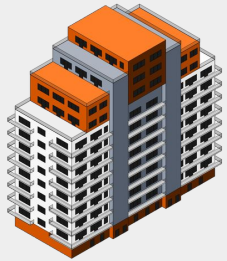
Projet 3

Parcours Machine Learning

Anticiper les besoins en consommation électrique des bâtiments

Seattle



Sommaire

1. Présentation
2. Nettoyage du jeu de données
3. Création de variables
4. Exploration des données
    - 4.1.1. Analyse univariée
    - 4.1.2. Analyse bivariée
5. Préparation des données et des modèles de base
6. Prédiction de la consommation d'électricité
    - 6.1. Avec ENERGYSTARScore
    - 6.2. Sans ENERGYSTARScore
7. Prédiction des émissions de CO2
    - 7.1. Avec ENERGYSTARScore
    - 7.2. Sans ENERGYSTARScore
8. Conclusion

I. Présentation

Mission

Notre objectif est de prédire les émissions de CO2 et la consommation d'électricité des bâtiments de la ville de Seattle à partir de relevés effectués par nos agents en 2015 et en 2016. Pour ce faire on se basera sur les informations contenues dans les données de consommation comme la taille et les usages des bâtiments, la date de construction etc.

Feuille de route :

- Nettoyage des données
- Création de features
- Sélection des features
- Evaluation de différents modèles
- Choix du modèle le plus adapté à notre problème métier

II. Nettoyage des données

Étapes du nettoyage

- Après avoir chargé les données, on concatène les deux jeux de données (celui de 2015 et 2016) pour en obtenir un seul.
- On filtre les variables jugées sans grand intérêt et on garde celles pertinentes (celles ayant trait à l'usage/tailles des bâtiments, voisinage, présence de parking ou pas… sans oublier les variables cibles, elles sont au nombre de 19).
- On calcule la moyenne de la consommation d'énergie ainsi que les émissions CO2 des bâtiments qui ont été relevés en 2015 et 2016.
- On filtre la valeur max de la variable qui correspond à la consommation d'énergie car elle représente un outlier.
- On impute les variables correspondant aux usages secondaires et tertiaires par 0 pour les variables numériques et par 'None' pour les variables catégorielles.
- On utilise KNNImputer pour imputer les variables numériques et on remplace les valeurs manquantes des variables catégorielles par la valeur la plus fréquente.
- On fusionne les sous-catégories redondantes des catégories correspondant au type de bâtiments, voisinage ainsi que l'usage primaire des bâtiments et on regroupe les sous-catégories qui sont en petit nombre sous "Diverse".
- Une fois l'opération de nettoyage effectuée, notre jeu de données est de dimension 3427x19.

III. Création de variables

Variables créées

‘GFA_per_Floor’ : qui est la surface par niveau.

```
dataset['GFA_per_floor']=(dataset['PropertyGFABuilding_s_'])/(dataset['NumberofFloors']+1)
```

‘old’ : correspond à l’âge du bâtiment.

```
dataset['Old']=dataset['DataYear']- dataset['YearBuilt']
```

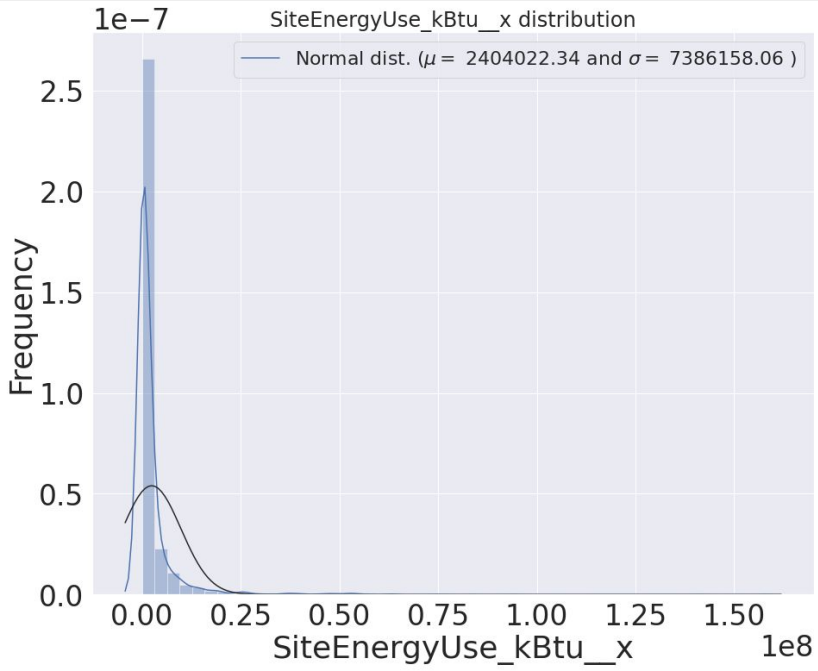
‘Parking_ratio’ : ratio du parking/surface total du bâtiment.

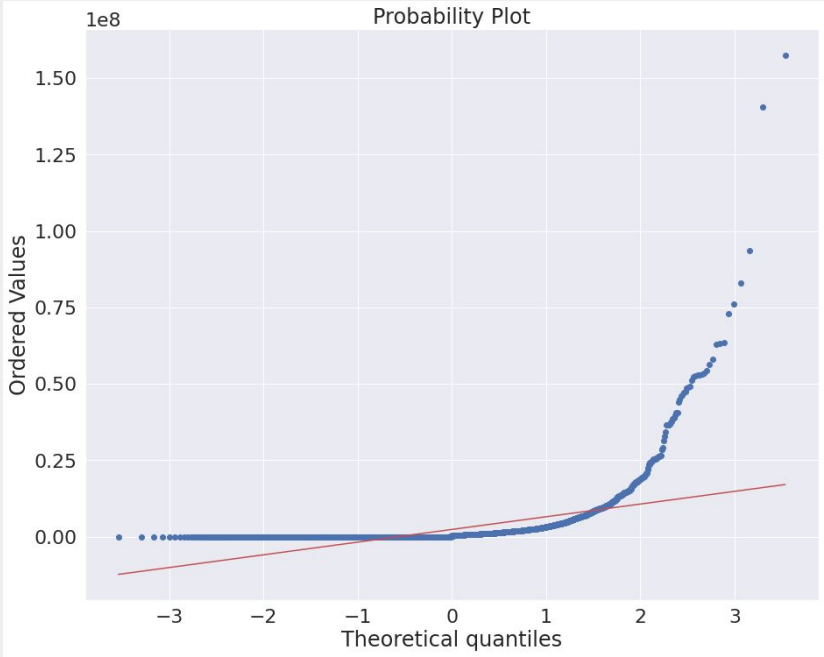
```
dataset['Parking_ratio'] = dataset['PropertyGFAParking']/dataset['PropertyGFATotal']
```

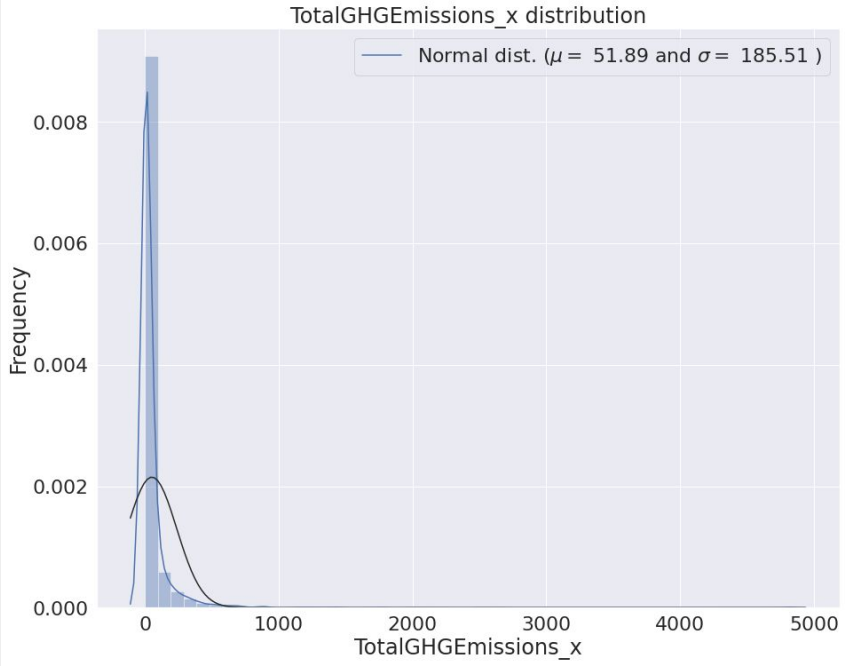
IV. Analyse exploratoire

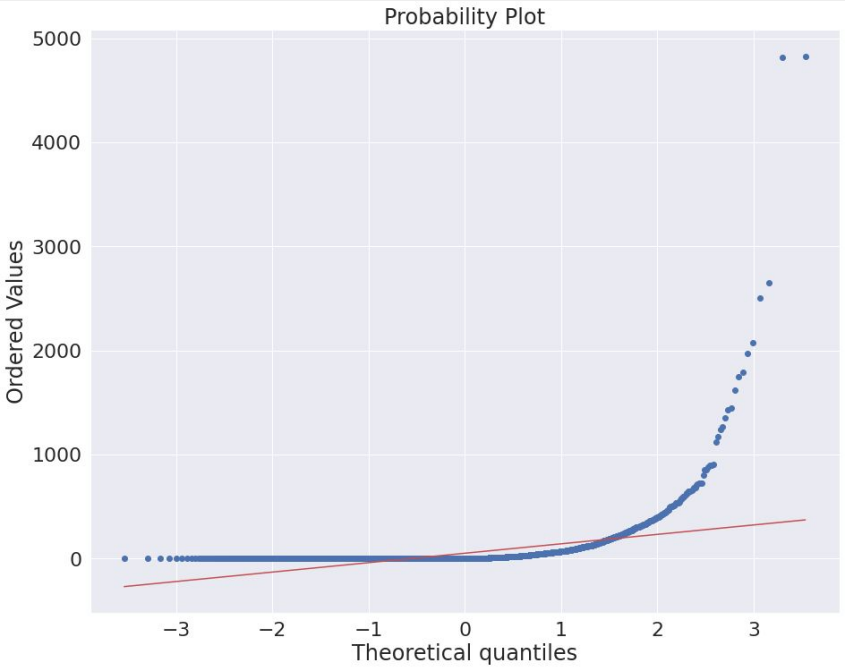
Analyse univariée

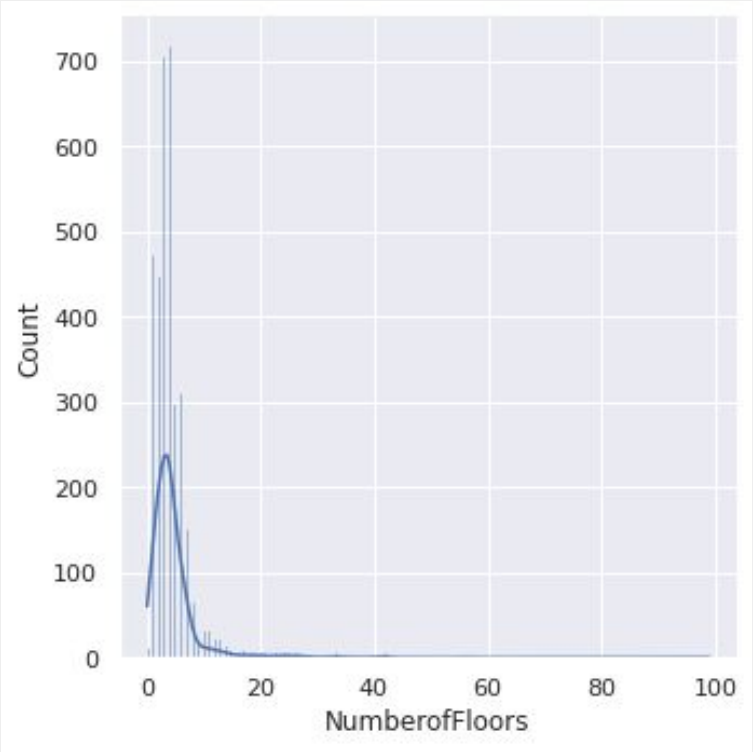
- L’analyse univariée nous permettra de décrire le comportement d’une variable, nous renseignera sur la nature de sa distribution.

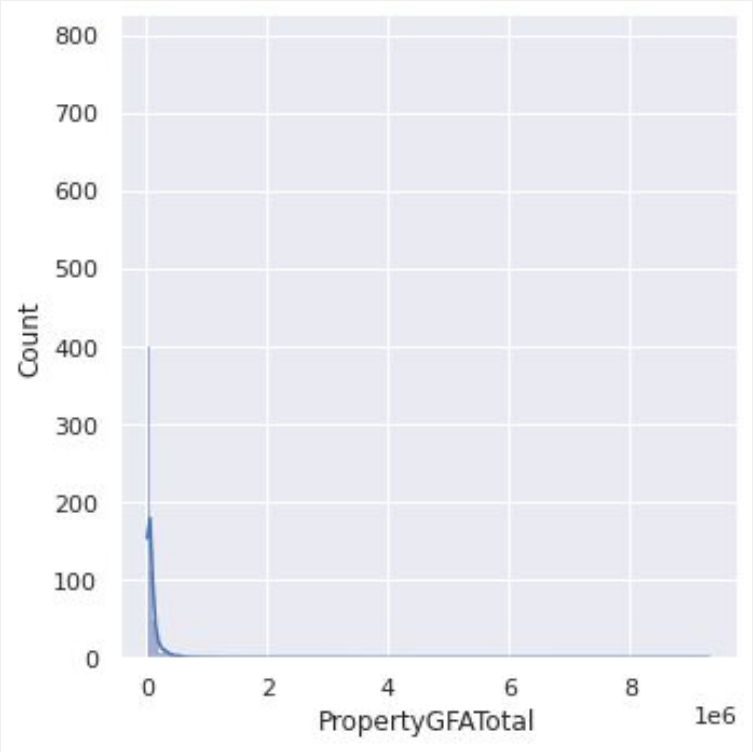

1e−7
SiteEnergyUse_kBtu__x distribution
Normal dist. (μ = 2404022.34 and σ = 7386158.06)
Frequency
2.5
2.0
1.5
1.0
0.5
0.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
SiteEnergyUse_kBtu__x
1e8


1e8
Probability Plot
Ordered Values
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−3
−2
−1
0
1
2
3
Theoretical quantiles


TotalGHGEmissions_x distribution
Normal dist. (μ = 51.89 and σ = 185.51)
Frequency
0.000
0.002
0.004
0.006
0.008
0
1000
2000
3000
4000
5000
TotalGHGEmissions_x

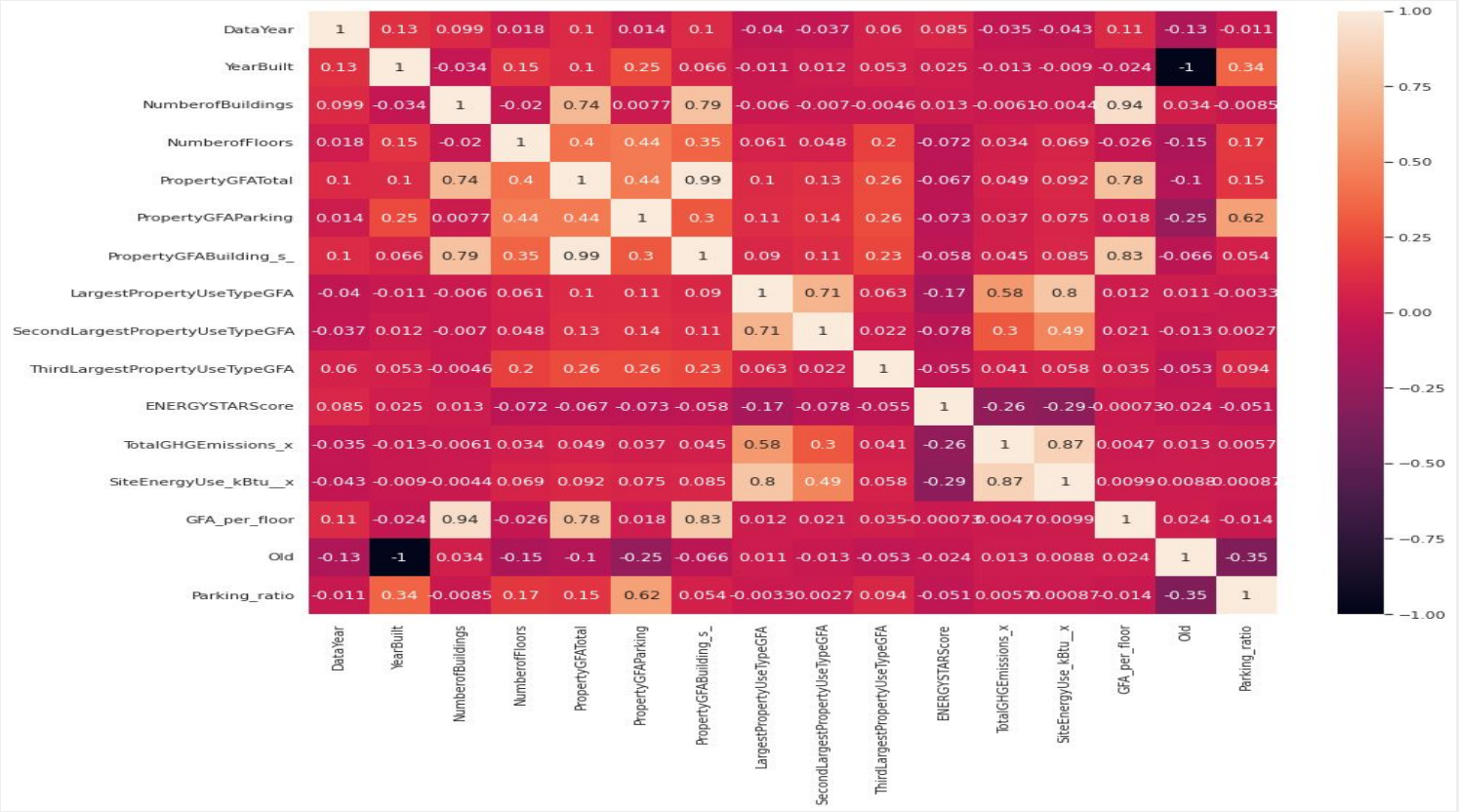

Probability Plot
Ordered Values
0
1000
2000
3000
4000
5000
−3
−2
−1
0
1
2
3
Theoretical quantiles

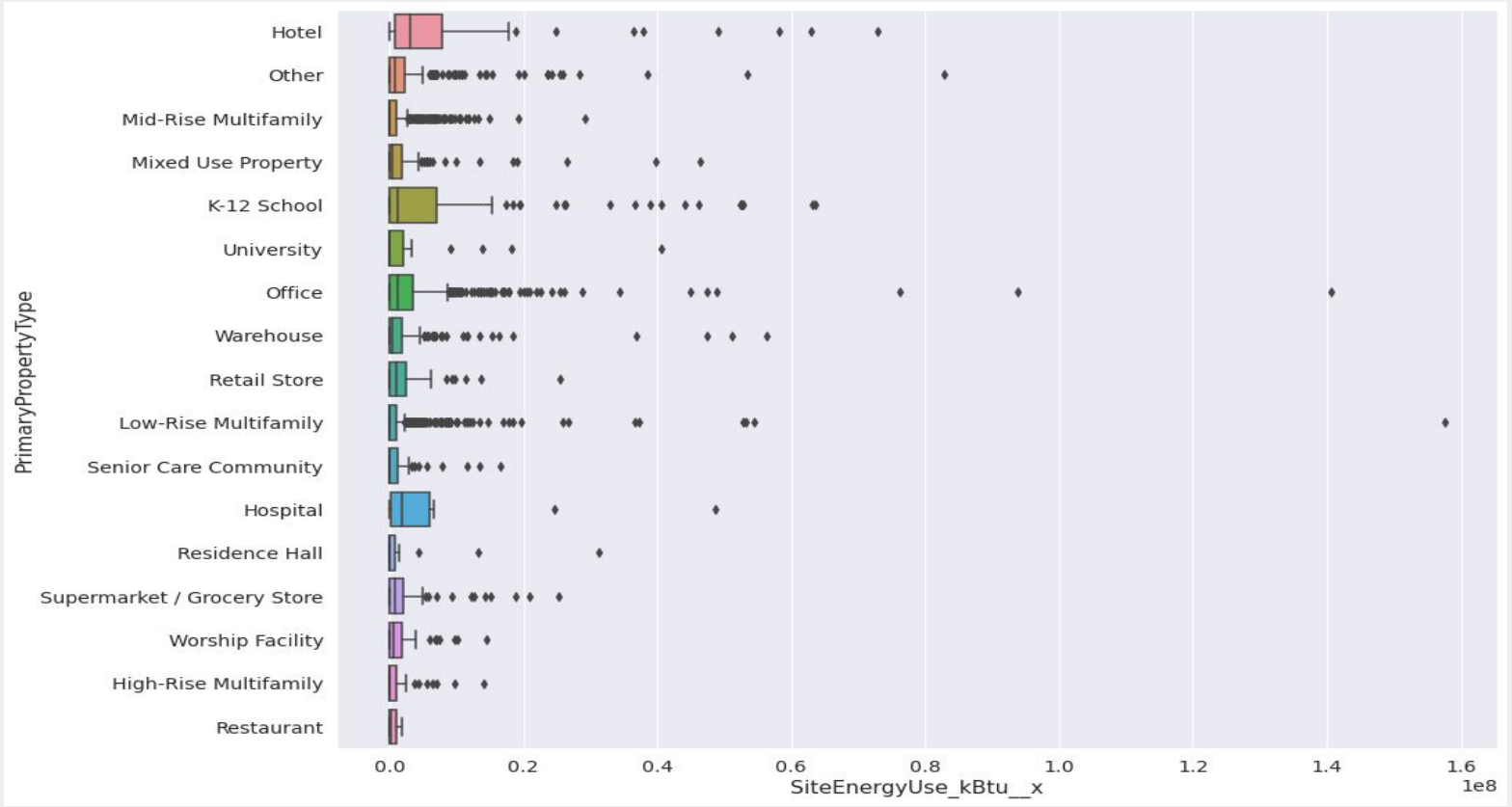

Count
NumberofFloors
0
100
200
300
400
500
600
700
0
20
40
60
80
100


Count
PropertyGFATotal
1e6
0
100
200
300
400
500
600
700
800
0
2
4
6
8

Analyse bivariée

- Grâce à l’analyse bivariée on pourra étudier les relations entre deux variables distinctes, quantitatives ou qualitatives.


DataYear 1 0.13 0.099 0.018 0.1 0.014 0.1 -0.04 -0.037 0.06 0.085 -0.035 -0.043 0.11 -0.13 -0.011
YearBuilt 0.13 1 -0.034 0.15 0.1 0.25 0.066 -0.011 0.012 0.053 0.025 -0.013 -0.009 -0.024 -1 0.34
NumberofBuildings 0.099 -0.034 1 -0.02 0.74 0.0077 0.79 -0.006 -0.007 -0.0046 0.013 -0.0061 -0.0044 0.94 0.034 -0.0085
NumberofFloors 0.018 0.15 -0.02 1 0.4 0.44 0.35 0.061 0.048 0.2 -0.072 0.034 0.069 -0.026 -0.15 0.17
PropertyGFATotal 0.1 0.1 0.74 0.4 1 0.44 0.99 0.1 0.13 0.26 -0.067 0.049 0.092 0.78 -0.1 0.15
PropertyGFAParking 0.014 0.25 0.0077 0.44 0.44 1 0.3 0.11 0.14 0.26 -0.073 0.037 0.075 0.018 -0.25 0.62
PropertyGFABuilding_s_ 0.1 0.066 0.79 0.35 0.99 0.3 1 0.09 0.11 0.23 -0.058 0.045 0.085 0.83 -0.066 0.054
LargestPropertyUseTypeGFA -0.04 -0.011 -0.006 0.061 0.1 0.11 0.09 1 0.71 0.063 -0.17 0.58 0.8 0.012 0.011 -0.0033
SecondLargestPropertyUseTypeGFA -0.037 0.012 -0.007 0.048 0.13 0.14 0.11 0.71 1 0.022 -0.078 0.3 0.49 0.021 -0.013 0.0027
ThirdLargestPropertyUseTypeGFA 0.06 0.053 -0.0046 0.2 0.26 0.26 0.23 0.063 0.022 1 -0.055 0.041 0.058 0.035 -0.053 0.094
ENERGYSTARScore 0.085 0.025 0.013 -0.072 -0.067 -0.073 -0.058 -0.17 -0.078 -0.055 1 -0.26 -0.29 -0.00073 0.024 -0.051
TotalGHGEmissions_x -0.035 -0.013 -0.0061 0.034 0.049 0.037 0.045 0.58 0.3 0.041 -0.26 1 0.87 0.0047 0.013 0.0057
SiteEnergyUse_kBtu__x -0.043 -0.009 -0.0044 0.069 0.092 0.075 0.085 0.8 0.49 0.058 -0.29 0.87 1 0.0099 0.0088 0.00087
GFA_per_floor 0.11 -0.024 0.94 -0.026 0.78 0.018 0.83 0.012 0.021 0.035 -0.00073 0.0047 0.0099 1 0.024 -0.014
Old -0.13 -1 0.034 -0.15 -0.1 -0.25 -0.066 0.011 -0.013 -0.053 -0.024 0.013 0.0088 0.024 1 -0.35
Parking_ratio -0.011 0.34 -0.0085 0.17 0.15 0.62 0.054 -0.0033 0.0027 0.094 -0.051 0.0057 0.00087 -0.014 -0.35 1
DataYear
YearBuilt
NumberofBuildings
NumberofFloors
PropertyGFATotal
PropertyGFAParking
PropertyGFABuilding_s_
LargestPropertyUseTypeGFA
SecondLargestPropertyUseTypeGFA
ThirdLargestPropertyUseTypeGFA
ENERGYSTARScore
TotalGHGEmissions_x
SiteEnergyUse_kBtu__x
GFA_per_floor
Old
Parking_ratio
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Hotel
Other
Mid-Rise Multifamily
Mixed Use Property
K-12 School
University
Office
Warehouse
Retail Store
Low-Rise Multifamily
Senior Care Community
Hospital
Residence Hall
Supermarket / Grocery Store
Worship Facility
High-Rise Multifamily
Restaurant
PrimaryPropertyType
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e8
SiteEnergyUse_kBtu__x

V. Préparation des données

Processus de préparation des données

- On choisit les variables qui nous serviront à notre modèle (variables concernant la taille et usage des bâtiments qu'on trouve dans les relevés effectués + les variables créées par nos soins).
- Un nouveau dataframe sera créé à partir des variables choisies. Celles numériques seront normalisées grâce à Robust Scaler, celles catégorielles seront encodées avec get_dummies afin que le modèle puisse les traiter.
- On procède au split du dataframe en 5 folds pour les besoins de la validation croisée.
- On crée une liste avec des modèles de bases (ie sans paramétrage).
- On définit une fonction qui retourne les scores en différentes métriques des modèles de base dont on se servira pour voir le modèle le plus adapté à notre problème (on considérera particulièrement la métrique R^2).

VII. Prédiction de la consommation d’énergie

Avec ENERGYSTARScore

Scores des modèles de base

	r2_score	neg_median_absolute_error	neg_mean_absolute_error	neg_mean_absolute_percentage_error	neg_root_mean_squared_erro
LinearRegression()	-5.25	-1.700354	-1.700354	-0.142001	-5.83715
Ridge()	0.39	-1.272924	-1.272924	-0.104016	-1.86951
Lasso()	0.42	-1.836115	-1.836115	-0.155372	-1.93626
RandomForestRegressor()	0.97	-0.226109	-0.227733	-0.015620	-0.45093
DecisionTreeRegressor()	0.94	-0.320277	-0.321959	-0.022041	-0.63440
GradientBoostingRegressor()	0.97	-0.231929	-0.231684	-0.016198	-0.44750
AdaBoostRegressor()	0.90	-0.695330	-0.727552	-0.065663	-0.78633

- Modèle le plus adapté : Random Forrest
- Meilleurs hyperparamètres : (n_estimators = 1516,min_samples_split= 8,min_samples_leaf= 2, max_features= 0.6, max_depth=78, bootstrap= True)
- Score en CV sur train set : 0.97
- Score en CV sur le test set : 0.97

Sans ENERGYSTARScore

Scores des modèles de base

	r2_score	neg_median_absolute_error	neg_mean_absolute_error	neg_mean_absolute_percentage_error	neg_root_mean_squared_error
LinearRegression()	-5.25	-1.700354	-1.700354	-0.142001	-5.837151
Ridge()	0.39	-1.272924	-1.272924	-0.104016	-1.869514
Lasso()	0.42	-1.836115	-1.836115	-0.155372	-1.936262
RandomForestRegressor()	0.97	-0.226094	-0.226716	-0.015668	-0.450699
DecisionTreeRegressor()	0.94	-0.320806	-0.322565	-0.022224	-0.625421
GradientBoostingRegressor()	0.97	-0.231920	-0.231571	-0.016201	-0.447906
AdaBoostRegressor()	0.90	-0.709910	-0.730256	-0.066834	-0.815233

- Modèle le plus adapté : Random Forrest
- Meilleurs hyperparamètres : (n_estimators = 1516,min_samples_split= 8, min_samples_leaf= 3,max_features= 0.8, max_depth=47, bootstrap= True)
- Score en CV sur train set : 0.97
- Score en CV sur le test set : 0.97

VIII. Prédiction des émissions CO2

Avec ENERGYSTARScore

Scores des modèles de base

	r2_score	neg_median_absolute_error	neg_mean_absolute_error	neg_mean_absolute_percentage_error	neg_root_mean_squared_error
LinearRegression()	-1996039.77	-71.402328	-71.402328	-39.901035	-969.941696
Ridge()	-0.00	-0.917433	-0.917433	-0.399683	-1.353280
Lasso()	0.25	-1.014302	-1.014302	-0.442460	-1.219367
RandomForestRegressor()	0.69	-0.461231	-0.461366	-0.176643	-0.787198
DecisionTreeRegressor()	0.42	-0.603048	-0.604067	-0.222413	-1.083449
GradientBoostingRegressor()	0.69	-0.479302	-0.479719	-0.187177	-0.782839
AdaBoostRegressor()	0.60	-0.812136	-0.711316	-0.410362	-0.935642

- Modèle le plus adapté : Random Forrest
- Meilleurs hyperparamètres : (n_estimators = 1516,min_samples_split= 8, min_samples_leaf= 3,max_features= 0.8, max_depth=47, bootstrap= True)
-
- Score en CV sur train set : 0.70
- Score en CV sur le test set : 0.74

Sans ENERGYSTARScore

Scores des modèles de base

	r2_score	neg_median_absolute_error	neg_mean_absolute_error	neg_mean_absolute_percentage_error	neg_root_mean_squared_erro
LinearRegression()	-1996039.77	-71.402328	-71.402328	-39.901035	-969.94169
Ridge()	-0.00	-0.917433	-0.917433	-0.399683	-1.35328
Lasso()	0.25	-1.014302	-1.014302	-0.442460	-1.21936
RandomForestRegressor()	0.69	-0.458869	-0.460023	-0.176171	-0.78809
DecisionTreeRegressor()	0.42	-0.603281	-0.599264	-0.222428	-1.08028
radientBoostingRegressor()	0.69	-0.479332	-0.479881	-0.187624	-0.78301
AdaBoostRegressor()	0.48	-0.899301	-0.708696	-0.373692	-0.88011

- Modèle le plus adapté : Random Forrest
- Meilleurs hyperparamètres : (n_estimators = 1516,min_samples_split= 8, min_samples_leaf=3,max_features= 0.8, max_depth=47, bootstrap= True)
-
- Score en CV sur train set : 0.69
- Score en CV sur le test set : 0.74

IX. Conclusion

Conclusion

- Le modèle le plus adapté à notre problématique est le Random Forrest.
- La variable Energystarscore n'est pas déterminante dans la prédiction
- Prépondérance de la variable correspondant à la surface consacrée à l'usage primaire du bâtiment (LargestPropertyUseTypeGFA) dans les prédictions.
- Le modèle très précis quand il s'agit de prédire la consommation d'énergie, arrivant à des scores très satisfaisants (R^2 à 0.97). Il l'est moins concernant les émissions CO2, cela dit, il reste assez fiable pour que son utilisation soit envisageable.