SEATTLE

Anticiper les besoins en énergie

Rappel

Le but de cette analyse est d'essayer de prédire la consommation d'énergie et les Émissions CO_2 afin d'éviter les relevés jugés trop coûteux.

Nous expliquerons nos choix éventuels pour arriver à une conclusion

Sommaire

Présentation des données

Csv de 2015

Il y a 3340 bâtiments différents

Colonnes dans données 2015 mais pas 2016

```
['Location',
 'OtherFuelUse(kBtu)',
 'GHGEmissions(MetricTonsCO2e)',
 'GHGEmissionsIntensity(kgCO2e/ft2)',
 'Comment',
 '2010 Census Tracts',
 'Seattle Police Department Micro Community Policing Plan Areas'
 'City Council Districts',
 'SPD Beats',
 'Zip Codes']
```

Csv de 2016

Il y a 3376 bâtiments différents

Colonnes dans données 2016 mais pas dans 2015

```
['Address',
 'City',
 'State',
 'ZipCode',
 'Latitude',
 'Longitude',
 'Comments',
 'TotalGHGEmissions',
 'GHGEmissionsIntensity']
```

La colonne ‘Location’ des données de 2015

dictionnaires → division → Renommer les colonnes

```
0       {'latitude': '47.61219025', 'longitude': '-122...
1       {'latitude': '47.61310583', 'longitude': '-122...
2       {'latitude': '47.61334897', 'longitude': '-122...
3       {'latitude': '47.61421585', 'longitude': '-122...
4       {'latitude': '47.6137544', 'longitude': '-122....
                              ...
3335    {'latitude': '47.59950256', 'longitude': '-122...
3336    {'latitude': '47.65752471', 'longitude': '-122...
3337    {'latitude': '47.61649845', 'longitude': '-122...
3338    {'latitude': '47.68396954', 'longitude': '-122...
3339    {'latitude': '47.68396954', 'longitude': '-122...
Name: Location, Length: 3340, dtype: object
```

18081	47.61219025	-122.33799744	405 OLIVE WAY	SEATTLE	WA	98101
18081	47.61310583	-122.33335756	724 PINE ST	SEATTLE	WA	98101
18081	47.61334897	-122.33769944	1900 5TH AVE	SEATTLE	WA	98101
18081	47.61421585	-122.33660889	620 STEWART ST	SEATTLE	WA	98101
19576	47.6137544	-122.3409238	401 LENORA ST	SEATTLE	WA	98121
...	...	...	...	...	...	...
18379	47.59950256	-122.32034302	321 10TH AVE S	SEATTLE	WA	98104
18383	47.65752471	-122.3160159	4123 12TH AVE NE	SEATTLE	WA	98105

```
.rename(columns={'latitude':'Latitude', 'longitude':'Longitude',
                 'address':'Address', 'city':'City',
                 'state':'State', 'zip':'ZipCode'})
```

Après ces modifications

données de 2015

```
['OtherFuelUse(kBtu)',
 'GHGEmissions(MetricTonsCO2e)',
 'GHGEmissionsIntensity(kgCO2e/ft2)',
 'Comment',
 '2010 Census Tracts',
 'Seattle Police Department Micro Community Policing Plan Areas'
 'City Council Districts',
 'SPD Beats',
 'Zip Codes']
```

données de 2016

```
['Comments', 'TotalGHGEmissions', 'GHGEmissionsIntensity']
```

Après une rapide analyse, je constate que les colonnes TotalGHGEmissions et GHGEmissionsIntensity sont identique avec GHGEmissions(MetricTonsCO2e) et GHEmissionsIntensity(kgCO2e/ft2), je renomme ces 2 colonnes à l'identique ainsi que Comments et Comment.

Pour le reste des colonnes, je les supprimes puisqu'elles ne sont pas dans les données de 2016

Après une jointure de nos 2 csv, on effectuons un nettoyage de nos données.

- Tout d'abord, nous sélectionnons les bâtiment non résidentiel. En effet, selon l'énoncé :
  'seul les bâtiments non résidentiel seront analysés'

ComplianceStatus	3318 non-null	object
Outlier	48 non-null	object
Latitude	3318 non-null	object
Longitude	3318 non-null	object

  Longitude et Latitude sont converties en float
  ComplianceStatus est convertie en int64

- Les colonnes electricity(kBtu) et electricity(Kwh) sont redondante, je supprime electricity(Kwh)
- De même pour naturalGas(kBtu) et naturalGas(therm), je supprime naturalGas(therm)
- Le suffixe EUIWN signifie, Energy Use Intensity Weather Normalize

les colonnes SiteEUI(kBtu/sf), SourceEUI(kBtu/sf), SiteEnergyUse(kBtu) sont avec et sans le suffixe WN. Je trouve intéressant de voir l'impact de la météo sur les bâtiments et garde les données WN

Dans le cadre de nos modélisations, les variables à prédire sont la consommation d'énergie du bâtiment (SiteEnergyUse(kBtu)) et ses émissions de CO2 (TotalGHGEmissions). Certaines lignes comportent des manquants sur ces variables, nous allons donc les supprimer :

Le nombre de bâtiments dont TotalGHGEmissions est manquant est de : 9

Le nombre de bâtiments dont TotalGHGEmissions est <=0 est de : 1

Le nombre de bâtiments dont SiteEnergyUseWN(kBtu) est manquant est de : 1

Le nombre de bâtiments dont SiteEnergyUseWN(kBtu) est <=0 est de : 0

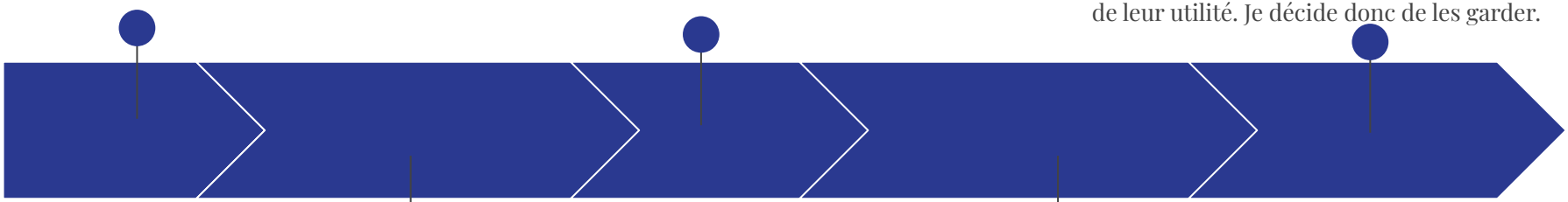
Suppression des colonnes avec beaucoup d'informations manquantes :

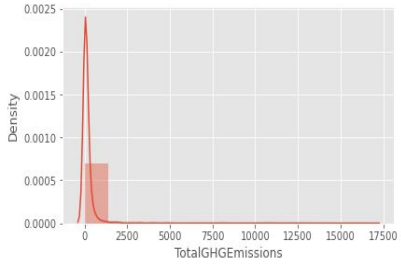
columns={'Comments', 'Outlier'}

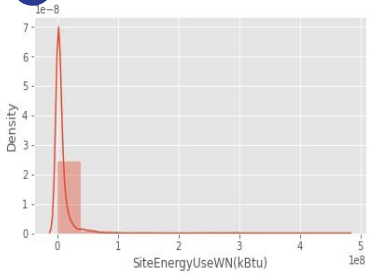
Regard sur les variables à prédire

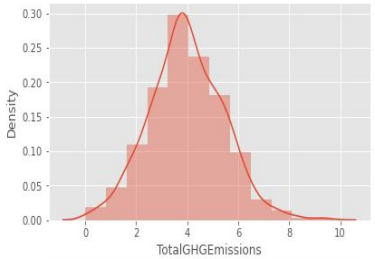
Transformation logarithmique, Elle permet en général de rapprocher des valeurs extrêmes pour obtenir des graphes de distribution moins étendus

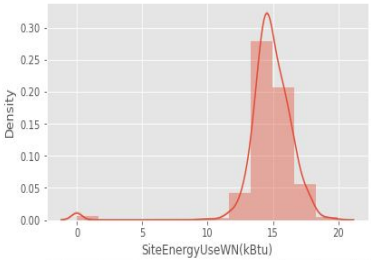
Maintenant que nos distributions suivent une loi normale, nous pouvons constater que certains bâtiments ont de forte Emissions et consommation d'énergie, tandis que d'autre, c'est l'inverse ! Cela vient peut être de l'âge des bâtiments, de leur matériaux de construction et de leur utilité. Je décide donc de les garder.

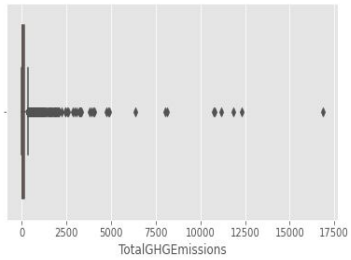


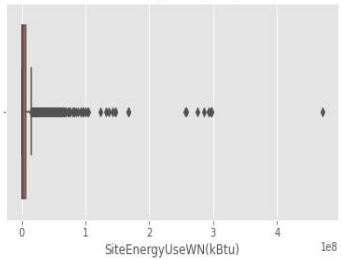


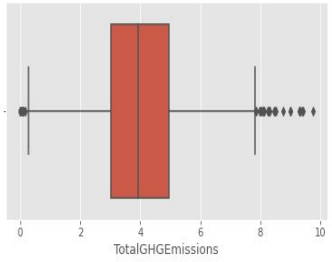


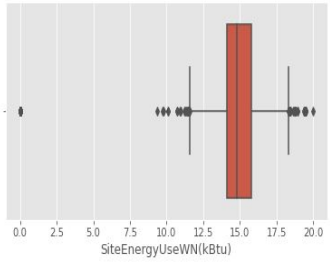




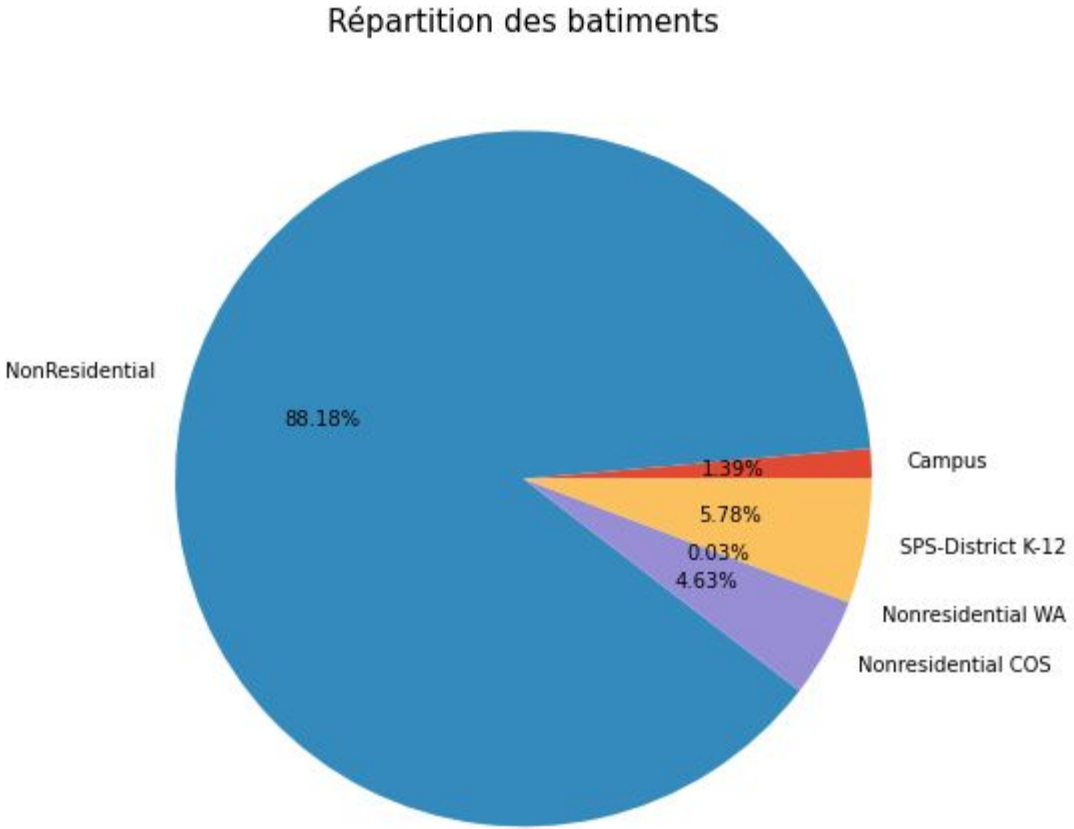








Analyse et Features Engineering



Les bâtiments non résidentiel occupent une grande partie de nos données

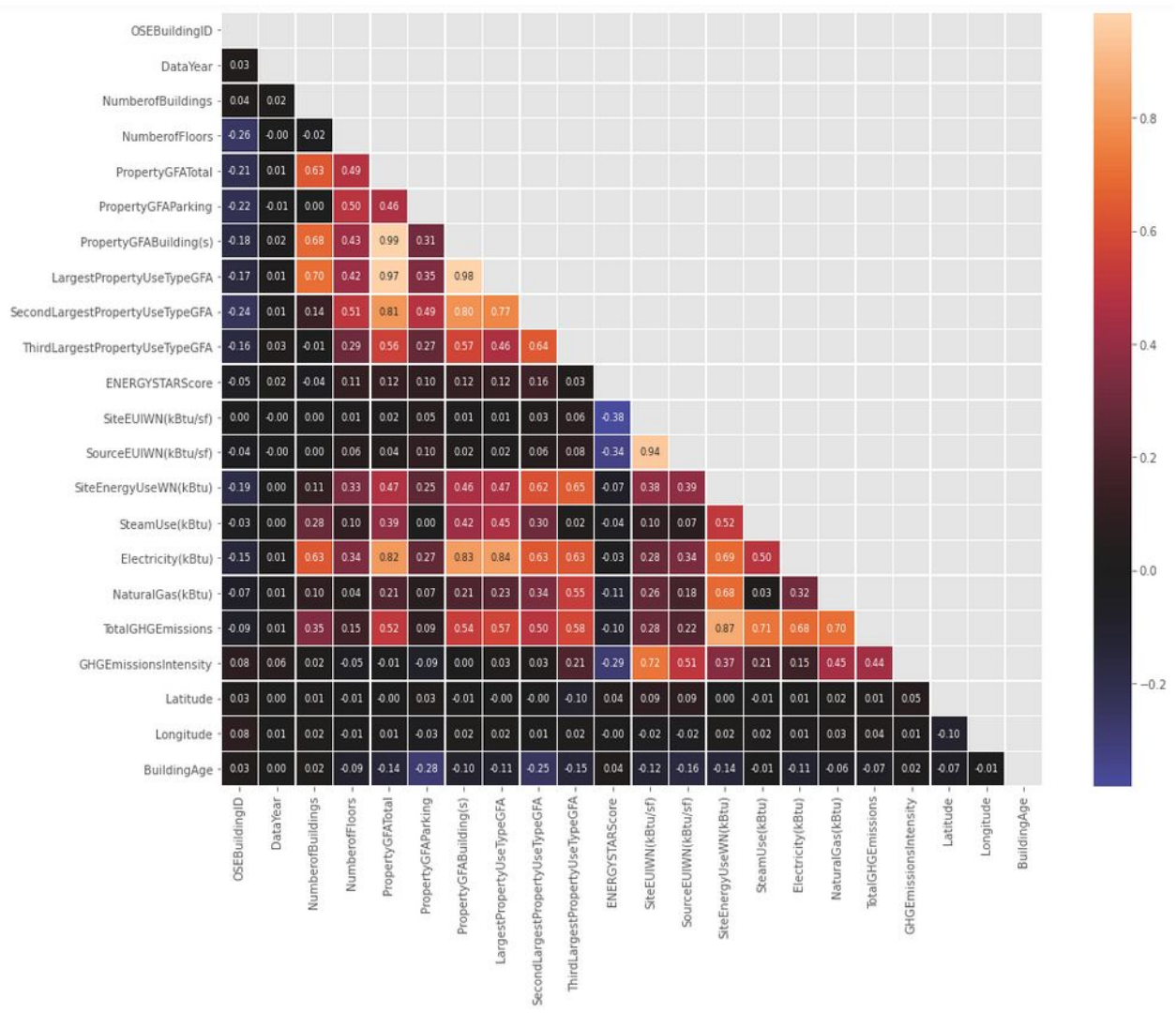
```
Small- and Mid-Sized Office    578
Other                          380
Large Office                   334
Mixed Use Property             200
Retail Store                   191
Non-Refrigerated Warehouse     181
Warehouse                      180
Hotel                          149
Worship Facility               143
Medical Office                  80
K-12 School                     78
Distribution Center             53
Distribution Center\n           49
Supermarket / Grocery Store     40
Senior Care Community           39
Supermarket/Grocery Store       36
Self-Storage Facility           29
Self-Storage Facility\n         27
Refrigerated Warehouse          25
Residence Hall                  21
Hospital                        20
University                      17
College/University              16
Residence Hall/Dormitory        15
Restaurant                      12
Laboratory                      11
Restaurant\n                     9
Low-Rise Multifamily             3
Name: PrimaryPropertyType, dtype: i
```

Après un bref coup d'oeil sur le type de bâtiment, on peut remarquer certains doublons avec le suffixe \n, et un défaut d' écriture sur Supermarket / Grocery Store.

Nous remplaçons toutes ces données.

nous faisons la même chose avec la colonnes 'Neighborhood'.

```
['DOWNTOWN',
 'SOUTHEAST',
 'NORTHEAST',
 'EAST',
 'CENTRAL',
 'NORTH',
 'MAGNOLIA / QUEEN ANNE',
 'LAKE UNION',
 'GREATER DUWAMISH',
 'BALLARD',
 'NORTHWEST',
 'SOUTHWEST',
 'DELRIDGE',
 'Central',
 'Ballard',
 'North',
 'Delridge',
 'Northwest',
 'DELRIDGE NEIGHBORHOODS']
```



Pour les variables à prédire TotalGHGEmissions et SiteEnergyUse(kBtu), on remarque des corrélations linéaires quasi similaires avec les variables de relevés (les consommations) mais également avec le nombre de bâtiments ou d'étages ainsi que les surfaces au sol.

On remarque sur ce Heatmap de fortes corrélations linéaires entre variables. Ces corrélations peuvent amener des problèmes de colinéarité dans nos futurs modèles.

	level_0	level_1	corr_coeff
26	PropertyGFATotal	PropertyGFABuilding(s)	0.986318
24	LargestPropertyUseTypeGFA	PropertyGFABuilding(s)	0.976089
22	LargestPropertyUseTypeGFA	PropertyGFATotal	0.971510
20	SourceEUIWN(kBtu/sf)	SiteEUIWN(kBtu/sf)	0.944694
18	TotalGHGEmissions	SiteEnergyUseWN(kBtu)	0.868615
16	Electricity(kBtu)	LargestPropertyUseTypeGFA	0.836142
14	Electricity(kBtu)	PropertyGFABuilding(s)	0.825159
12	PropertyGFATotal	Electricity(kBtu)	0.815379
10	SecondLargestPropertyUseTypeGFA	PropertyGFATotal	0.811361
8	SecondLargestPropertyUseTypeGFA	PropertyGFABuilding(s)	0.797458
6	LargestPropertyUseTypeGFA	SecondLargestPropertyUseTypeGFA	0.766965
4	GHGEmissionsIntensity	SiteEUIWN(kBtu/sf)	0.724120
2	SteamUse(kBtu)	TotalGHGEmissions	0.714102
0	TotalGHGEmissions	NaturalGas(kBtu)	0.704136

En isolant, les paires de variables avec des corrélations de Pearson supérieurs à 0.7, On remarque que les variables suffixées GFA présentent de fortes corrélations avec plusieurs autres variables. Nous allons donc créer de nouvelles variables pour tenter de gommer ces corrélations linéaires.

Nous allons donc créer une variable nous donnant le nombre total d'usage du bâtiment, puis supprimer la liste complète des usages.

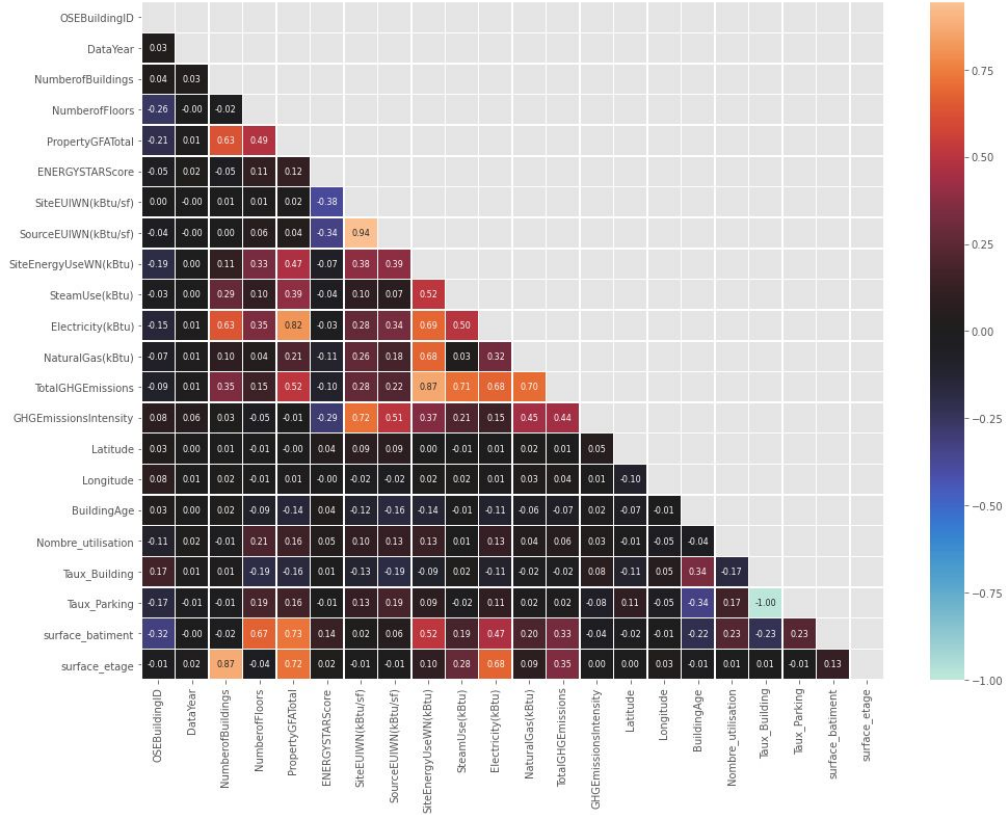
- result['Nombre_utilisation']
- result['Taux_Building']
- result['Taux_Parking']

Et nous supprimons ensuite les variables inutiles:

```
drop(['LargestPropertyUseTypeGFA',
    'SecondLargestPropertyUseTypeGFA'
    'SecondLargestPropertyUseType',
    'ThirdLargestPropertyUseTypeGFA',
    'ThirdLargestPropertyUseType',
    'PropertyGFAParking',
    'PropertyGFABuilding(s)'],
    'ListOfAllPropertyUseTypes'
```

Nous créons 2 autres variables afin de calculer la surface moyenne par bâtiment et par étages

```
result['surface_batiment'] = round((result['PropertyGFATotal'] / result['NumberofBuildings']),2)
result['surface_etage'] = round((result['PropertyGFATotal'] / result['NumberofFloors']),2)
```



Maintenant que nos données sont complétés, regardons l'impact du features engineering.

Certaines variables sont corrélées entre elles, mais plus de corrélations linéaire.

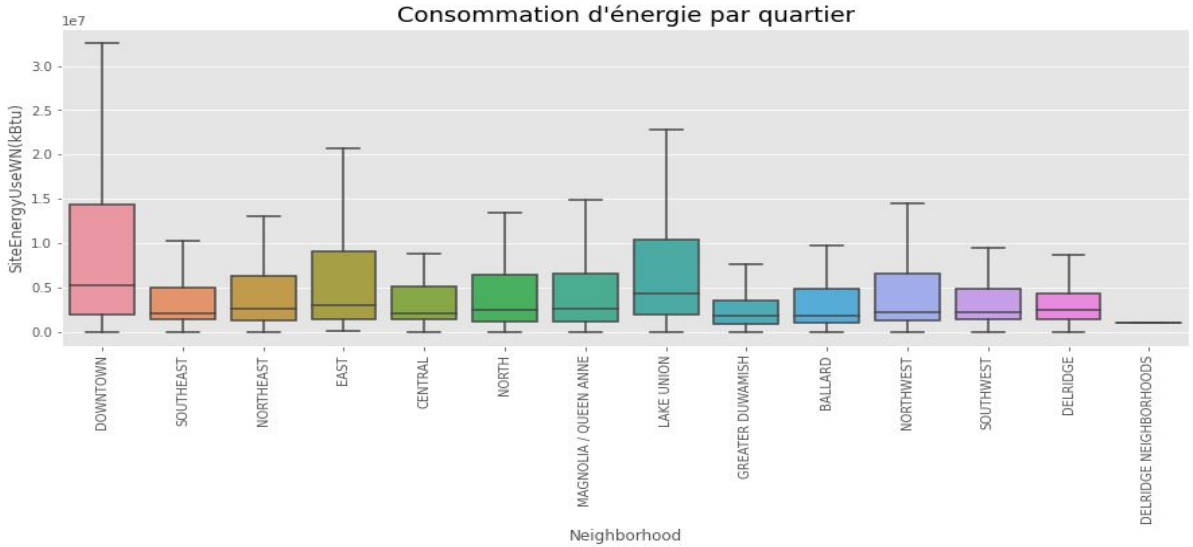
Analyse du VIF : (Variance Inflation Factor) signifie Facteur d'Inflation de la Variance. Au cours de l'analyse de régression, VIF évalue si les facteurs sont corrélés les uns aux autres (multi-colinéarité), ce qui pourrait influencer les autres facteurs et réduire la fiabilité du modèle.Si un VIF est supérieur à 10, vous avez une multi-colinéarité élevée : la variation semblera plus grande et le facteur apparaîtra plus influent qu'il ne l'est. Si VIF est plus proche de 1, alors le modèle est beaucoup plus robuste, car les facteurs ne sont pas influencés par la corrélation avec d'autres facteurs

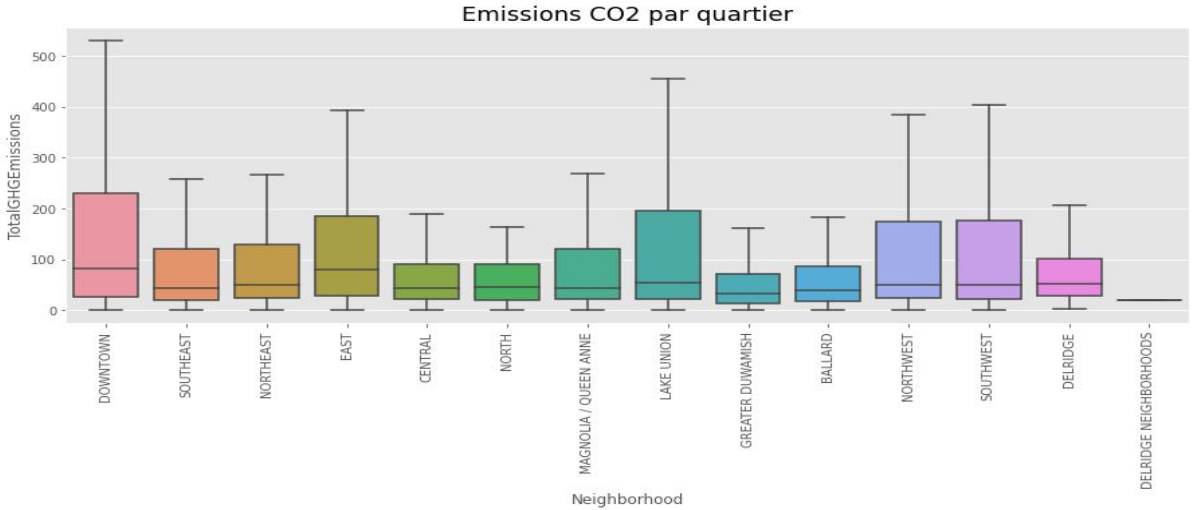
	feature	VIF
2	NumberofBuildings	1.102119e+01
4	PropertyGFATotal	6.110074e+01
5	SiteEUIWN(kBtu/sf)	4.286249e+01
6	SourceEUIWN(kBtu/sf)	2.912880e+01
7	SiteEnergyUseWN(kBtu)	3.119434e+01
8	SteamUse(kBtu)	1.371107e+10
9	Electricity(kBtu)	1.240752e+09
10	NaturalGas(kBtu)	1.482335e+10
11	TotalGHGEmissions	3.741432e+10
17	Taux_Building	3.869803e+07
18	Taux_Parking	1.105176e+06
19	surface_batiment	2.909072e+01
20	surface_etage	1.484896e+01

Des scores VIF supérieur à 10 indiquent généralement une forte multicolinéarité. Ces variables fortement corrélées risquent d'impacter nos modèles. Les features suffixées EUIWN(kBtu/sf), sont des variables dont les valeurs sont ramenées à la surface par étage. Nous allons les supprimer car nous avons créer des variables pouvant permettre de ramener nos données à l'étage ou au building. Idem pour la variable GHGEmissionsIntensity.

Donc, suppression des colonnes :

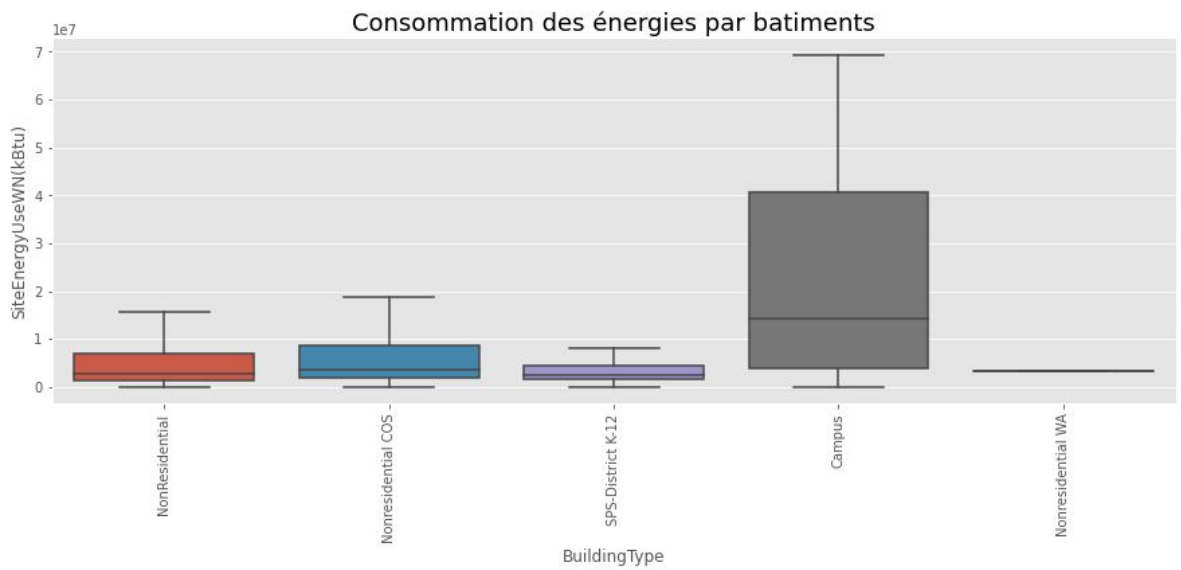
- SiteEUIWN(kBtu/sf)
- SourceEUIWN(kBtu/sf)
- GHGEmissionsIntensity

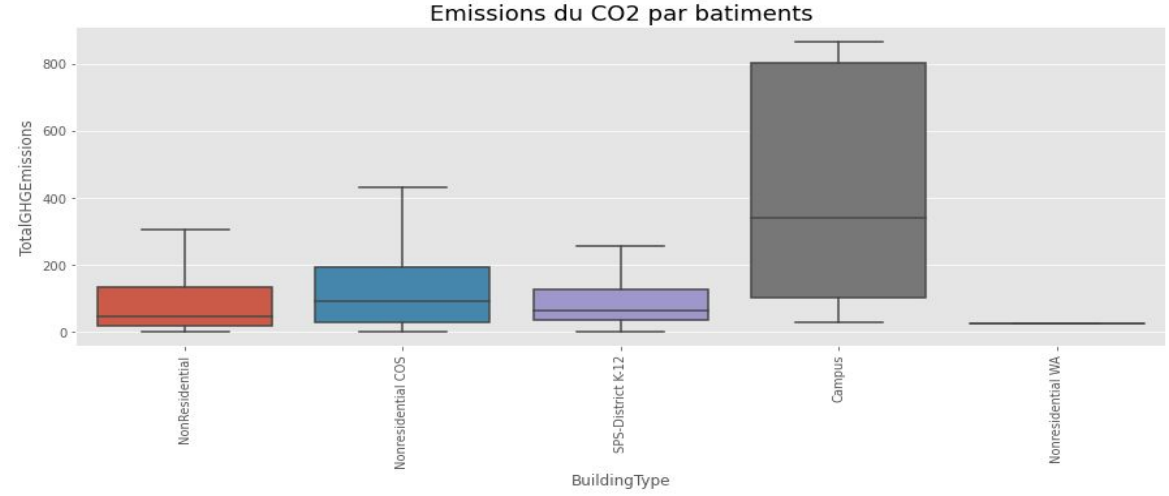




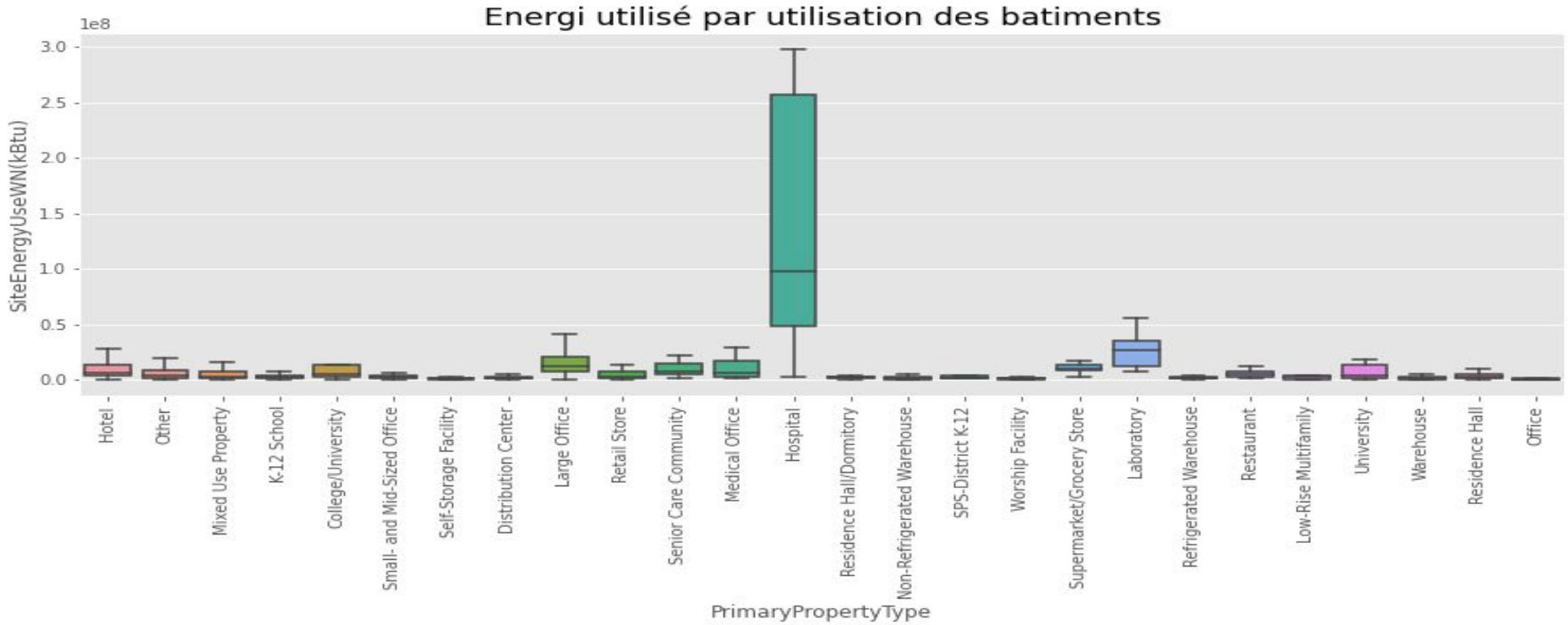
Concernant la consommation et les émissions de CO_2, le quartier des ‘downtown’ se détache du lot en ayant des relevés élevés.

Les downtown ayant la plus forte concentration de hauts buildings.





D'un point de vue des bâtiments, ce sont les campus qui ont la plus forte consommations d'énergie et d' émission de CO_2.



Nous pouvons déjà constater que l'on peut fusionner les types University et College/University, ainsi que SPS-District K-12 et K-12 School.

On remarque que les hôpitaux sont de très gros consommateurs d'énergie

Et pour finir, je supprime les colonnes qui ne me seront d'aucunes utilités pour les prédictions.

```
drop(columns={'DataYear',
              'DefaultData',
              'ComplianceStatus',
              'City',
              'State',
              'ZipCode',
              'Address',
              'TaxParcelIdentificationNumber',
              'CouncilDistrictCode',
              'YearsENERGYSTARCertified',
              'Latitude',
              'Longitude'})
```

Analyse du modèle

```
drop(columns={'SteamUse(kBtu)',
              'Electricity(kBtu)',
              'NaturalGas(kBtu)'})
```

Le but de notre programme est de supprimer les relevés coûteux pour les années à venir. Nous allons donc exclure toutes les données de relève de notre dataset.

```
1 variables_cat.nunique()
```

executed in 21ms, finished 08:38:35 2022-04-14

```
BuildingType              5
PrimaryPropertyType      24
PropertyName           3194
Neighborhood             14
LargestPropertyUseType   58
dtype: int64
```

Sur les variables ‘catégories’, le nom des bâtiments ne sera pas utile.

```
1 variables_num.nunique()
```

executed in 20ms, finished 08:38:36 2022-04-14

```
OSEBuildingID           1697
NumberofBuildings         17
NumberofFloors            44
PropertyGFATotal        1666
ENERGYSTARScore          100
SiteEnergyUseWN(kBtu)   3271
TotalGHGEmissions       3012
BuildingAge              115
Nombre_utilisation        11
Taux_Building            380
Taux_Parking             380
surface_batiment        1714
surface_etage           1705
dtype: int64
```

Il peut être intéressant de savoir si le EnergyStarScore a une influence sur les émission et l'energy utilisés. Je garde ces données, mais je les mets de côté pour ne pas influencer les prédictions

Sur les variables catégories, nous avons utilisé un Encoder afin de transformer ces variables en variables numériques

Pour cela, nous avons employé le OneHotEncoder de la librairie Sklearn,

qui consiste à encoder une variable à *n* états sur *n* bits dont un seul prend la valeur 1, le numéro du bit valant 1 étant le numéro de l'état pris par la variable.

Nous avons effectué un encodage sur les variables 'BuildingType', 'Neighborhood' et 'PrimaryPropertyType' afin de les intégrer dans notre analyse.

Ensuite, étape crucial, c'est la standardisation de nos données qui nous donne de meilleurs résultats lors de nos prédictions.

La standardisation des données a pour objectif d'assurer une compatibilité optimale des données, en vue de leur réutilisation. L'application d'une « commune mesure » permet d'améliorer la qualité des données.

Pour cela, j'ai utilisé RobustScaler de la librairie SKLearn, qui a une façon optimal de gérer les valeurs extrêmes.

Passons ensuite à la séparation de nos données, qui consiste à diviser nos données en 2 parties.

Un train_set et un test_set.

Le but est d'entraîner nos modèles sur un jeu d'entrainement de nos données(train_set), en général environ 75% de nos données.

Et ensuite, tester nos modèles sélectionnés sur des données que notre algorithme n'a jamais vu(test_set).

Ces étapes étant faites, nous pouvons commencer à entraîner différents modèles

Afin de déterminer la précision de nos modèles, nous allons utiliser plusieurs paramètres, appelés :

métric

la MAE : Mean Absolut Error ou l'erreur absolue moyenne est une mesure des erreurs entre des observations adaptées exprimant le même phénomène.

Le R^2 : le coefficient de détermination linéaire de Pearson, noté $\boldsymbol{R^2}$ ou $\boldsymbol{r^2}$, est une mesure de la qualité de la prédiction d'une régression linéaire.

le temps d'entraînement : Correspond au temps qu'il faut à l'algorithme pour faire la prédiction.

Il y aura aussi : meilleurs paramètres

Sur tous les modèles, l'algorithme GridSearchCV sera appliqué pour optimiser tous les paramètres de tous modèles sélectionnés.

Meilleurs paramètres affichera les meilleurs paramètres du modèles en cours

Afin de faire nos prédictions sur TotalGHGEmissions et SiteEnergyUseWN, nous allons utiliser plusieurs algorithmes pour nos résultats.

Nous choisissons dans la famille de l’apprentissage supervisé, qui est une tâche d'apprentissage automatique consistant à apprendre une fonction de prédiction à partir d'exemples annotés, au contraire de l'apprentissage non supervisé. On distingue les problèmes de régression et des problèmes de classement.

Random Forest :

La forêt aléatoire est un algorithme d'apprentissage automatique supervisé largement utilisé dans les problèmes de classification et de régression. Il construit des arbres de décision sur différents échantillons et prend la moyenne en cas de régression.

KNeighbors :

la **méthode des *k* plus proches voisins** est une méthode d'apprentissage supervisé. Pour estimer la sortie associée à une nouvelle entrée x, la méthode des k plus proches voisins consiste à prendre en compte (de façon identique) les k échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelle entrée x, selon une distance à définir.

ElasticNet :

La régression linéaire nette élastique utilise les pénalités des techniques de lasso et de crête pour régulariser les modèles de régression. La technique combine à la fois les méthodes de régression lasso et ridge en apprenant de leurs erreurs pour améliorer la régularisation des modèles statistiques.

Résultat ??

La Régression Linéaire :

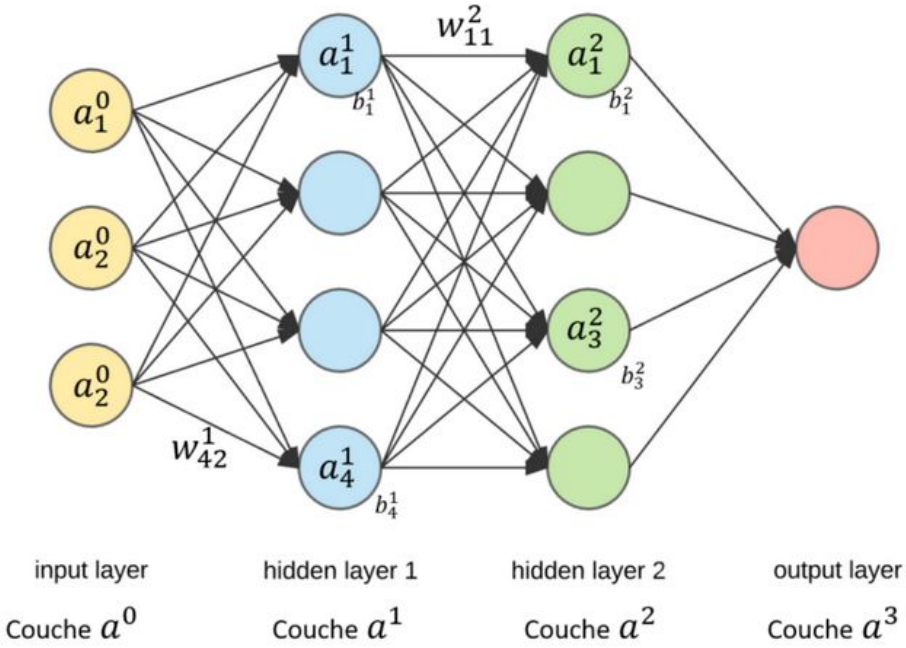
Un modèle de régression linéaire est un modèle de régression qui cherche à établir une relation linéaire entre une variable, dite expliquée, et une ou plusieurs variables, dites explicatives

MLP Regressor :

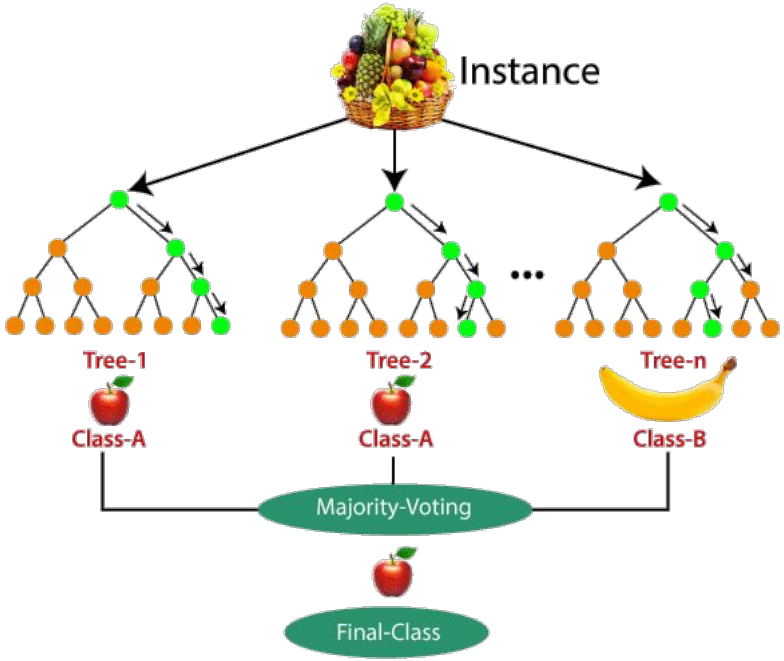
Le **perceptron multicouche** *(multilayer perceptron* MLP) est un type de réseau neuronal artificiel organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement ; il s'agit donc d'un réseau à propagation directe (*feedforward*). Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche (dite « de sortie ») étant les sorties du système global.

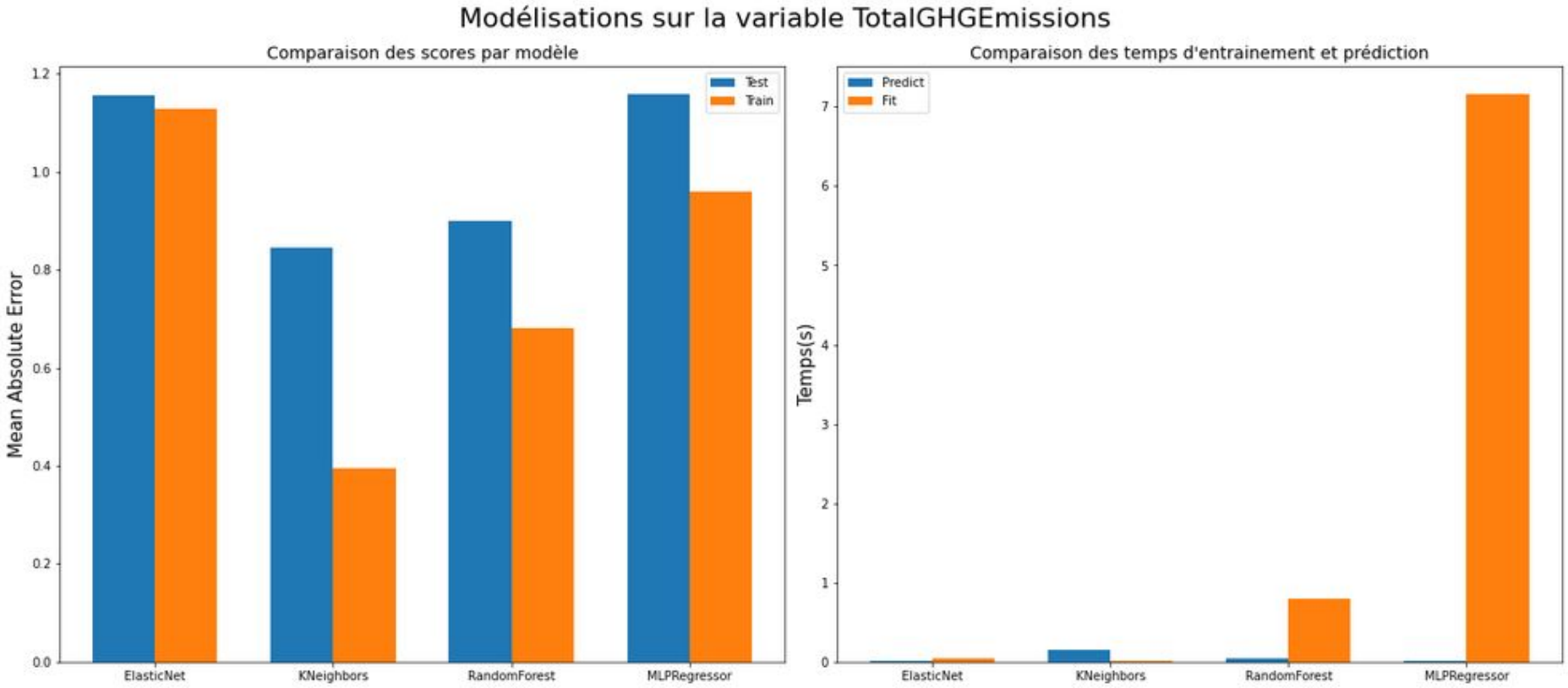
Pour plus de précision

MLP Regressor

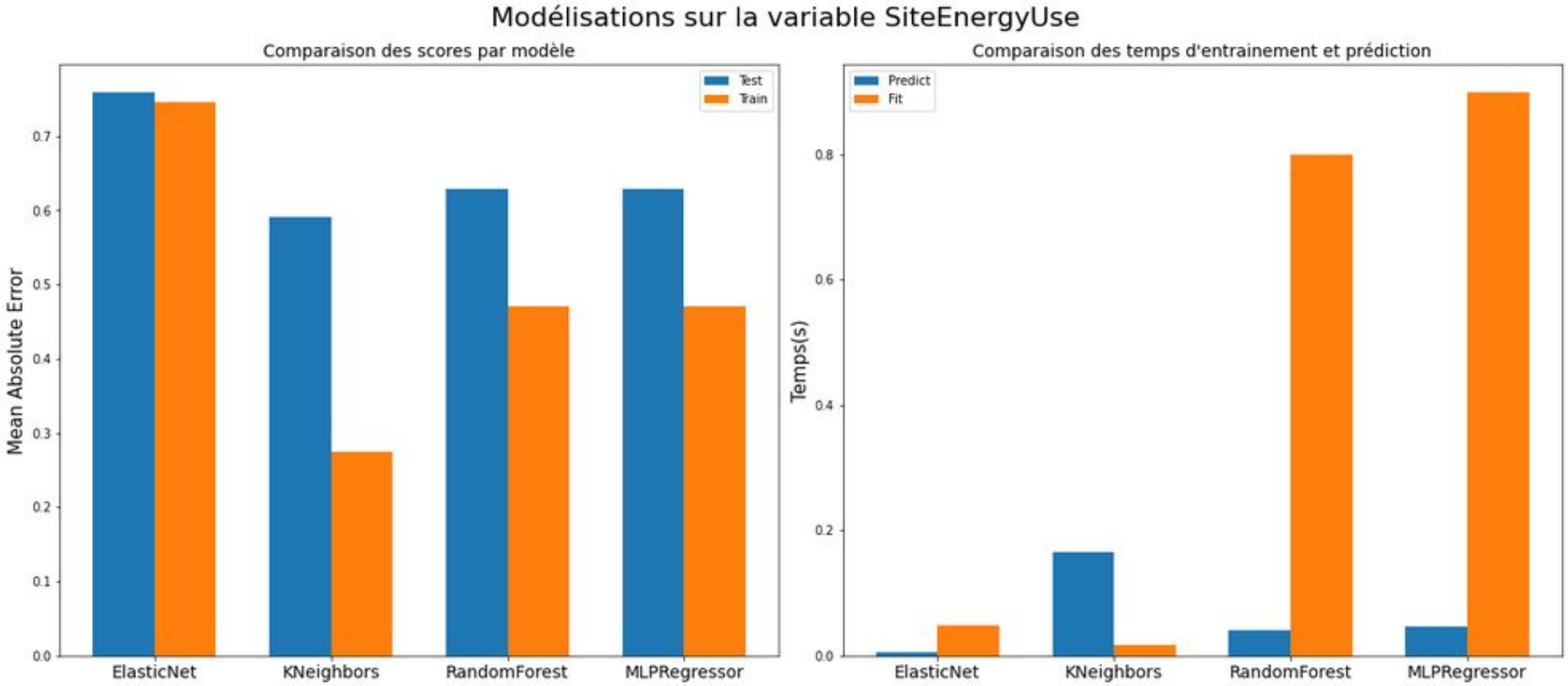


Random Forest





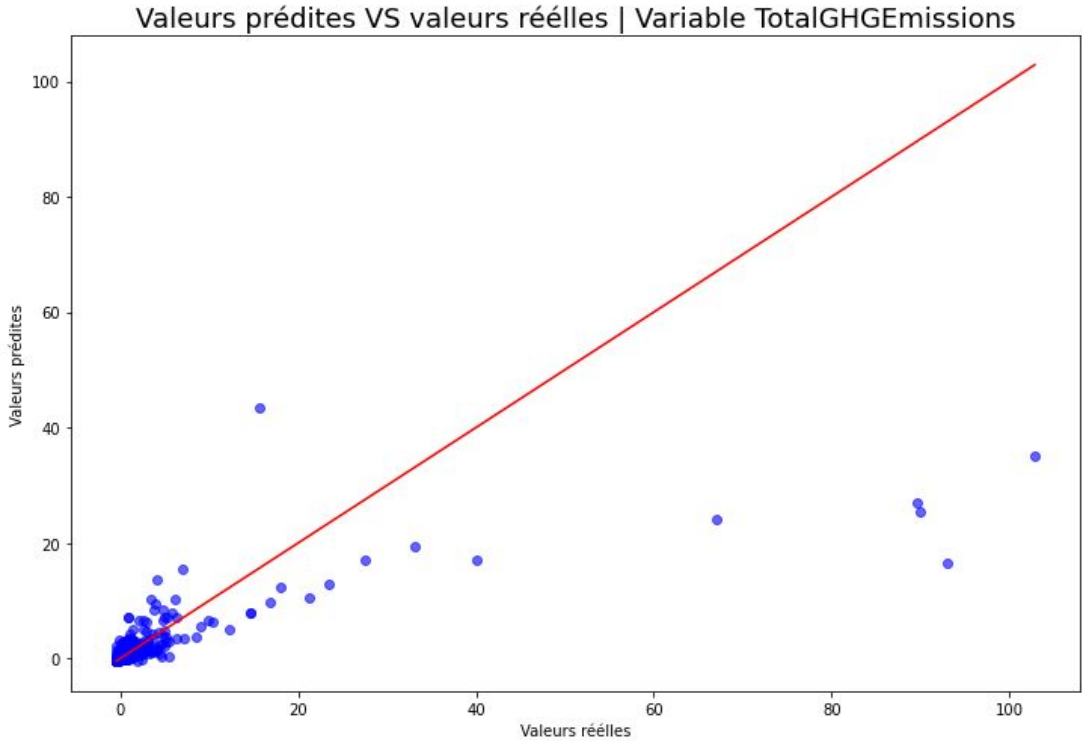
Conclusion : Le modèle RandomForest offre un score MAE bas par rapport aux autres, avec un temps d'entraînement et un R^2 bon. Ce modèle est choisi pour TotalGHGEmissions.



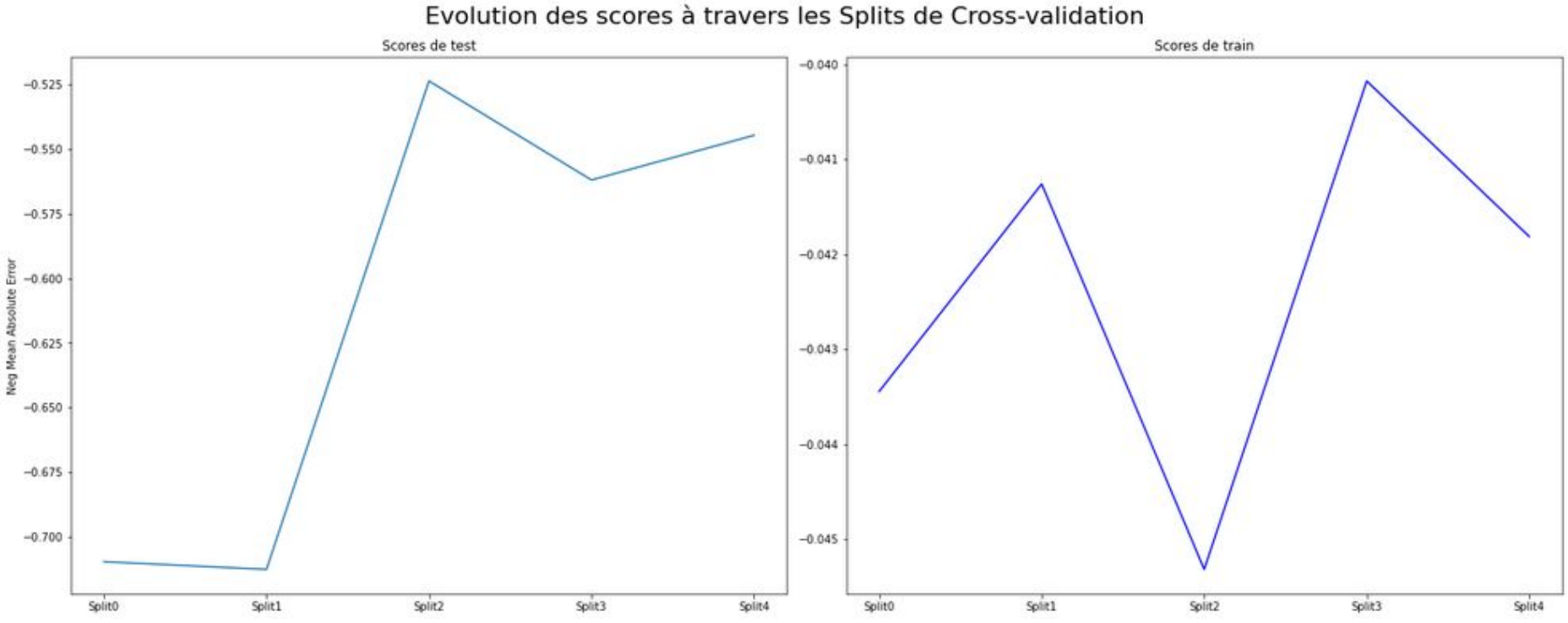
Conclusion : idem pour SiteEnergyUse, le modèle RandomForest est choisi pour les mêmes raisons

Vérifions maintenant le modèle choisi sur notre jeu de test, jeu que notre algorithme n'a encore jamais vu

	Métrique	Résultats
0	MAE	1.272185
1	R^2	0.489218



Les résultats se sont dégradées sur notre jeu de test par rapport à ceux obtenu avant

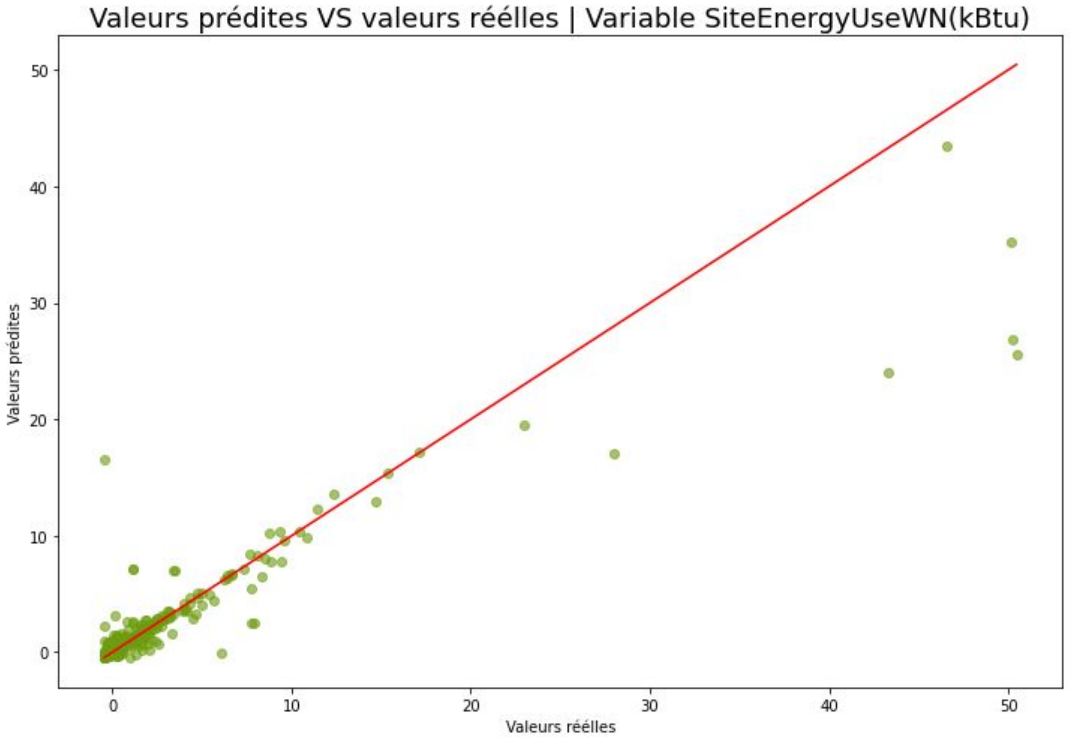


On voit ici que les scores des différents splits de cross-validation, pour les meilleurs paramètres obtenus, évoluent correctement lors de l'entraînement et des test, tout en restant dans la même échelle.

Les écarts et mauvais résultats obtenus dépendent donc du faible nombre de données qui impactent le Train_Test_Split initial. Le modèle est correctement entraîné mais n'obtient pas de bon résultats sur le jeu de test

Essayons maintenant pour SiteEnergyUseWN

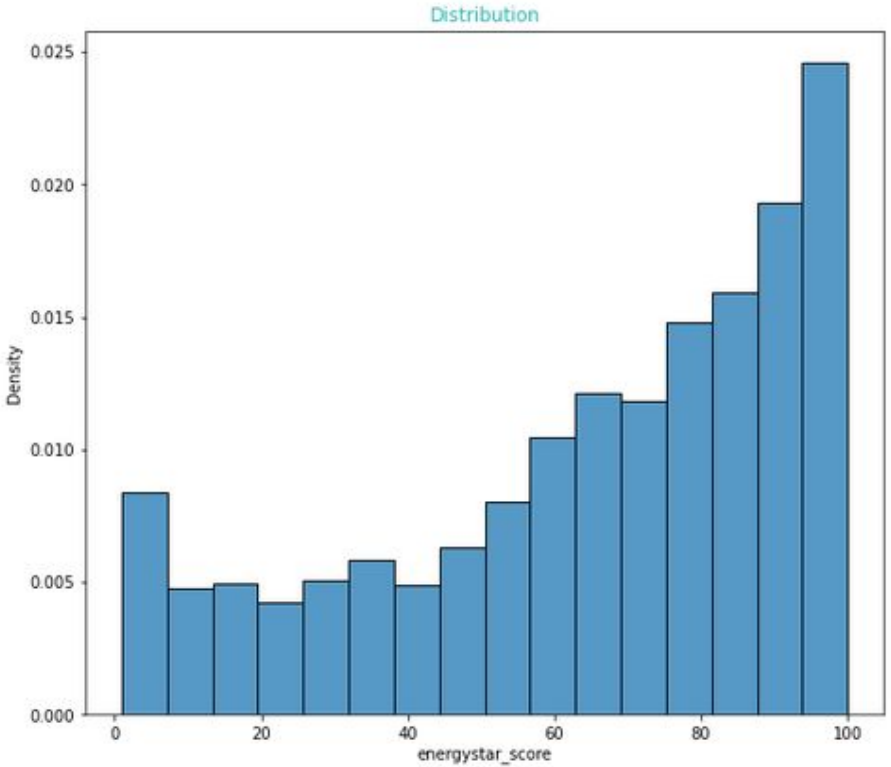
	Métrique	Résultats
0	MAE	0.445837
1	R^2	0.835516

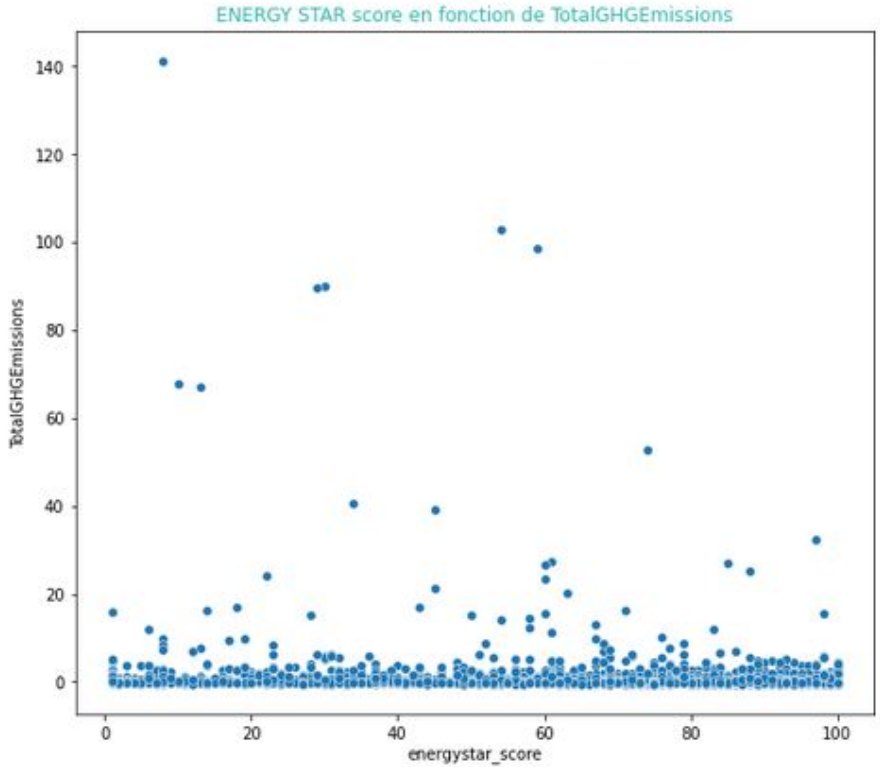


Contrairement à TotalGHGEmissions, les résultats se sont améliorés, cependant, ils ne restent pas assez bon pour prédire ces résultats

Influence de l’EnergyStar

Analyse de la variable ENERGY STAR Score





De nombreux batiments sont au dessus de 60 en energyStar ce qui indique que ces batiments sont bien classés(donc un faible taux), en emissions de CO2.

Nous faisons exactement la même chose, Nous séparons les données, utilisons le modèle RandomForest avec les mêmes hyperparamètres.

Ce qui nous donne ce résultat

	Métrique	Sans ENERGY STAR	Avec ENERGY STAR
0	MAE	1.272185	0.734729
1	R^2	0.489218	0.725581

Conclusion

Le résultat du R^2 reste encore faible. On peut voir une amélioration du à l'EnergyStar, mais en raison du manque de données, il est difficile à dire si cette amélioration est significative.

Mon Ressenti

Projet vraiment intéressant dans la recherche et la compréhension de nouveaux algorithmes, travailler sur ce genre de projet nous montre à quel point, le métier de la data est passionnant aussi bien dans la recherche que dans l'évolution constant du métier.

Certain problème ont été rencontré notamment, définir une stratégie pour arriver à une conclusion, les nouveaux algorithmes et apprendre leur fonctionnement,
Et enfin beaucoup de nouveaux termes à comprendre donc beaucoup de recherches sur internet pour faire face et régler ces problèmes.