



Projet n°3 : Anticipez les besoins de consommation électrique des bâtiments

Parcours Ingénieur Machine Learning

Céline Mendola

Présentation de la problématique

- Notre but est de prédire les émissions de CO₂ et la consommation totale d'énergie des bâtiments non destinés à l'habitation de la ville de Seattle.

Pour cela, on dispose des relevés de 2015 et 2016, ainsi que des données déclaratives du permis d'exploitation commerciale des bâtiments.

- Après nettoyage, feature engineering et analyses des données, nous allons élaborer des modèles de machine learning linéaires, non linéaires et ensemblistes afin de faire nos prédictions.
- La donnée de l'energy STAR score, fastidieuse à calculer, est elle nécessaire pour effectuer de bonnes prédictions?



1. Data cleaning

Les Datasets

Variables principales dans les datasets de 2015 et 2016 :

- données de localisation : state, city, address, zipcode, longitude, latitude
- caractéristiques de la propriété: type, le nombre de bâtiments, d'étages, ses premier, second et troisième usage s'ils existent, la présence d'un parking, les superficies associées.
- Consommation annuelle des différentes sources d'énergie en électricité, gaz et géothermie.
- La consommation d'énergie annuelle totale du bâtiment ('SiteEnergyUse(kBtu)'), ainsi que depuis la source de production d'énergie et la consommation WN. On a aussi ces informations normalisées par square feet du bâtiment.
- Enfin, les émissions de gaz à effet de serre en tonnes de co2, nommé 'TotalGHGEmissions' et le ratio en square feet par rapport à la superficie du bâtiment.

Les deux datasets ont chacun plus de 97% de leurs données communes. Nous avons utilisé le dataset de 2016. Il contient 3376 observations.

Filtres sur les bâtiments non destinés à l'habitation

- Filtre sur la variable BuildingType

```
df=data_2016[data_2016['BuildingType'].isin(['NONRESIDENTIAL','NONRESIDENTIAL_COS','SPS_DISTRICT_K_12',\ 'CAMPUS','NONRESIDENTIAL_WA'])]
```

- Filtre sur la variable PrimaryPropertyType

```
df=df[df['PrimaryPropertyType']!='RESIDENCE_HALL']  
df=df[df['PrimaryPropertyType']!='LOW_RISE_MULTIFAMILY']
```

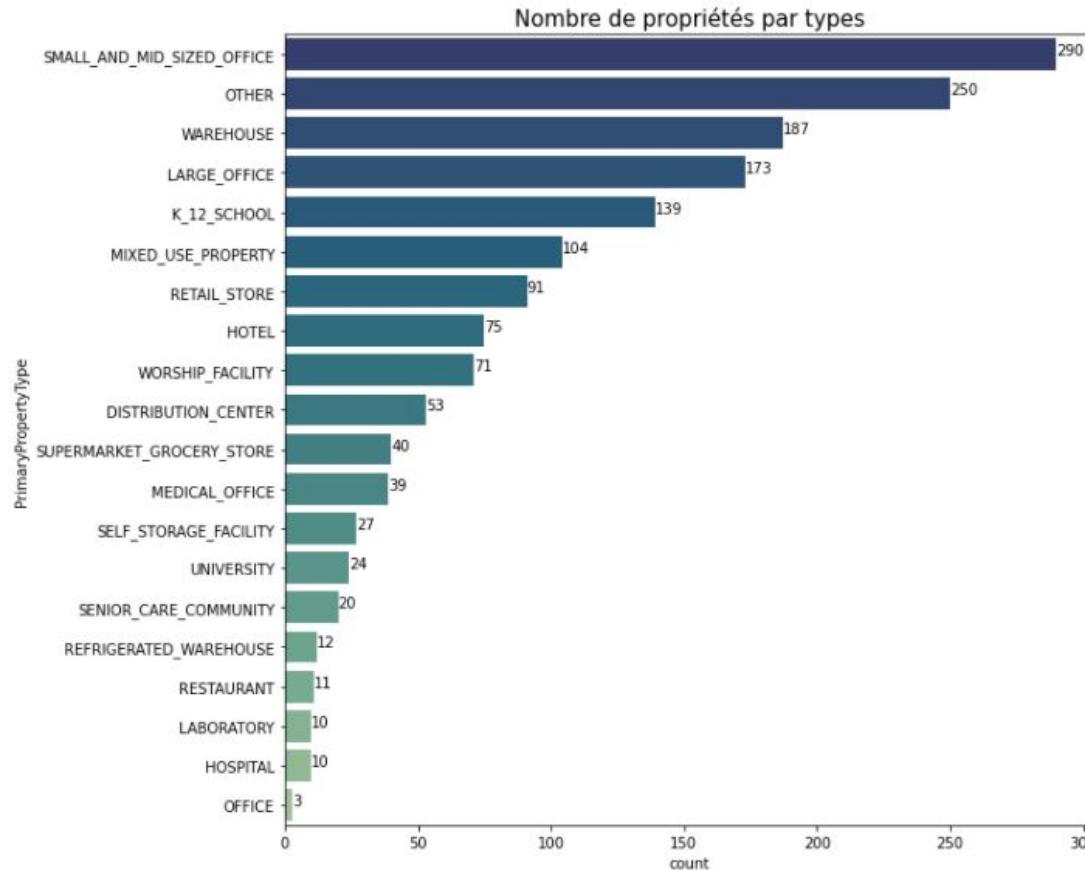
- Filtre sur la variable LargestpropertyUseType

```
df = df[df['LargestPropertyUseType'].str.contains('housing',case=False)==False]
```



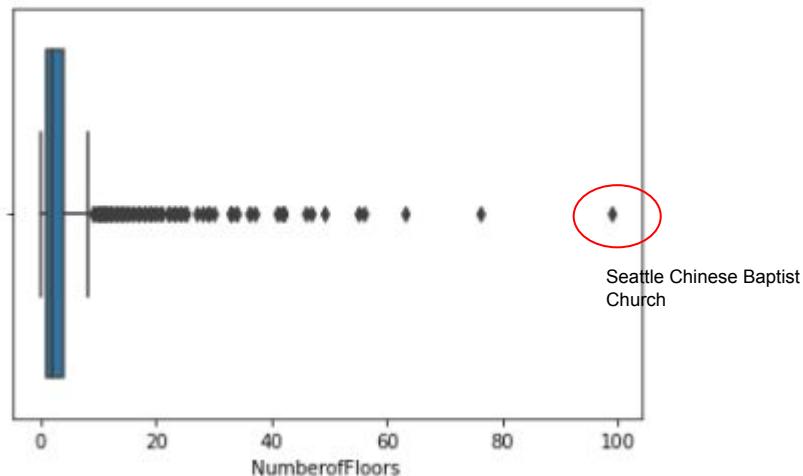
1629 observations restantes

Premières analyses



Premières analyses

Nombre d'étages



Nombre de bâtiments par propriété

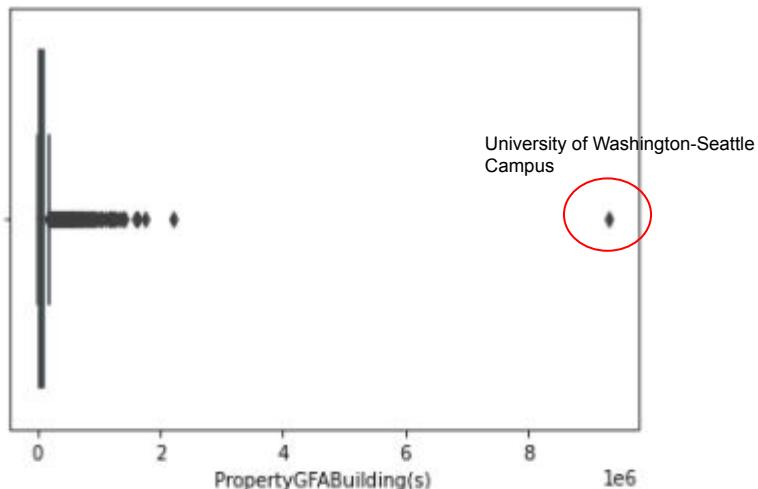
```
df['NumberOfBuildings'].describe()
```

```
count      1629.000000
mean       1.170043
std        2.962698
min        0.000000
25%        1.000000
50%        1.000000
75%        1.000000
max       111.000000
Name: NumberOfBuildings, dtype: float64
```

PropertyName	PrimaryPropertyType	NumberOfBuildings
University of Washington - Seattle Campus	UNIVERSITY	111.0
Entire Campus	UNIVERSITY	39.0
South Park	LARGE_OFFICE	16.0
NSCC MAIN CAMPUS	UNIVERSITY	11.0
KC Metro Transit Atlantic Ce...	OTHER	10.0
(ID#24086)Campus1		

Premières analyses

Superficie des bâtiments



Superficie des parkings

```
df[ 'PropertyGFAParking'].describe()
```

```
count      1629.000000
mean     13053.118478
std      42657.818840
min       0.000000
25%      0.000000
50%      0.000000
75%      0.000000
max     512608.000000
Name: PropertyGFAParking, dtype: float64
```

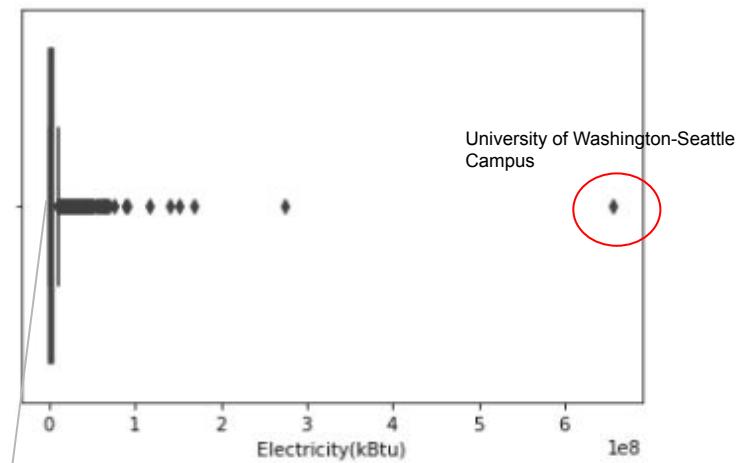
Premières analyses

Consommation annuelle en géothermie

```
: df['SteamUse(kBtu)'].describe()
```

	SteamUse(kBtu)
count	1.629000e+03
mean	5.211473e+05
std	5.575266e+06
min	0.000000e+00
25%	0.000000e+00
50%	0.000000e+00
75%	0.000000e+00
max	1.349435e+08
Name:	SteamUse(kBtu), dtype: float64

Consommation annuelle en électricité



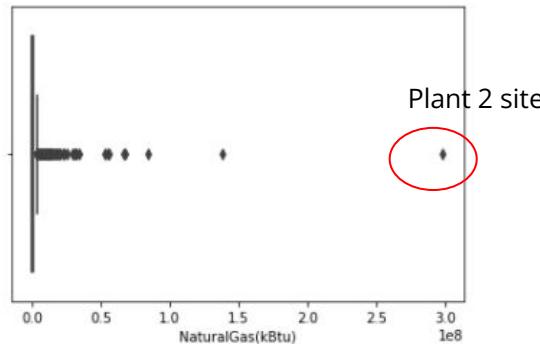
Electricity(kBtu)	PrimaryPropertyType	PropertyName
1570	-115417.0	SMALL_AND_MID_SIZED_OFFICE
		Bullitt Center

Premières analyses

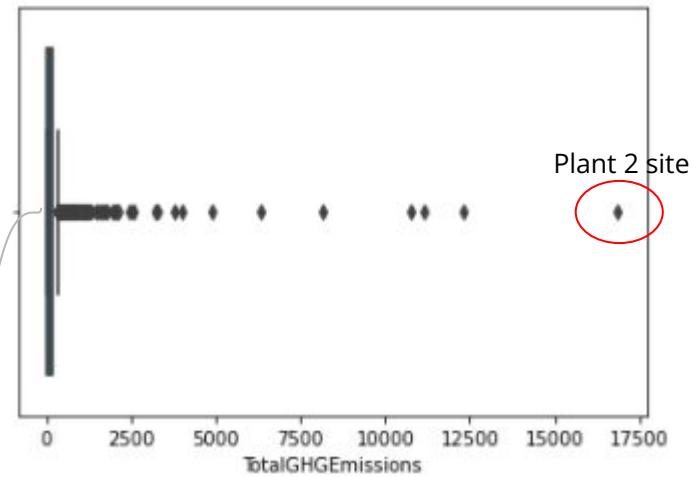
Consommation annuelle en gaz naturel

```
: df['NaturalGas(kBtu)'].describe()
```

```
count      1.629000e+03
mean      1.990318e+06
std       9.473053e+06
min       0.000000e+00
25%      0.000000e+00
50%      4.960960e+05
75%      1.522809e+06
max      2.979090e+08
Name: NaturalGas(kBtu), dtype: float64
```



Emissions de gaz à effet de serre



TotalGHGEmissions	PrimaryPropertyType	PropertyName
1570	-0.8	SMALL_AND_MID_SIZED_OFFICE

Suite des traitements

Traitement des valeurs manquantes



Imputation par 0 ou par la médiane selon les variables

Filtre sur l'energy STAR score présent



1069 observations restantes
Enlève certains outliers comme l'université de washington

Traitement des outliers

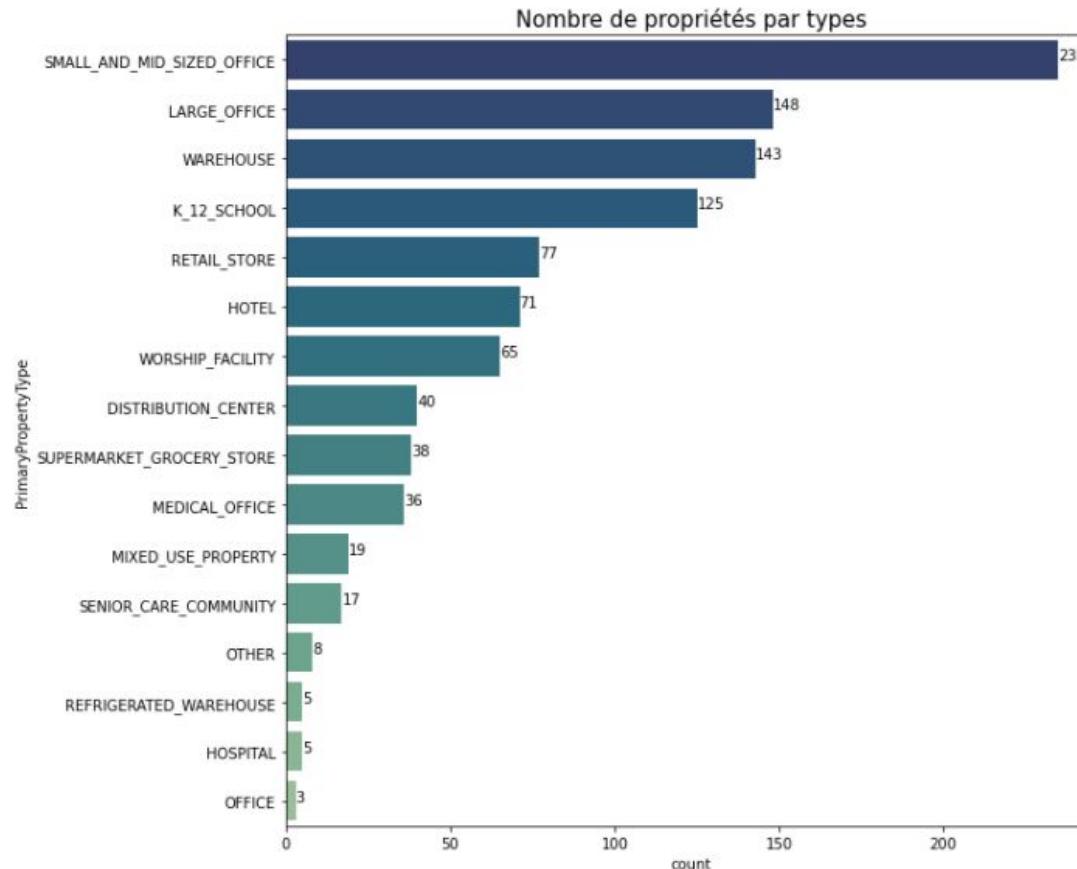


Filtre sur les z-scores <5
1035 observations restantes

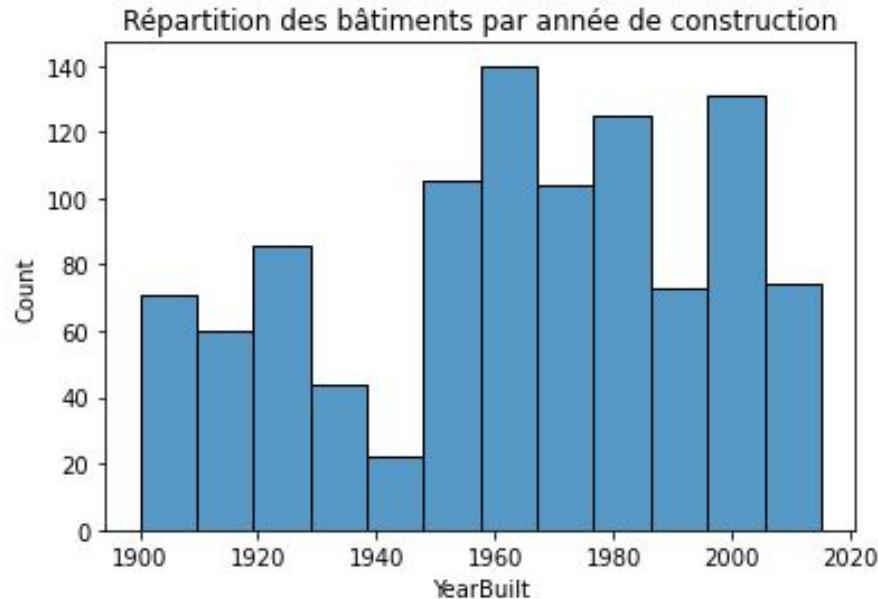


2. Analyses exploratoires

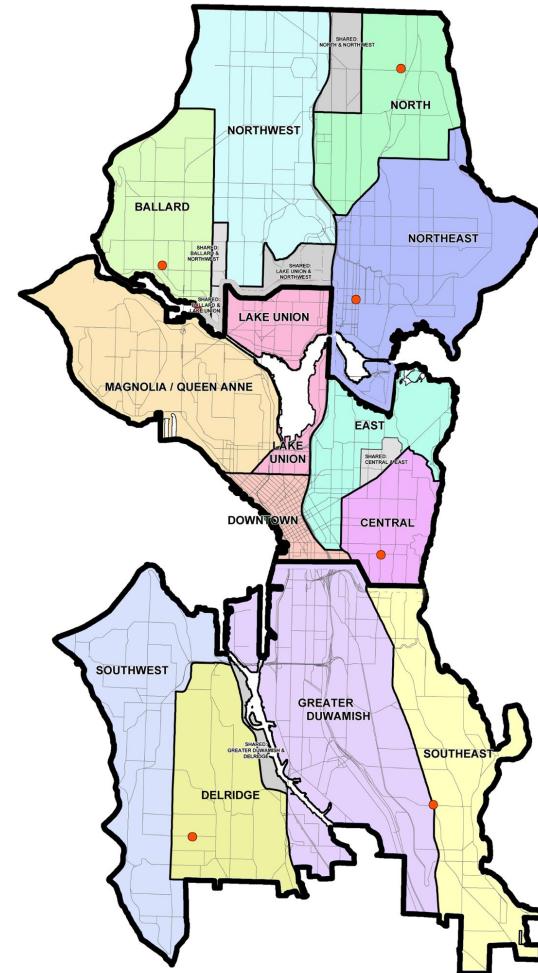
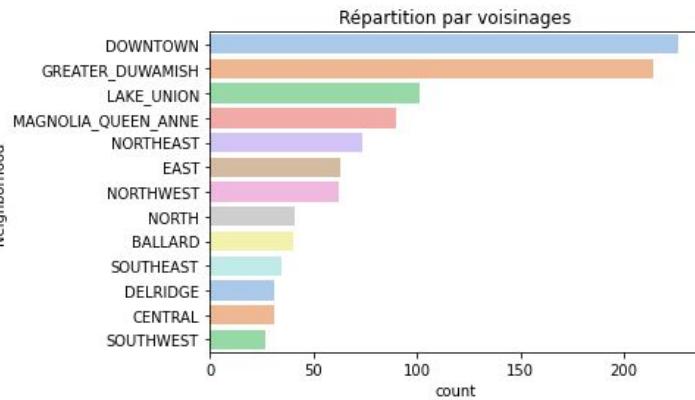
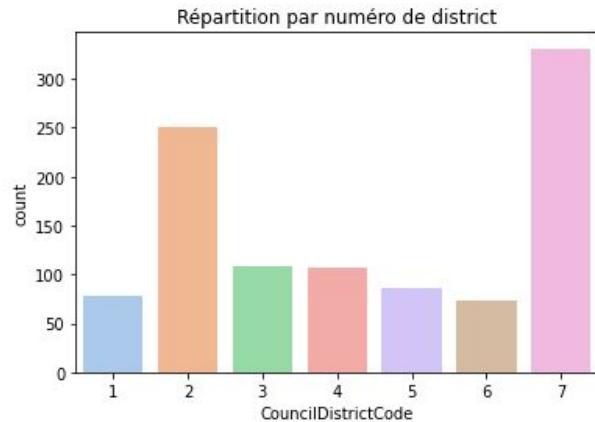
Analyses univariées



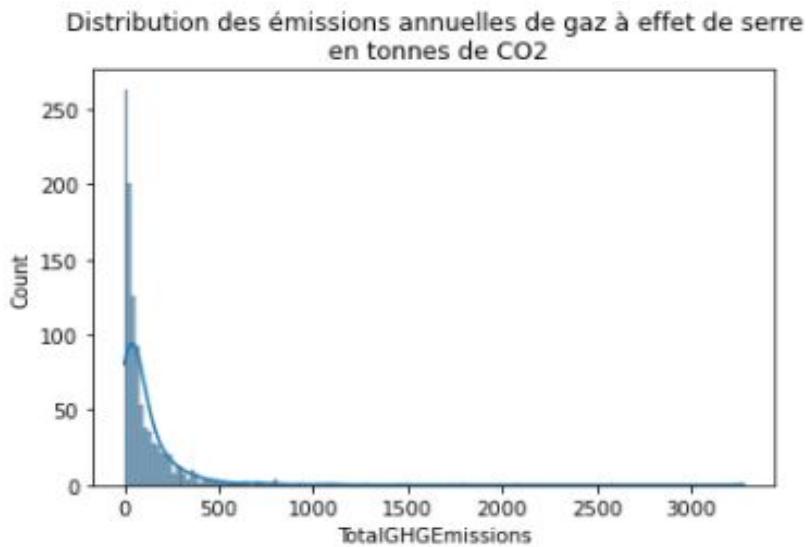
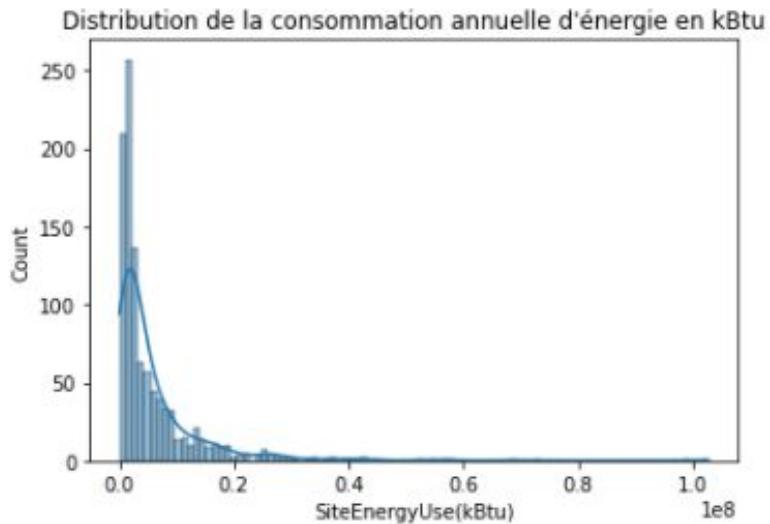
Analyses univariées



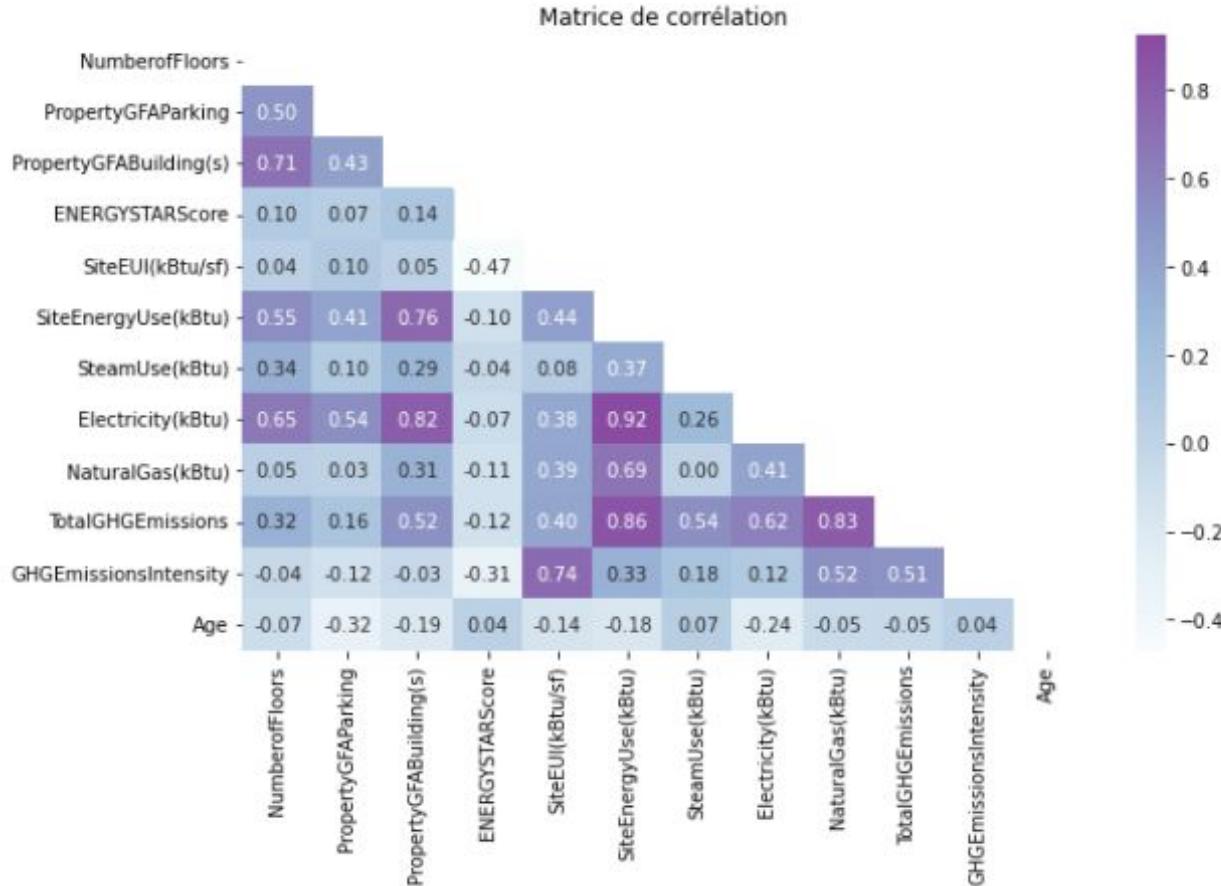
Analyses univariées



Analyses univariées

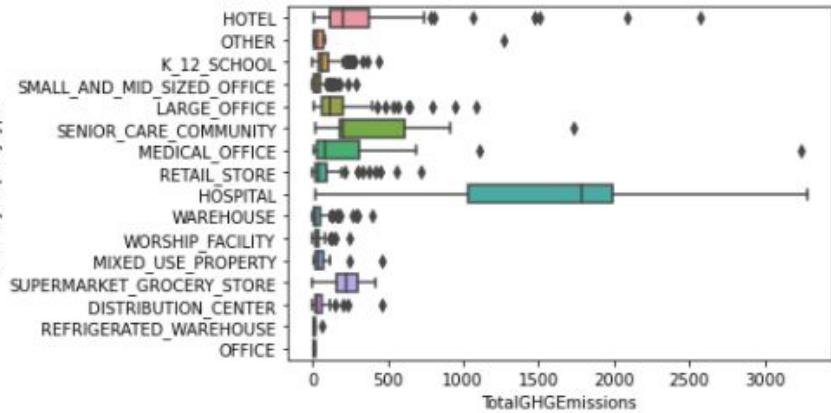


Analyses bi-variées: corrélations

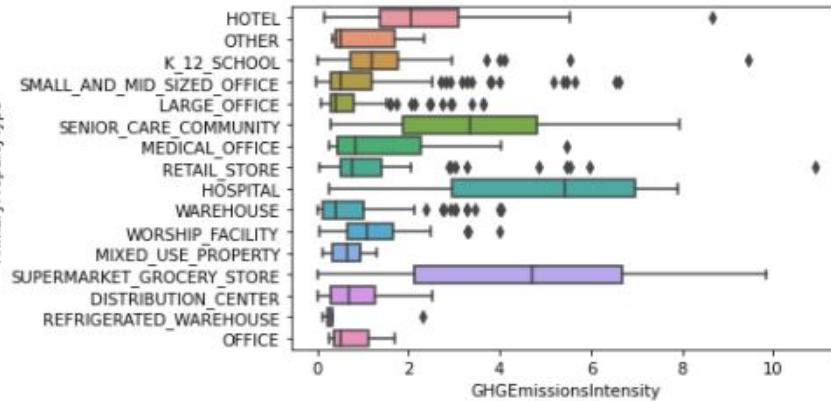


Analyses bi-variées

PrimaryPropertyType



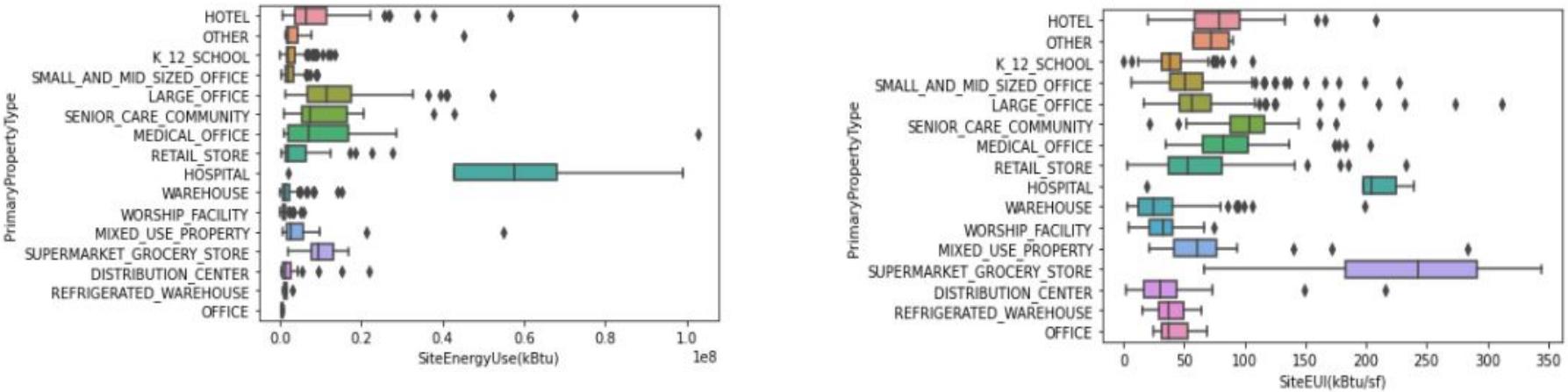
PrimaryPropertyType



Les hôpitaux sont les bâtiments qui consomment le plus d'énergie et émettent le plus de CO₂ au global.

Cependant, en prenant le ratio par square feet, ils sont devancés par les supermarchés.

Analyses bi-variées



On regroupe les types de bâtiments

```
def groupType(x):
    if x in ['K_12 SCHOOL', 'DISTRIBUTION_CENTER', 'WORSHIP_FACILITY', 'OFFICE', 'SMALL_AND_MID_SIZED_OFFICE', 'WAREHOUSE', \
        'MIXED_USE_PROPERTY', 'REFRIGERATED_WAREHOUSE']:
        return 'low_energyuse_co2emi'
    else:
        return x
```



3. Feature engineering

Ratios des différentes sources d'énergies

```
df['propSteamUse']=df['SteamUse(kBtu)']/(df['SteamUse(kBtu)']+df['NaturalGas(kBtu)']+df['Electricity(kBtu)'])
```

```
df['propNaturalGas']=df['NaturalGas(kBtu)']/(df['SteamUse(kBtu)']+df['NaturalGas(kBtu)']+df['Electricity(kBtu)'])
```

```
df['propElectricity']=df['Electricity(kBtu)']/(df['SteamUse(kBtu)']+df['NaturalGas(kBtu)']+df['Electricity(kBtu)'])
```

Ratios des superficies des différents usages du bâtiment

On connaît les superficies associées aux premiers seconds et troisièmes usages s'ils existent.

Nous avons créé une variable par type de propriété contenant la proportion de la superficie correspondante de chaque bâtiment.

exemple : La deuxième observation correspond à un bâtiment dont 80 % de la superficie est un hôtel, 14.5 % un parking et 0.5% un restaurant.

HOTEL	OTHER	low_energyuse_co2emi	LARGE_OFFICE	SENIOR_CARE_COMMUNITY	MEDICAL_OFFICE	RETAIL_STORE	HOSPITAL
1.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.809918	0.0	0.0	0.0	0.0	0.0	0.0	0.0

SUPERMARKET_GROCERY_STORE	PARKING	RESTAURANT	UNIVERSITY	LABORATORY	NA
0.0	0.000000	0.000000	0.0	0.0	0
0.0	0.145453	0.044629	0.0	0.0	0



3. Modélisation

Variables conservées

On conserve les variables suivantes :

- Les proportions des différentes sources d'énergie utilisées
- Les proportions des différents types d'usage : Parking, hôpital, restaurant, université...
- Les superficies des bâtiments et du parking, le nombre d'étages, l'âge
- Le numéro du District
- L'energy star score
- Les variables à prédire : SiteEnergyUse(kBtu) et TotalGHGEmissions

Pre-processing

- On utilise la fonction `pd.get_dummies()` afin de traiter les variables catégorielles et leur associer des valeurs numériques sans relation d'ordre.
- On standardise l'ensemble du dataset.

Pre-processing : Recherche des variables significatives

Pour les deux prédictions, on crée un modèle de régression linéaire multiple par sélection backward, afin de ne repérer que les variables significatives.

exemple : modèle de régression linéaire créé pour la consommation d'énergie

		coef	std err	t	P> t
R-squared:	0.747				
Adj. R-squared:	0.744				
F-statistic:	200.9				
Prob (F-statistic):	1.33e-291				
Log-Likelihood:	-756.69				
AIC:	1545.				
BIC:	1624.				
-----	-----	-----	-----	-----	-----
	Intercept	-1.000e-16	0.016	-6.36e-15	1.000
	SENIOR_CARE_COMMUNITY	0.1241	0.023	5.369	0.000
	propElectricity	-0.0617	0.018	-3.517	0.000
	PARKING	0.0845	0.020	4.316	0.000
	low_energyuse_co2emi	0.1818	0.065	2.781	0.006
	propSteamUse	0.0552	0.017	3.308	0.001
	RETAIL_STORE	0.1386	0.037	3.767	0.000
	PropertyGFAParking	0.1320	0.021	6.349	0.000
	LARGE_OFFICE	0.1003	0.044	2.303	0.022
	HOTEL	0.1856	0.036	5.153	0.000
	HOSPITAL	0.3331	0.018	18.480	0.000
	OTHER	0.0716	0.020	3.593	0.000
	MEDICAL_OFFICE	0.1626	0.027	6.003	0.000
	PropertyGFABuildings	0.6567	0.020	32.798	0.000
	LABORATORY	0.0485	0.016	3.037	0.002
	SUPERMARKET_GROCERY_STORE	0.2378	0.029	8.173	0.000
-----	-----	-----	-----	-----	-----

Recherche des variables significatives

On garde les variables significatives pour l'une des deux prédictions

```
df_std=df_std[['propSteamUse','propNaturalGas','propElectricity','LARGE_OFFICE','MEDICAL_OFFICE','HOSPITAL','RETAIL_STORE',\n    'PARKING','SUPERMARKET_GROCERY_STORE','low_energyuse_co2emi',\n    'OTHER','LABORATORY','HOTEL','SENIOR_CARE_COMMUNITY','PropertyGFAParking','PropertyGFABuildings',\n    'SiteEnergyUse_kBtu','ENERGYSTARScore','TotalGHGEmissions']]
```

Observations atypiques et influentes

Par rapport au modèle de régression linéaire multiple que l'on a créé, on enlève les observations atypiques et influentes.

Source :

<https://openclassrooms.com/fr/courses/4525326-realisez-des-modelisations-de-donnees-performantes/5754143-analysez-les-resultats>

L'observation i est atypique sur une variable prédictive si le i -ème élément diagonal de la matrice de projection sur X dépasse un certain seuil.

L'observation i est atypique sur la variable à prédire le résidu de Student interne dépasse un certain seuil.

L'observation i est influente sur le modèle linéaire si la distance de cook de l'observation i dépasse un certain seuil

Observations atypiques et influentes

```
#ensemble des valeurs atypiques et influentes au modèle de regression linéaire multiple de la consommation d'énergie.
```

```
dh=df_init.loc[(df_init['dcooks'] > seuil_dcook)&((df_init['rstudent'] > seuil_rstudent)|(df_init['levier'] > seuil_levier)), :]
```

```
#Nombre de valeurs atypiques et influentes
```

```
dh.shape[0]
```

57

```
#répartition par type de propriétés avant suppression  
df_init.PrimaryPropertyType.value_counts()
```

SMALL_AND_MID_SIZED_OFFICE	235
LARGE_OFFICE	148
WAREHOUSE	143
K_12 SCHOOL	125
RETAIL_STORE	77
HOTEL	71
WORSHIP_FACILITY	65
DISTRIBUTION_CENTER	40
SUPERMARKET_GROCERY_STORE	38
MEDICAL_OFFICE	36
MIXED_USE_PROPERTY	19
SENIOR_CARE_COMMUNITY	17
OTHER	8
HOSPITAL	5
REFRIGERATED_WAREHOUSE	5
OFFICE	3

Name: PrimaryPropertyType, dtype: int64

```
#répartition par type de propriétés après suppression  
df_after.PrimaryPropertyType.value_counts()
```

SMALL_AND_MID_SIZED_OFFICE	234
WAREHOUSE	143
LARGE_OFFICE	128
K_12 SCHOOL	125
RETAIL_STORE	76
WORSHIP_FACILITY	64
HOTEL	63
DISTRIBUTION_CENTER	40
SUPERMARKET_GROCERY_STORE	36
MEDICAL_OFFICE	28
MIXED_USE_PROPERTY	17
SENIOR_CARE_COMMUNITY	9
OTHER	6
REFRIGERATED_WAREHOUSE	5
OFFICE	3
HOSPITAL	1

Name: PrimaryPropertyType, dtype: int64

Train_test_split

On sépare notre jeu de données en un jeu d'entraînement (80%) et un jeu de test (20%).

On veille à garder la même distribution de la variable à prédire (utilisation des classes formés par les déciles).

Prédictions de la consommation d'énergie

sans energy star score

	reg moyenne	Ridge	Lasso	SVR	SVR noyau gaussien	RIDGE noyau gaussien	Random Forest	Gradient boosting
Type du modèle	NaN	linéaire	linéaire	linéaire	non linéaire	non linéaire	ensembliste	ensembliste
Meilleurs hyperparamètres	NaN	{'alpha': 3.594}	{'alpha': 0.002}	{'C': 100, 'epsilon': 0.1}	{'C': 10, 'epsilon': 0.1, 'gamma': 0.01}	{'alpha': 0.1, 'gamma': 0.01}	{'max_depth': 10, 'max_features': 'auto'}	{'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 100}
RMSE	0.673	0.263	0.264	0.263	0.244	0.249	0.313	0.286
R ²	0.000	0.815	0.813	0.806	0.844	0.828	0.735	0.781
Temps d'exécution entraînement (sec)	NaN	0.006	0.003	3.136	0.048	0.039	0.491	0.139
Temps d'exécution prédiction (sec)	NaN	0.003	0.002	0.004	0.008	0.006	0.02	0.002
moyenne pour 20 splits	NaN	0.768	0.765	0.743	0.648	0.639	0.725	0.721
ecart-type(R ²) pour 20 split	NaN	0.057	0.059	0.065	0.125	0.135	0.059	0.079

Prédictions de la consommation d'énergie

avec energy star score

	reg moyenne	Ridge	Lasso	SVR	SVR noyau gaussien	RIDGE noyau polynomial	Random Forest	Gradient boosting
Type du modèle	NaN	linéaire	linéaire	linéaire	non linéaire	non linéaire	ensembliste	ensembliste
Meilleurs hyperparamètres	NaN	{'alpha': 3.594}	{'alpha': 0.0}	{'C': 100, 'epsilon': 0.1}	{'C': 10, 'epsilon': 0.1, 'gamma': 0.01}	{'alpha': 1.0}	{'max_depth': 15, 'max_features': 'auto'}	{'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 100}
RMSE	0.673	0.24	0.239	0.24	0.213	0.191	0.272	0.252
R ²	0.000	0.848	0.851	0.844	0.875	0.905	0.793	0.826
Temps d'exécution entraînement (sec)	NaN	0.013	0.012	3.299	0.053	0.058	0.633	0.313
Temps d'exécution prédiction (sec)	NaN	0.013	0.002	0.004	0.01	0.013	0.019	0.003
moyenne pour 20 splits	NaN	0.816	0.818	0.8	0.748	0.874	0.818	0.836
écart-type(R ²) pour 20 split	NaN	0.043	0.042	0.048	0.089	0.025	0.038	0.039

Prédictions des émissions de co2

sans energy star score

	reg aleatoire	Ridge	Lasso	SVR	SVR noyau gaussien	RIDGE noyau polynomial	Random Forest	Gradient boosting
Type du modèle	NaN	linéaire	linéaire	linéaire	non linéaire	non linéaire	ensembliste	ensembliste
Meilleurs hyperparamètres	NaN	{'alpha': 3.594}	{'alpha': 0.002}	{'C': 100, 'epsilon': 0.1}	{'C': 10, 'epsilon': 0.1, 'gamma': 0.01}	{'alpha': 1.0}	{'max_depth': 20, 'max_features': 'auto'}	{'learning_rate': 0.1, 'max_depth': 2, 'n_estimators': 100}
RMSE	1.450	0.294	0.293	0.294	0.19	0.189	0.22	0.208
R ²	-2.196	0.447	0.448	0.211	0.739	0.779	0.682	0.738
Temps d'exécution entraînement (sec)	NaN	0.004	0.003	4.133	0.053	0.039	0.652	0.201
Temps d'exécution prédiction (sec)	NaN	0.003	0.002	0.003	0.011	0.008	0.023	0.002
moyenne pour 20 splits	NaN	0.436	0.426	0.191	0.643	0.79	0.658	0.699
ecart-type(R ²) pour 20 split	NaN	0.116	0.12	0.161	0.149	0.054	0.087	0.081

Prédictions des émissions de co2

avec energy star score

	reg aleatoire	Ridge	Lasso	SVR	SVR noyau gaussien	RIDGE noyau polynomial	Random Forest	Gradient boosting
Type du modèle	NaN	linéaire	linéaire	linéaire	non linéaire	non linéaire	ensembliste	ensembliste
Meilleurs hyperparamètres	NaN	{'alpha': 3.594}	{'alpha': 0.002}	{'C': 1, 'epsilon': 0.1}	{'C': 100, 'epsilon': 0.1, 'gamma': 0.01}	{'alpha': 1.0}	{'max_depth': 30, 'max_features': 'auto'}	{'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 100}
RMSE	1.472	0.281	0.28	0.281	0.155	0.155	0.184	0.208
R ²	-2.396	0.517	0.518	0.282	0.876	0.861	0.783	0.774
Temps d'exécution entraînement (sec)	NaN	0.003	0.002	0.08	0.09	0.055	0.703	0.146
Temps d'exécution prédiction (sec)	NaN	0.001	0.001	0.003	0.007	0.012	0.025	0.002
moyenne pour 20 splits	NaN	0.494	0.486	0.245	0.722	0.849	0.724	0.753
écart-type(R ²) pour 20 split	NaN	0.097	0.102	0.143	0.141	0.037	0.084	0.074

Amélioration

On constate que les prédictions des émissions de co2 sans energy star score sont un peu moins performantes que celles de la consommation d'énergie.

On décide alors d'utiliser la prédiction de consommation d'énergie pour prédire les émissions de co2.

Amélioration

	reg aleatoire	Ridge	Lasso	SVR	SVR noyau gaussien	RIDGE noyau polynomial	Random Forest	Gradient boosting
Type du modèle	NaN	linéaire	linéaire	linéaire	non linéaire	non linéaire	ensembliste	ensembliste
Meilleurs hyperparamètres	NaN	{'alpha': 3.594}	{'alpha': 0.002}	{'C': 10, 'epsilon': 0.1}	{'C': 100, 'epsilon': 0.1, 'gamma': 0.01}	{'alpha': 1.0}	{'max_depth': 10, 'max_features': 'auto'}	{'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 100}
RMSE	1.429	0.261	0.26	0.261	0.138	0.141	0.153	0.161
R ²	-1.906	0.583	0.592	0.579	0.899	0.885	0.868	0.855
meilleur score gridsearchCV	NaN	0.749884	0.752365	0.747401	0.878011	0.885217	0.887695	0.894456
Temps d'exécution entraînement (sec)	NaN	0.005	0.004	1.174	0.131	0.072	1.176	0.307
Temps d'exécution prédiction (sec)	NaN	0.003	0.003	0.006	0.011	0.012	0.048	0.005
moyenne pour 20 splits	NaN	0.649	0.655	0.652	0.754	0.882	0.867	0.872
écart-type(R ²) pour 20 split	NaN	0.071	0.069	0.066	0.142	0.036	0.037	0.036

Conclusion

Pour conclure,

- Nos analyses ont montré des valeurs assez dispersées dans le dataset et nous avons effectué des traitements pour recentrer ces valeurs: en filtrant sur les z-scores et en utilisant des techniques sur les régressions linéaires.
- Pour prédire la consommation d'énergie, nous avons montré que les modèles linéaires étaient suffisamment performants.
- Finalement, nous avons montré que la présence de l'energy star score n'était pas cruciale pour avoir de bonnes prédictions des émissions de co2. Nous avons pu améliorer ces prédictions en utilisant la prédiction précédente de la consommation d'énergie.