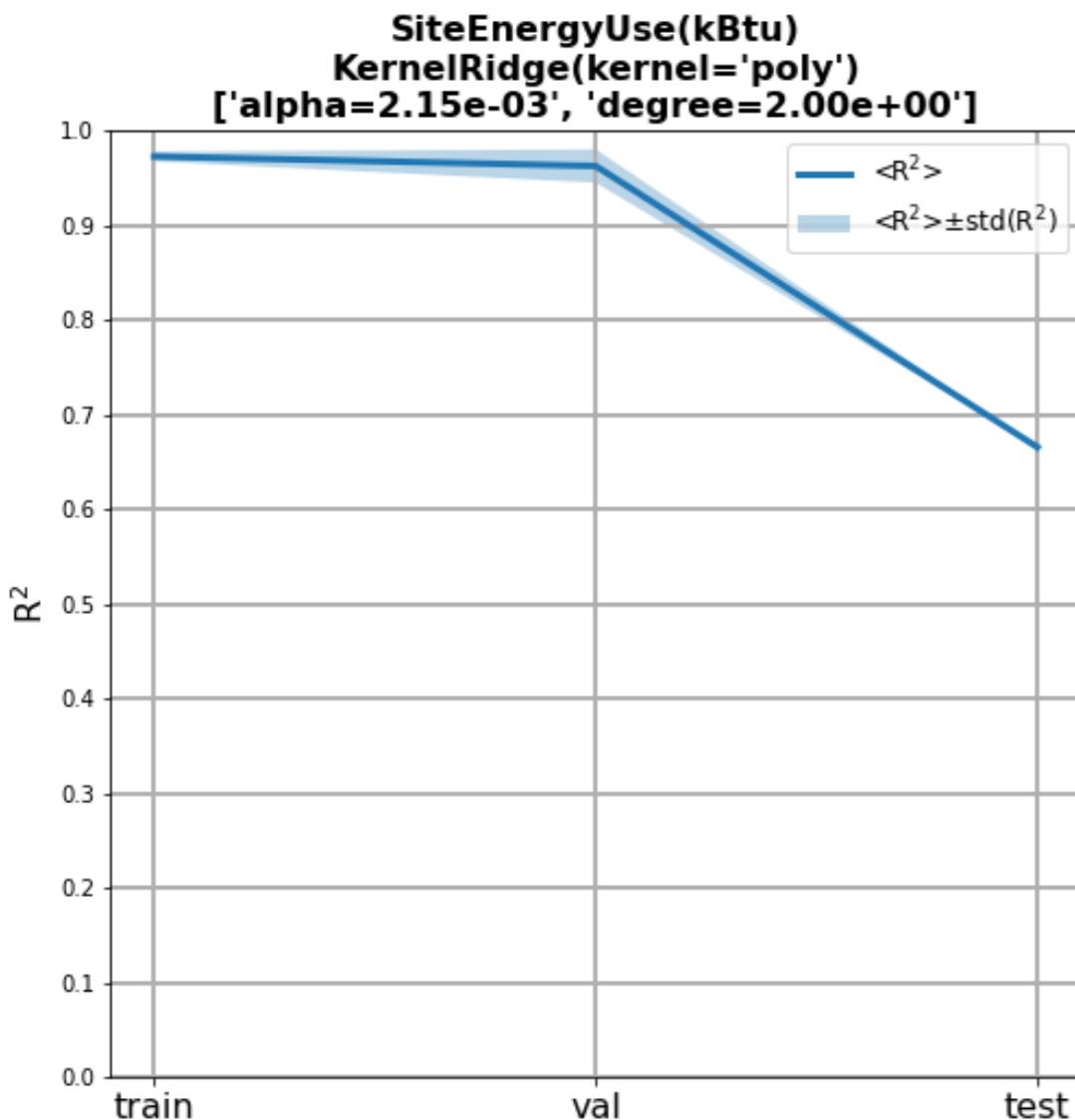
# Résumé des performances de tous ces modèles (algos + data set) :

- On fait le bilan des modèles (algo optimisé + *data set*) pour chaque étiquette.
- Noyau polynomial un des meilleurs SIMULTANÉMENT pour les deux étiquettes [bon R<sup>2</sup>(test) et faible sur-apprentissage] ⇒ on le retient pour tester *ENERGYSTARScore*.

Y	Format data	Modèle	Hyper-paramètres	R <sup>2</sup> (train)	R <sup>2</sup> (val)	R <sup>2</sup> (test)	RMSE(test)	MAE(test)
GHGEmissions(MetricTonsCO2e)	dataQuantNonNorm	LinearRegression()	{} {'alpha': 2.154434690031882}	0.838961 0.690764	0.775721 0.658780	0.336598 0.650882	158.604728 115.057164	78.974443 53.556410
GHGEmissions(MetricTonsCO2e)	dataQuantNorm	LinearRegression()	{} {'alpha': 0.0014677992676220691}	0.716230 0.470818	0.601231 0.430707	0.545674 0.713039	131.621421 104.313147	56.698837 49.902632
GHGEmissions(MetricTonsCO2e)	dataQuantNorm	Ridge()	{} {'alpha': 0.6812920690579608}	0.012743 0.317771	-0.050324 0.345416	0.682251 0.697980	109.766470 107.015275	52.454803 51.532342
GHGEmissions(MetricTonsCO2e)	dataQuantNorm_sFeatCoeff0	LinearRegression()	{} {'alpha': 0.00046415888336127773}	0.208854 0.865884	0.208975 0.807493	0.712302 0.491487	104.447007 135.426749	50.733182 57.524057
GHGEmissions(MetricTonsCO2e)	dataQuantNorm_sFeatCoeff0	Ridge()	{} {'alpha': 1.4677992676220675}	0.595331 0.464447	0.677745 0.615669	0.640869 0.689298	116.695628 108.542535	55.264355 53.037496
GHGEmissions(MetricTonsCO2e)	dataQuantNorm_featreduits	LinearRegression()	{} {'alpha': 0.0014677992676220691}	0.184661 0.797590	0.258811 0.720018	0.332160 0.692672	159.134439 107.951480	62.356930 51.315641
GHGEmissions(MetricTonsCO2e)	dataQuantNorm	KNeighborsRegressor()	{} {'n_neighbors': 23}	0.509087 0.800655	0.520187 0.764664	0.607840 0.694395	121.943769 107.648544	52.345750 51.157558
GHGEmissions(MetricTonsCO2e)	dataQuantNorm	KernelRidge(kernel='laplacian')	{} {'alpha': 0.15848931924611143, 'gamma': 0.0215...}	0.885606 0.828648	0.898033 0.729248	0.520812 0.520534	134.797157 134.836308	55.093471 56.677106
GHGEmissions(MetricTonsCO2e)	dataQuantNorm_FeatImport	KernelRidge(kernel='rbf')	{} {'alpha': 0.025118864315095794, 'gamma': 0.01}	0.828648 0.800655	0.729248 0.764664	0.520534 0.694395	134.836308 107.648544	56.677106 51.157558
GHGEmissions(MetricTonsCO2e)	dataQuantNorm	KernelRidge(kernel='poly')	{} {'alpha': 0.039810717055349734, 'degree': 5.0}	0.828648 0.800655	0.729248 0.764664	0.520534 0.694395	134.836308 107.648544	56.677106 51.157558

# Pertinence d'ENERGYSTARScore :

- Nouvelles NaN issues d'ESS  $\Rightarrow$  nouveau *data set* (+1 colonne, -1600 lignes). Après optimisation d'un noyau polynomial, conclusions  $\neq$  selon étiquettes :
  - *SiteEnergyUSE* : dégradation du sur-apprentissage, et  $R^2(\text{test})$  inchangé pour hyper-paramètres plus contraignants ;
  - *GHGEmissions* : meilleur apprentissage et meilleure généralisation.



# Conclusion : résumé et critiques du projet.

- Analyse exploratoire des données basée sur
  - l'analyse bi-variée des étiquettes ;  
⇒ a probablement conduit à une mauvaise sélection des *features* ;
  - la non corruption du *data set* par la suppression des NaN ;  
⇒ a été dominée par les *features* les moins pertinents (*3rdLargest[...]*).
- Plusieurs modèles testés pour chercher à maximiser  $R^2(\text{test})$  sans générer de surapprentissage ;  
⇒ peut être trop de modèles différents, coûteux en temps de calcul (surtout avec 199 *features*).
- Mise en évidence de PB dans le choix des *features* ;  
⇒ fait trop tardivement, aurait dû être l'une des 1ères étapes.
- Prise en compte de l'ENERGYSTARScore pertinente seulement pour la production de CO<sub>2</sub> (*GHGEmissions*).

# Conclusion : pistes d'améliorations.

- Conserver tous les paramètres suggérés par l'intitulé du projet indépendamment de l'analyse univariée.
- Utiliser en 1er Lasso et/ou RandomForrest pour faire de la sélection des *features* ⇒ permettra un gain de temps lors du test des algorithmes.
- Ne plus chercher à conserver hiérarchies entre *features* lors du *features engineering*.
- Travailler sur un plus grand nombre de données : imputation au lieu de réduction du nombre de lignes.
- Chercher si l'alternative « PB de classification » est possible (*clustering* sur les étiquettes).

**FIN DE SOUTENANCE, MERCI  
POUR VOTRE ATTENTION.**

Soutenance du projet n°3 : Parcours « Ingénieur Machine Learning »

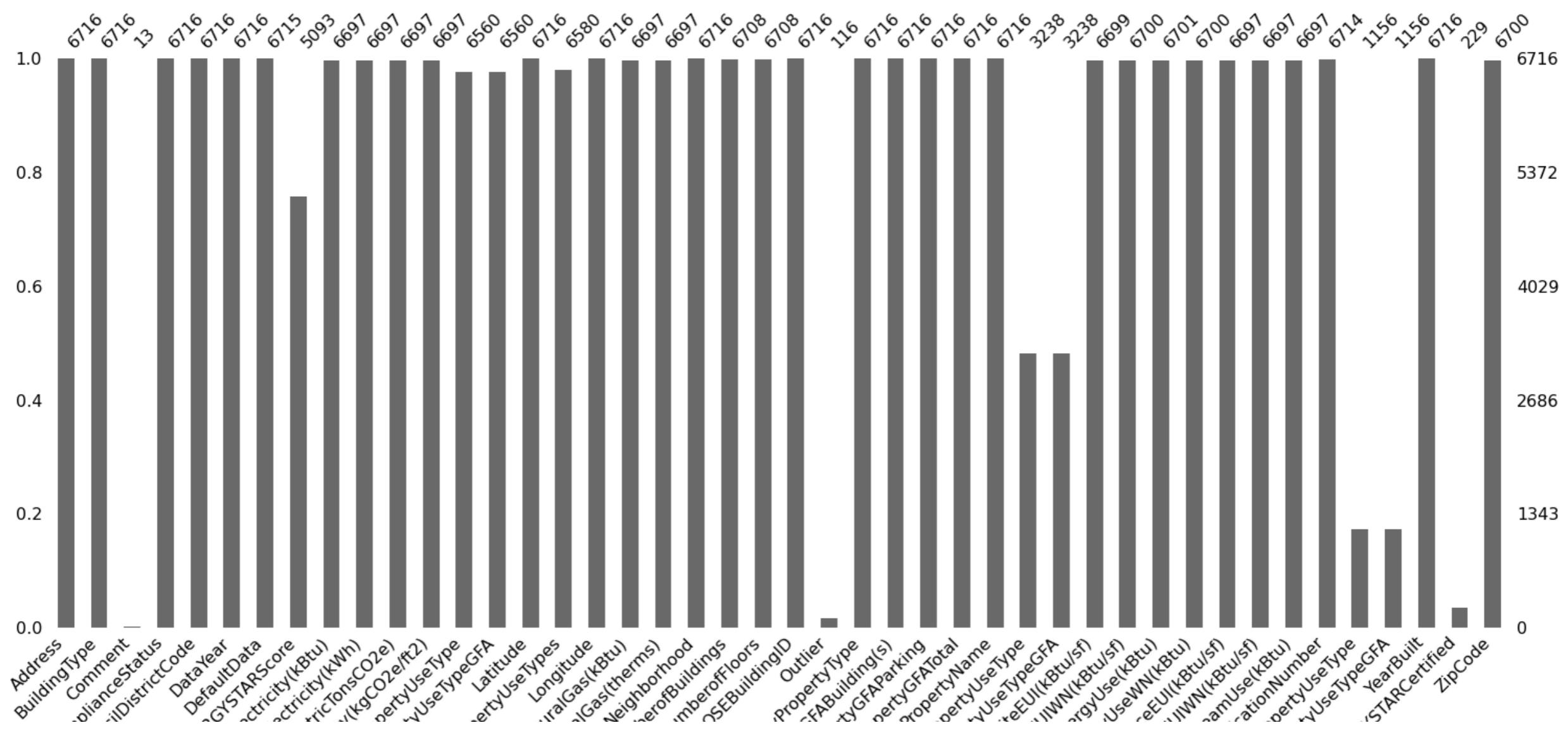
Luke Duthoit

# ANNEXE : Cleaning

## Cleaning effectué :

# Suppression d'embrée de certains paramètres

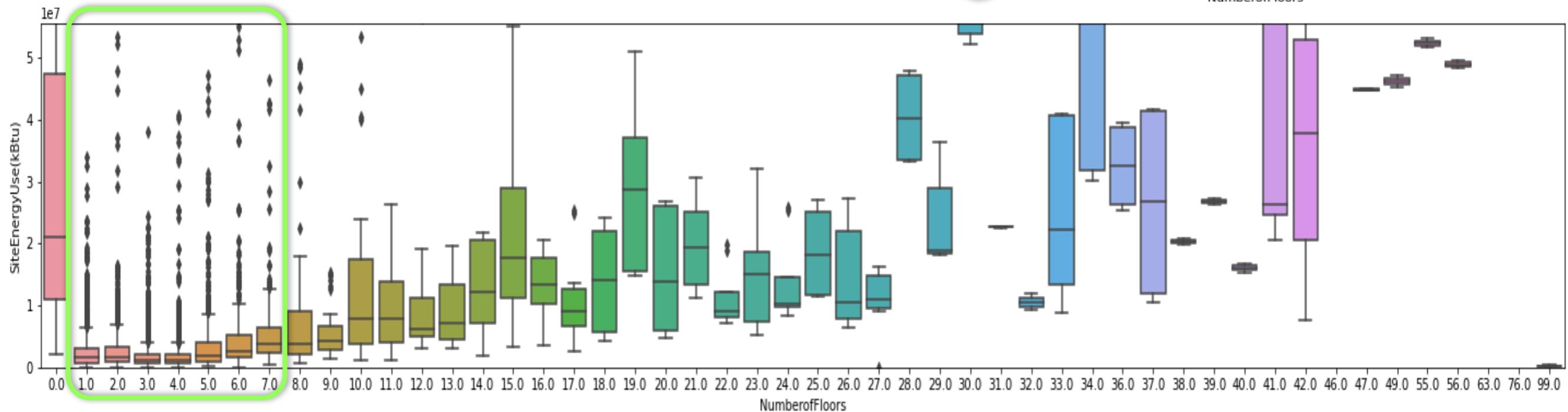
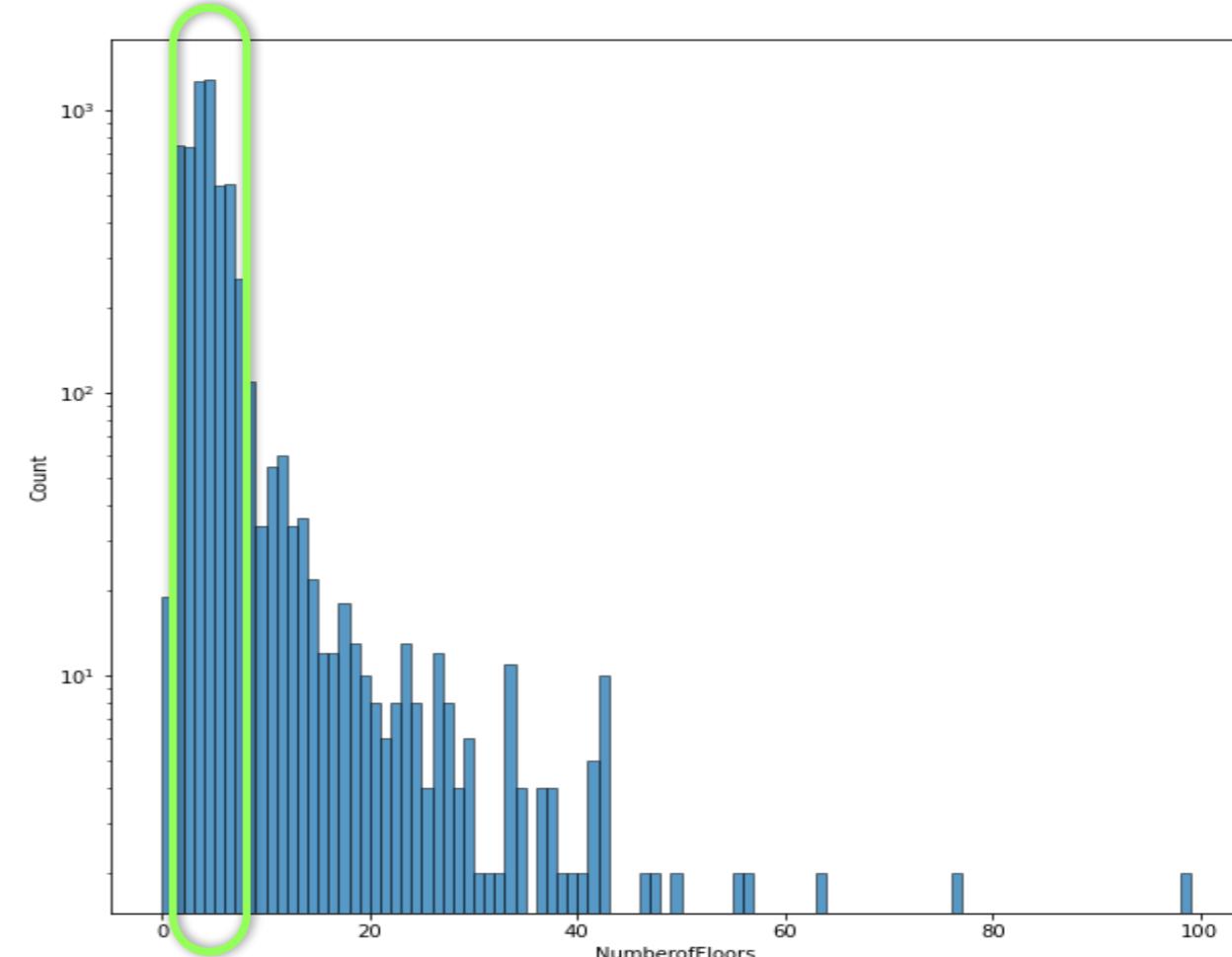
- Ceux spécifiques à 2015, soit trop peu nombreux (ex : *2010 Census Tracts*), soit inutiles (ex : *SPD Beats*).
- Suppression des paramètres communs les moins remplis (*Comment*, *Outliers*) et/ou inutiles (*ENERGYSTARCertified*, *Adress*, etc).



## ANNEXE : *Feature selection (1)*

### Nombre d'étage : *NumberofFloors*.

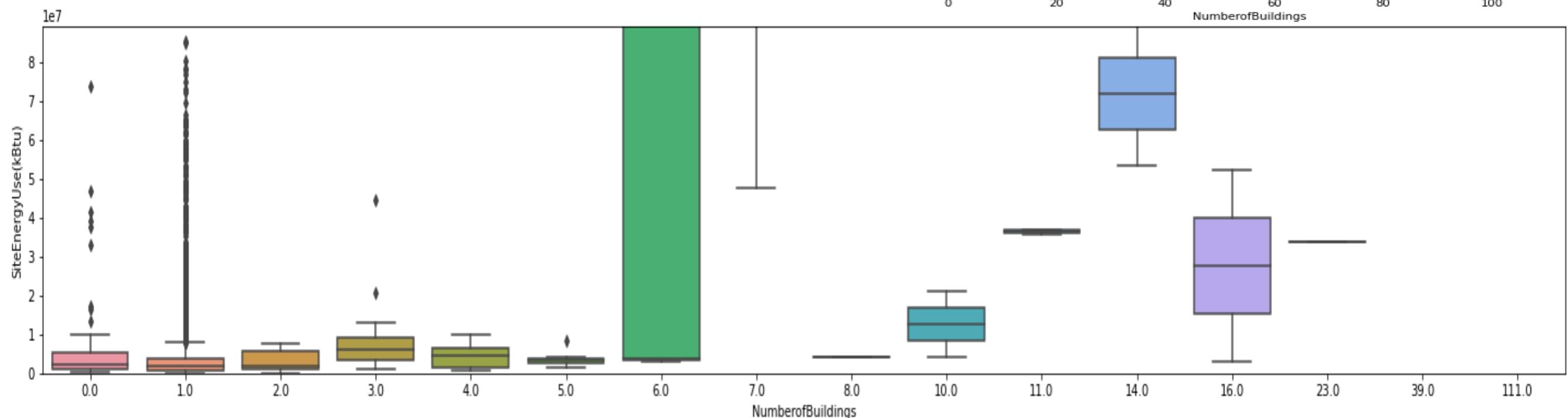
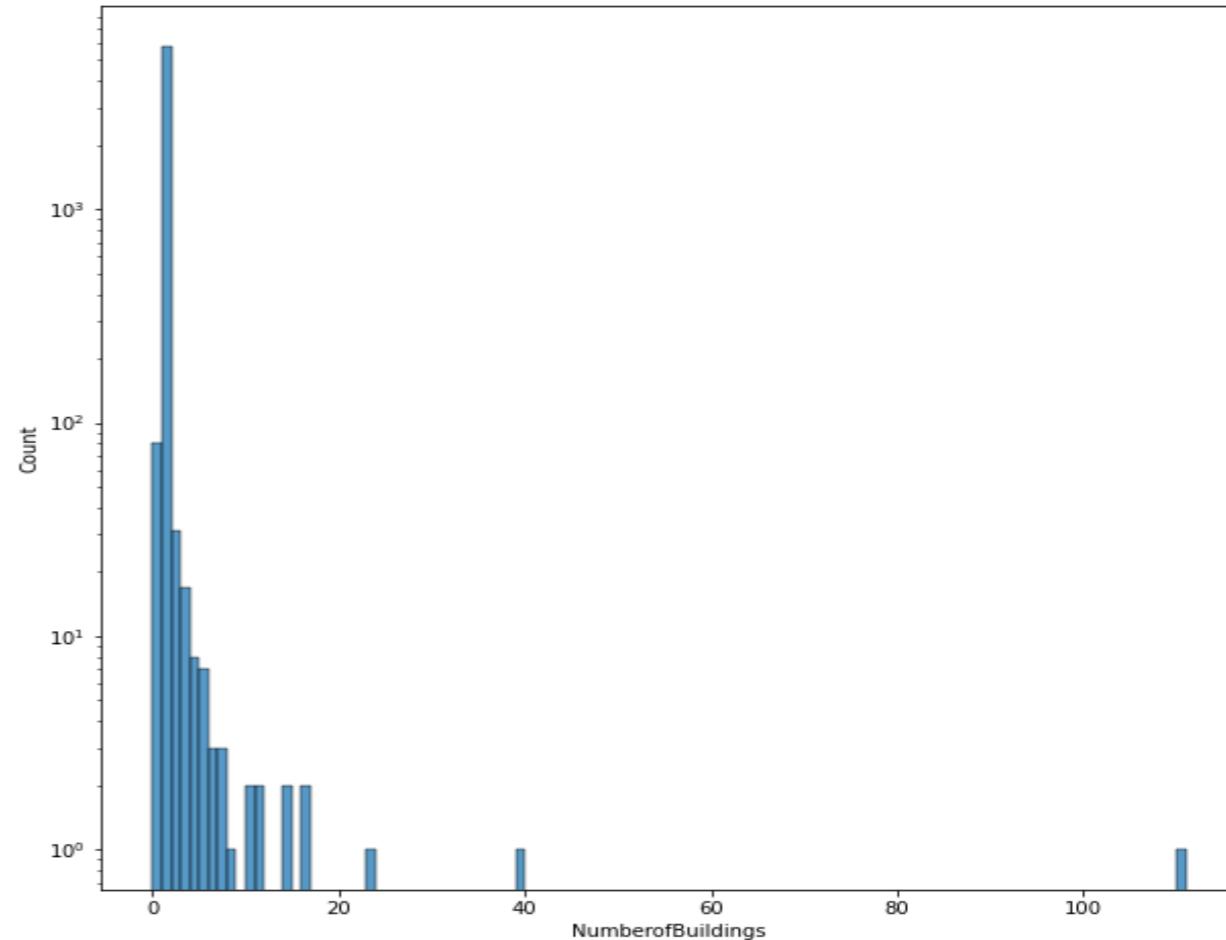
- Distribution dominée par les faibles valeurs non nulles (>80% entre 1 et 7)
- Analyse mono-variée : les distributions des étiquette en fonction du nombre d'étages sont très similaires pour la majorité des éléments !
- ⇒ On se sépare de ce paramètre.



# ANNEXE : *Feature selection (2)*

## Nombre d'immeubles : *NumberOfBuildings.*

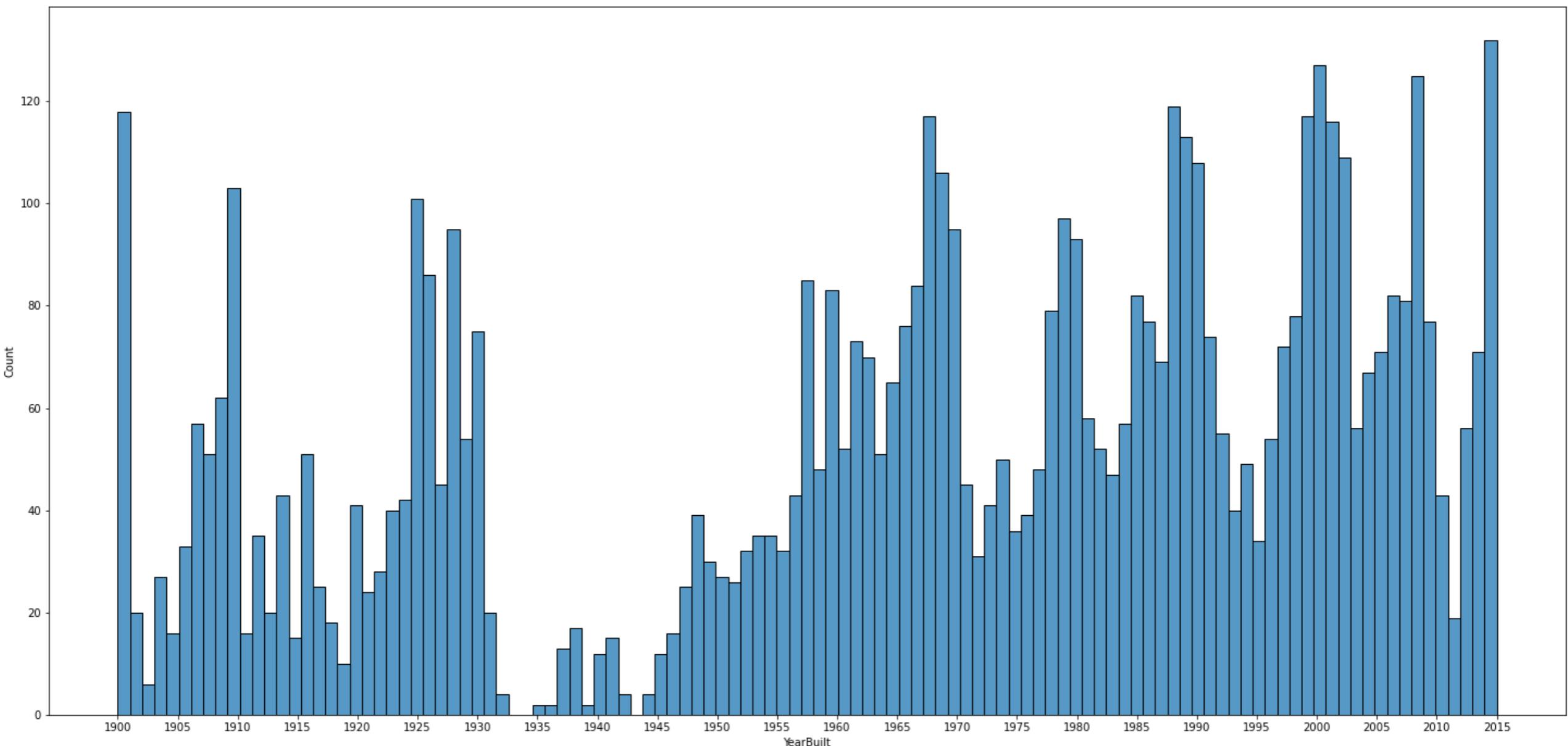
- Distribution écrasée par une seule valeur : 1.
- Analyse mono-variée : même effet que pour *NumberofFloors*, en pire.
- ⇒ On se sépare de ce paramètre.



## ANNEXE : *Feature selection (3)*

### Année de construction : *YearBuilt.*

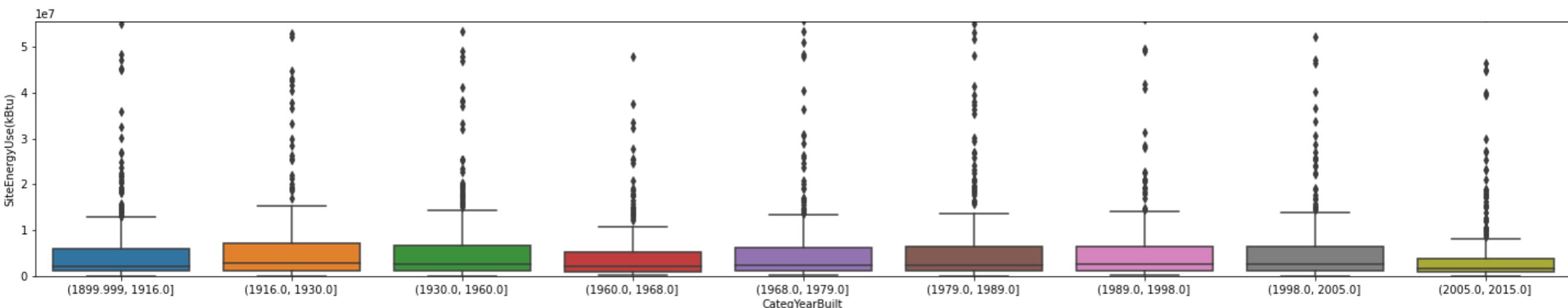
- Distribution périodique avec pic de construction  $\approx$  tous les 10 ans  $\Rightarrow$  on va tenter une analyse uni-variée en la découplant en déciles.



## ANNEXE : *Feature selection (4)*

# Année de construction : *YearBuilt.*

- Distribution périodique avec pic de construction  $\approx$  tous les 10 ans  $\Rightarrow$  on va tenter une analyse uni-variée en la découplant en déciles.



- Distributions particularisées au déciles quasi-identiques.
- On laisse de côté ce paramètre.

# ANNEXE : *Feature selection (5)*

## Paramètres de type d'usage :

	BuildingType	PrimaryPropertyType	ListOfAllPropertyUseTypes	LargestPropertyUseType	SecondLargestPropertyUseType	ThirdLargestPropertyUseType
<b>count</b>	5955	5955	5820	5801	2940	1073
<b>unique</b>	7	32	465	56	48	45
<b>top</b>	NonResidential	Low-Rise Multifamily	Multifamily Housing	Multifamily Housing	Parking	Retail Store
<b>freq</b>	2562	1823	1584	2981	1673	200

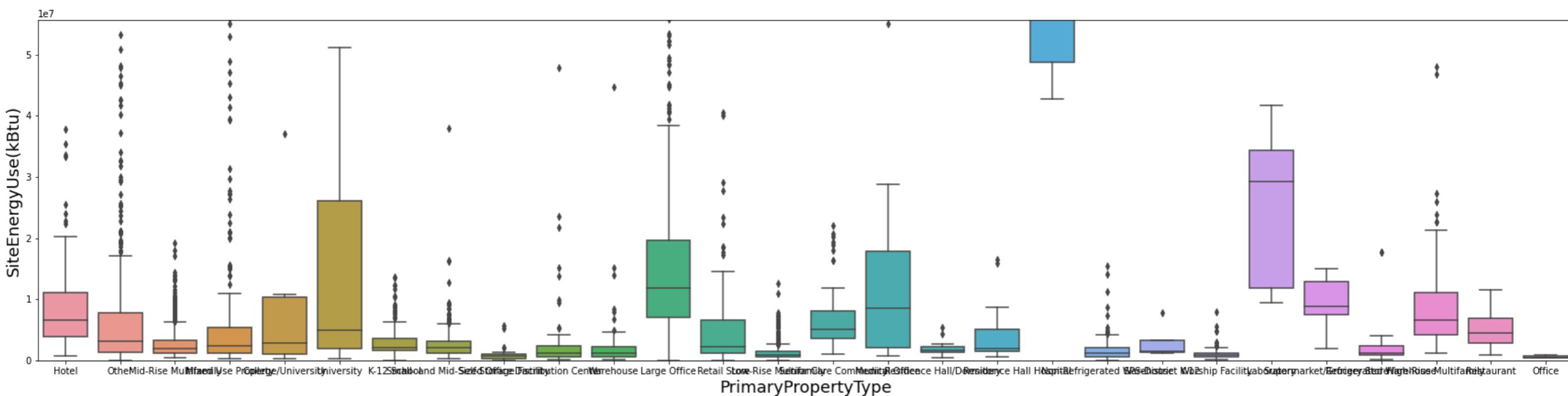
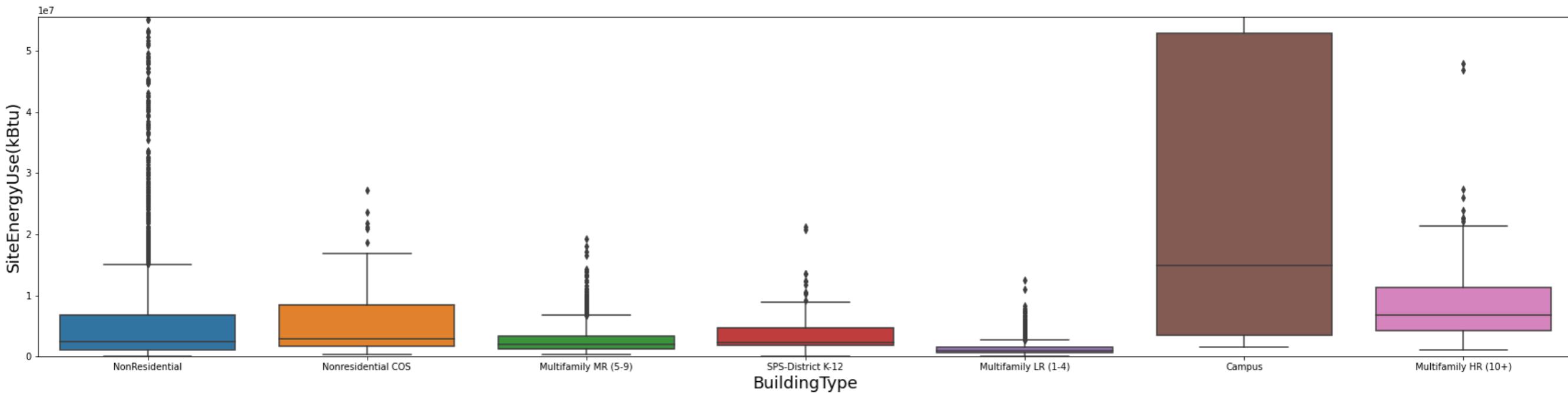
- Grande disparité du nombre de catégories possibles [*unique*] ⇒ différentes échelles de description.
- Nettoyage : Réduction de *unique* pour plusieurs paramètres après correction de formats différents pour catégories en réalité identiques.
- ListOfAllPropertyUseTypes* : unique élevé du fait de la nature de ce paramètre, apporte peu d'information supplémentaire v.à.v. des autres paramètres → on s'en sépare.

Low-Rise Multifamily	1823
Mid-Rise Multifamily	1010
Small- and Mid-Sized Office	533
Other	430
Large Office	319
Mixed Use Property	256
K-12 School	183
High-Rise Multifamily	178
Warehouse	170
Non-Refrigerated Warehouse	168
Retail Store	156
Hotel	130
Worship Facility	119
Medical Office	73
Senior Care Community	72
Distribution Center	47
Distribution Center\n	45
Supermarket / Grocery Store	31
Supermarket/Grocery Store	27
Self-Storage Facility	25
Self-Storage Facility\n	23
Refrigerated Warehouse	22
Residence Hall	20
University	19
College/University	15
Restaurant	13
Residence Hall/Dormitory	13
Restaurant\n	11
Hospital	9
Laboratory	8
SPS-District K-12	4
Office	3
Name: PrimaryPropertyType, dtype: int64	

# ANNEXE : *Feature selection (6)*

## Paramètres de type d'usage :

- Analyse mono-variée : plus il y a de catégories ≠, plus les distributions particularisées des étiquettes se distinguent ⇒ on garde *PrimaryPropertyUseType*.



## ANNEXE : *Feature selection* (7)

### Paramètres de type d'usage :

- De même, on garde *First/Second/ThirdPropertyUseType* pour conserver relations hiérarchiques (cf paramètres surfaciques).
- ⚠ On garde *BuildingType* au cas où le niveau faiblement détaillé de description suffirait.

# ANNEXE : *train set, test set (1).*

## Séparation du *data set* en jeux d'entraînement et de test.

- Les *targets* et les *features* quantitatives conservent ~ les mêmes distributions sur les deux jeux : ci-dessous, rapport des propriétés statistiques de leurs distributions, entre test set et train set.

	SiteEUI(kBtu/sf)	SiteEnergyUse(kBtu)	NaturalGas(kBtu)	GHGEmissions(MetricTonsCO2e)	GHGEmissionsIntensity(kgCO2e/ft2)
count	0.250471	0.250471	0.250471	0.250471	0.250471
mean	1.136676	0.945342	0.985629	0.933675	1.180801
std	1.328203	0.484782	0.557511	0.552218	1.124330
min	NaN	NaN	NaN	NaN	NaN
25%	0.992647	0.889601	NaN	0.905726	1.000000
50%	1.021680	0.902937	0.927145	0.930760	1.150943
75%	1.176682	1.055286	1.031559	1.043398	1.161290
max	0.959492	0.097672	0.174312	0.255469	0.744556

	PropertyGFATotal	PropertyGFAParking	PropertyGFABuilding(s)	LargestPropertyUseTypeGFA	SecondLargestPropertyUseTypeGFA	ThirdLargestPropertyUseTypeGFA	YearBuilt	Latitude	Longitude
count	0.250471	0.250471	0.250471	0.250471	0.250471	0.250471	0.250471	0.250471	0.250471
mean	0.892477	0.883479	0.893428	0.893794	0.915982	0.837631	0.997896	0.999935	0.999996
std	0.682259	0.969751	0.613650	0.619939	1.021248	0.751259	1.001881	0.982927	1.014752
min	1.060523	NaN	0.479268	0.963215	NaN	NaN	1.000000	1.000312	0.999999
25%	0.933289	NaN	0.944709	0.948817	NaN	NaN	0.993302	0.999994	1.000008
50%	0.891001	NaN	0.883726	0.878423	NaN	NaN	0.997463	0.999980	0.999994
75%	0.837123	NaN	0.843251	0.868660	0.873539	0.000000	0.996992	0.999939	0.999981
max	0.172269	0.760542	0.130440	0.141036	0.888558	0.439597	1.000000	0.999968	1.000029

# ANNEXE : *train set, test set (2).* Séparation du *data set* en jeux d'entraînement et de test.

- Les *features* qualitatives conservent les mêmes catégories les plus fréquentes [*top*].
- ⚠ En revanche, le *test set* contient moins de catégories que le *train set* [*unique*].

## train set

	BuildingType	PrimaryPropertyType	LargestPropertyUseType	SecondLargestPropertyUseType	ThirdLargestPropertyUseType	Neighborhood	
<b>count</b>	3182	3182	3182	3182	3182	3182	3182
<b>unique</b>	7	27	48	45	42	13	
<b>top</b>	NonResidential	Low-Rise Multifamily	Multifamily Housing	None	None	DOWNTOWN	
<b>freq</b>	1377	973	1614	1713	2302	540	

## test set

	BuildingType	PrimaryPropertyType	LargestPropertyUseType	SecondLargestPropertyUseType	ThirdLargestPropertyUseType	Neighborhood	
<b>count</b>	797	797	797	797	797	797	797
<b>unique</b>	7	26	32	27	24	13	
<b>top</b>	NonResidential	Low-Rise Multifamily	Multifamily Housing	None	None	DOWNTOWN	
<b>freq</b>	337	281	405	441	604	131	