

Anticipez les besoins en consommation de bâtiments

Etudiant: Etienne NGENZI

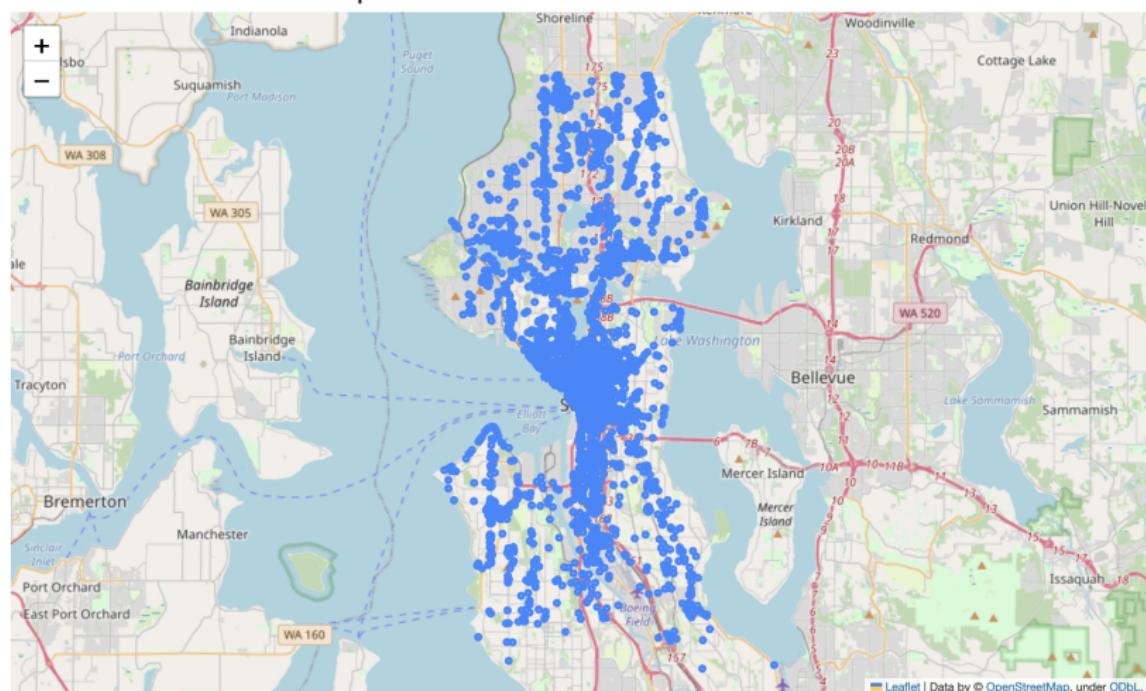
Mentor : Mohamed Laaraiedh

April 4, 2023

Exploration des données

Introduction

- Une carte montrant l'emplacement du bâtiment en bleu dans la ville de Seattle

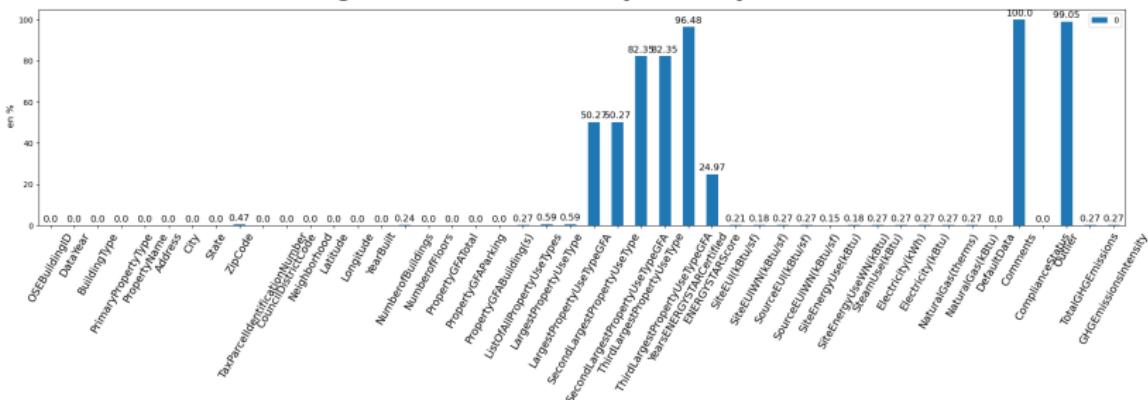


- L'objectif du projet est de prédire la consommation d'énergie et l'émission totale de C02 du bâtiment non résidentiel.

Taille d'une dataframe

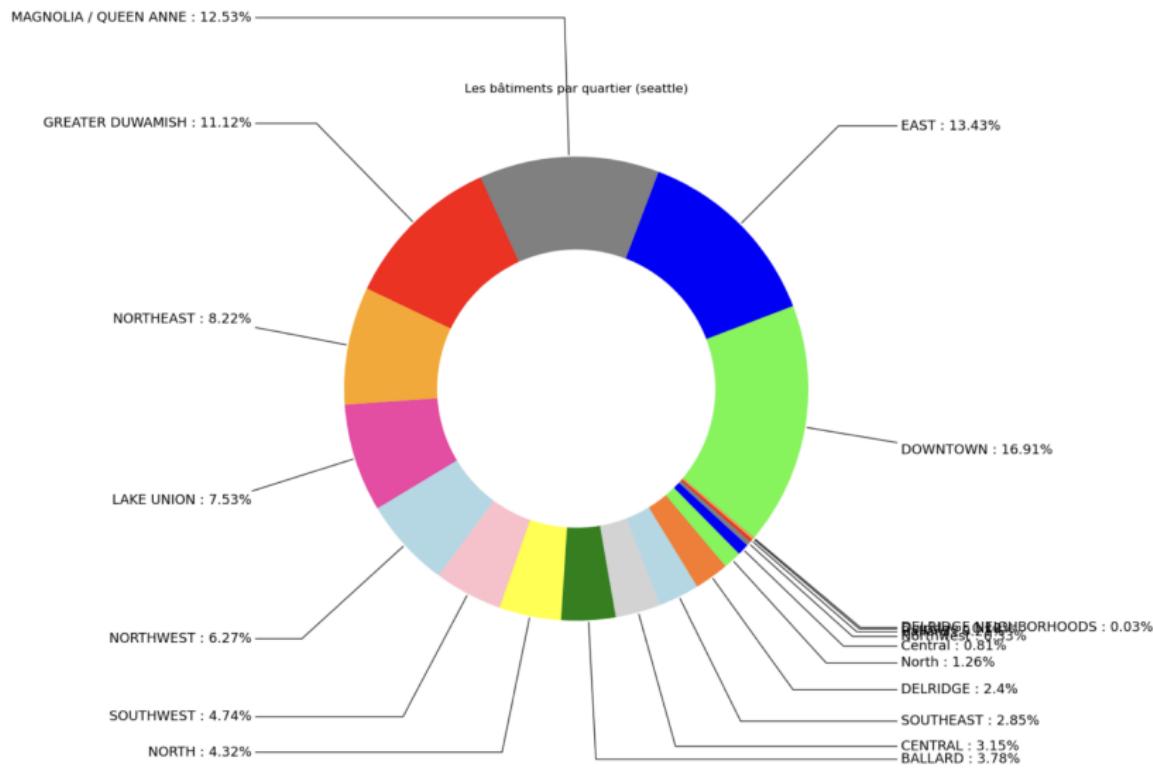
- 1 Le jeu de données contient 3376 lignes et 46 colonnes.
- 2 Le jeu de données contient aussi les valeurs manquantes.

Pourcentage des valeurs manquantes par variable

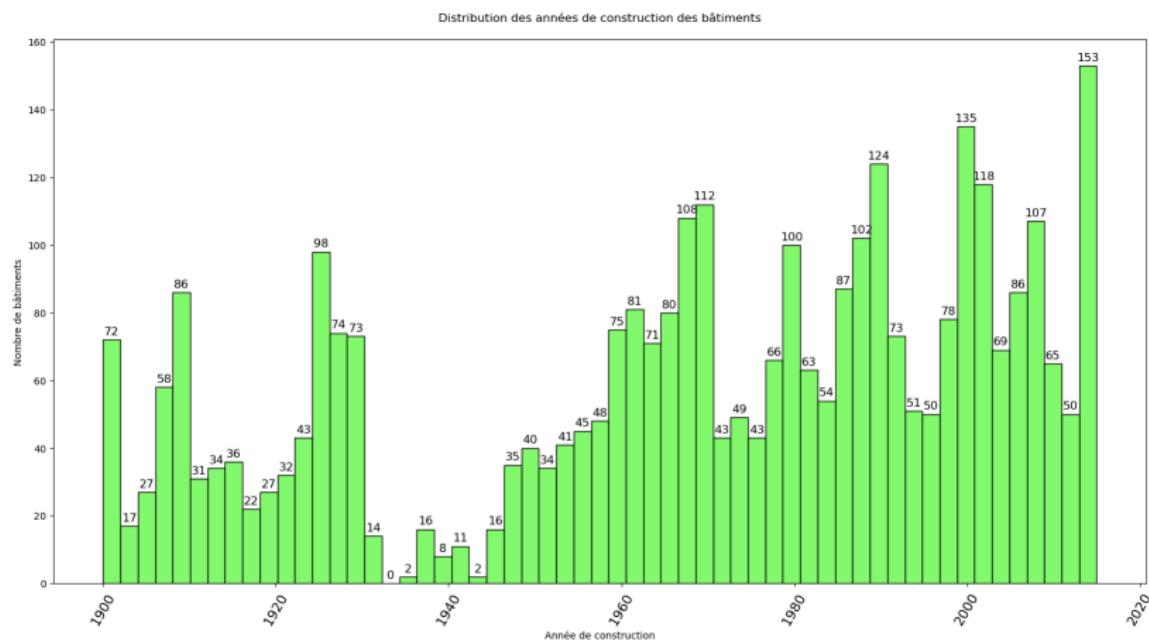


- 3 Le jeu de données utilise plus de 1,2 Mo de mémoire 1.2+ MB.

Bâtiment par quartier



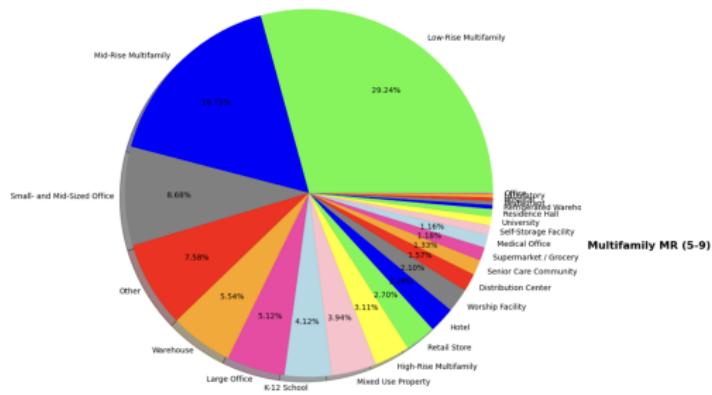
Distribution des années de construction des bâtiments



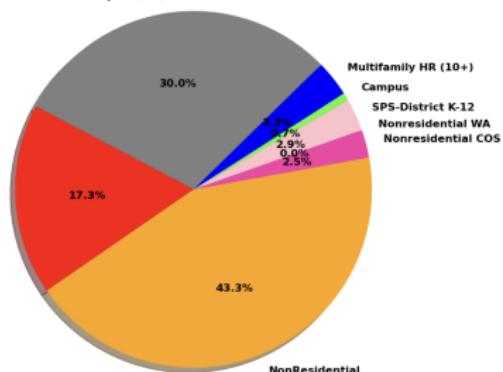
- On constate que la plupart des constructions ont moins de 50 ans et que le nombre de bâtiments augmente avec le temps.

Représentation des bâtiments par catégorie

- On voit que la majorité des immeubles ne sont pas résidentiels.



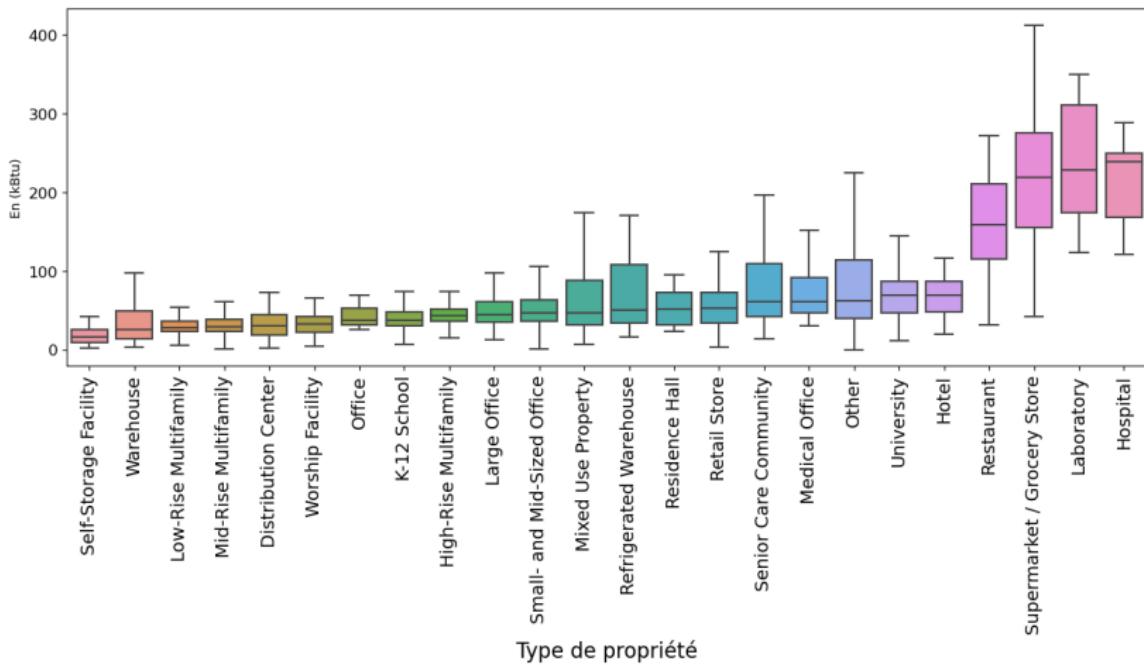
Répartition des types de bâtiments du Dataset
Multifamily LR (1-4)



- On voit que les bâtiments destinés à l'habitation sont moins de 35%.

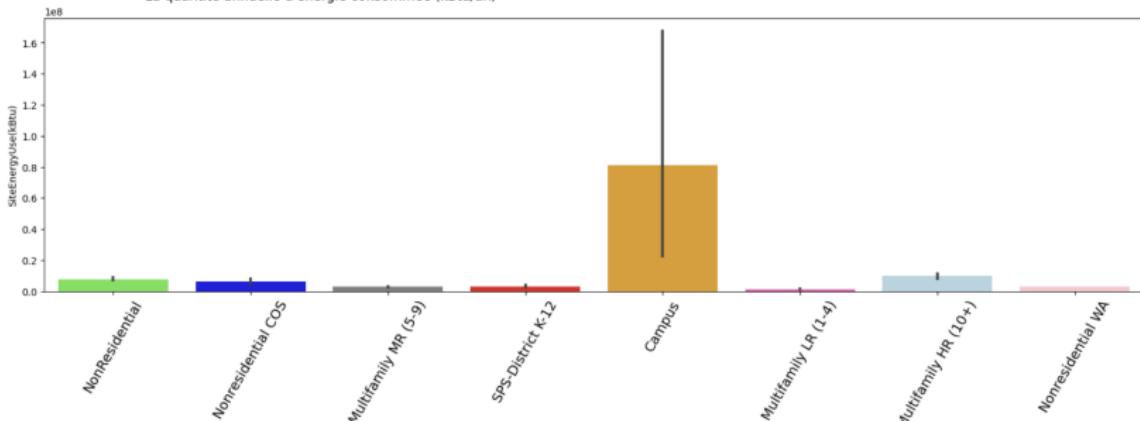
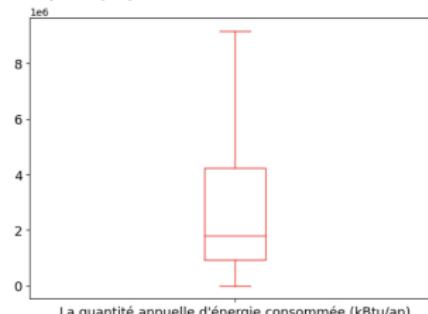
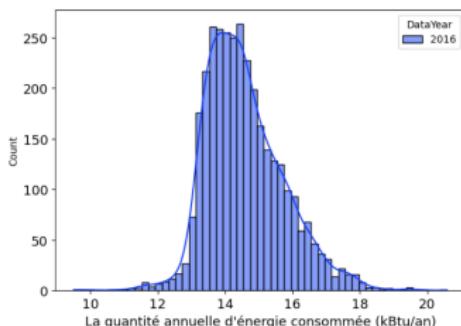
La quantité annuelle d'énergie consommée par pied carré et par type de bâtiment

La quantité annuelle d'énergie consommée par pied carré et par type de bâtiment

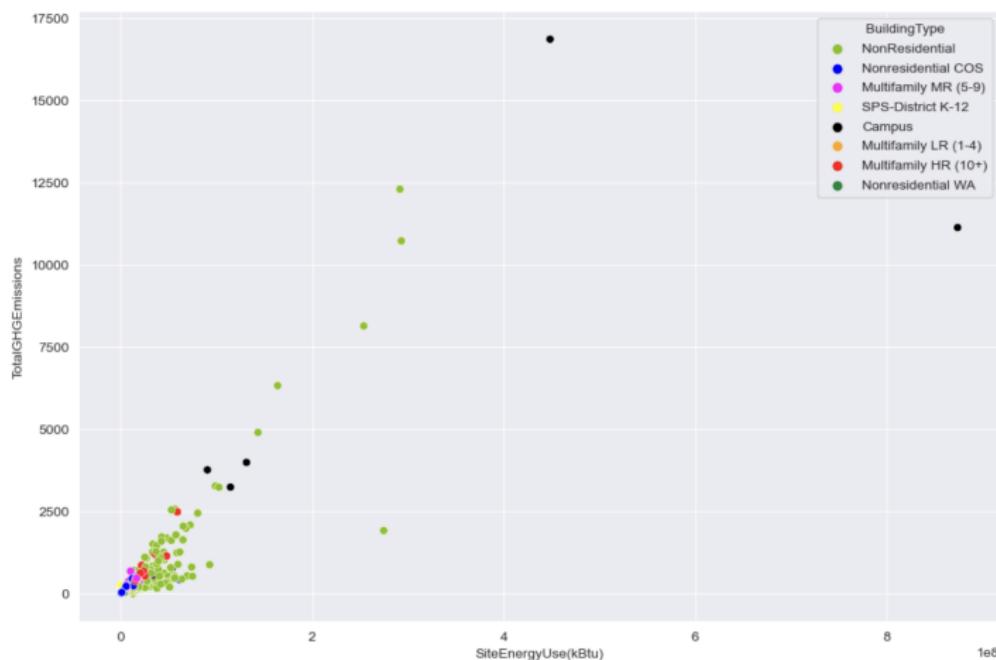


La quantité annuelle d'énergie consommée

La quantité annuelle d'énergie consommée par la propriété (kBtu/an)



Dépendance des émissions totales de la consommation d'énergie



- Il existe une dépendance linéaire des émissions totales sur la consommation d'énergie

Feature engineering

- 1 L'identifiant **outliers** correspond au fait qu'une propriété est une valeur aberrante haute ou basse. Pour nettoyer les données nous n'avons conservé que les bâtiments avec cette valeur manquante.
- 2 Nous avons effacé les lignes identifiées comme non conforme (c'est à dire les lignes où le **compliantstatus** n'est pas **compliant**, au total il ya 128 lignes).
- 3 Nous avons ensuite effacé les doublons sur l'identifiant **OSEBuildingID**.
- 4 Nous avons supprimé les variables redondantes (ex : **Gaz naturel(thermique)**, **electricité(kWh)** etc... grâce à la matrice de corrélation).
- 5 Les suffixes **WN:Weather Normalized** - Ce sont les mesures normalisées avec les conditions climatiques. Comme dans le cadre de notre analyse, la météo ne rentrera pas en compte nous avons effacé les variables avec les suffixes **WN**.
- 6 Comme l'objectif du project est de se concrétiser que sur les bâtiments non résidentiels, nous avons effacé tous les bâtiments avec le keyword **Multifamily**.
- 7 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 les avons donc supprimé.
- 8 Les variables avec suffix **GFA** pertinentes dans nos analyses présentent des fortes corrélations linéaires avec les variables à predire. Nous avons réduit cette correlation en les normalisant par la surface total **PropertyGFTotal**.
- 9 Nous avons aussi normalisé **Electricity,Naturalgas,SteamUse** par **SiteEnergyUse(kBtu)**.

Modélisation

Encodage et standardisation des données

- 1 Nous avons opté à l'utilisation de la validation croisée. L'objectif est de pouvoir vérifier la précision de votre modèle sur plusieurs sous-ensembles de données différents. Cela assure bien que notre model se généralise bien aux données et Cela améliore la précision du modèle.
- 2 Comme notre jeu de données contient des variables numériques et catégorielles, nous avons défini un pipeline qui nous permet d'encoder (en utilisant **OneHotEncoding**) des variables catégorielles et de standardiser les variables numériques en utilisant le **RobustScaler()**.
- 3 Nous avons separer notre données en jeux d'entraînement qui compte 80% et jeux de test qui compte 20% du jeux de données.
- 4 Nous avons utilisé les différents modèles puis nous avons choisi le plus efficace selon les métriques définies.
- 5 Pour expliquer la sortie du modèle sélectionné, nous avons utilisé le **Shap**.

Les modèles utilisés

1 Modèles linéaires:

- **Linear regression:** est un modèle dans lequel on trouve la ligne qui correspond le mieux aux données selon un critère spécifique
- **Elastic Net :** incorpore des pénalités de régularisation L1 et L2 :

$$\frac{\sum_{i=1}^n (y_i - x_j^i \hat{\beta})^2}{2n} + \lambda \left(\frac{1-\alpha}{2} \sum_{j=1}^m \hat{\beta}_j^2 + \alpha \sum_{j=1}^m |\hat{\beta}_j| \right)$$

En plus de définir et de choisir une valeur lambda, ElasticNet permet également d'ajuster le paramètre alpha où $\alpha = 0$ correspond à Ridge et $\alpha = 1$ à Lasso. On peut donc choisir une valeur alpha entre 0 et 1 pour optimiser ElasticNet.

- **Support Vector Regression:** est un algorithme d'apprentissage supervisé utilisé pour prédire des valeurs discrètes. Support Vector Regression utilise le même principe que les SVM. L'idée de base derrière SVR est de trouver la meilleure ligne d'ajustement. Dans SVR, la droite de meilleur ajustement est l'hyperplan qui a le nombre maximum de points.

2 Modèles non linéaires:

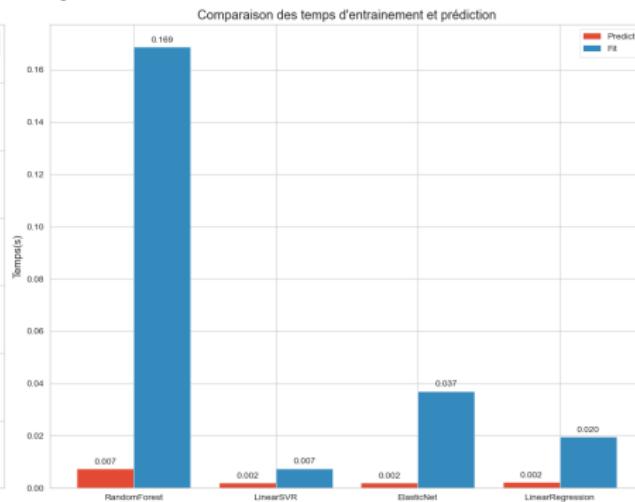
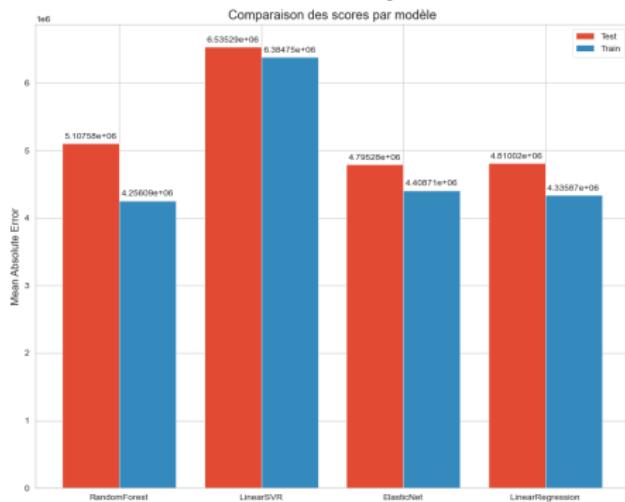
- **Les forêts aléatoires:** sont des algorithmes qui se basent sur l'assemblage d'arbre de décision indépendants. Chaque arbre traitant seulement une partie du problème grâce à un double tirage aléatoire : Un tirage avec remplacement sur les individus : C'est le tree bagging Un tirage aléatoire sur les variables : le feature sampling Au final, tous ces arbres de décisions indépendants sont assemblés. La prédiction faite par le random forest pour des données inconnues est alors la moyenne de tous les arbres dans le cas de la régression.

Comparaison des modèles

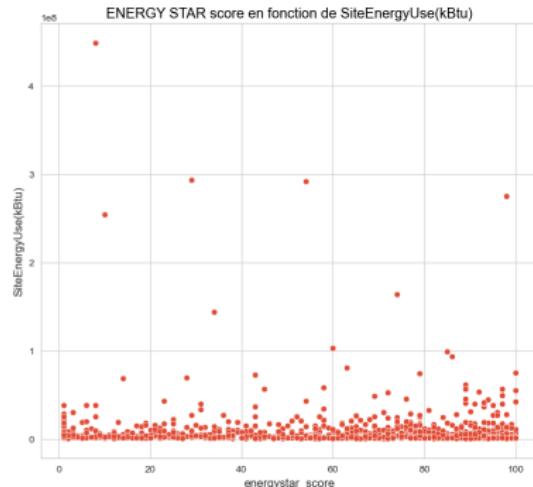
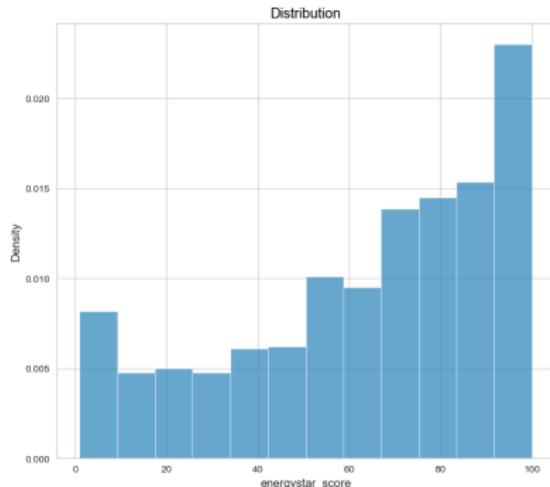
Comparons les métriques obtenues

	mean_fit_time	mean_score_time	mean_test_neg_mean_absolute_error	mean_train_neg_mean_absolute_error	MAE	R2
RandomForest	0.168824	0.007344	-5.107579e+06	-4.256089e+06	3.702201e+06	0.670703
LinearSVR	0.007486	0.002041	-6.535293e+06	-6.384745e+06	4.548969e+06	0.492921
ElasticNet	0.037101	0.002099	-4.795282e+06	-4.408708e+06	4.647208e+06	0.518582
LinearRegression	0.019592	0.002275	-4.810017e+06	-4.335867e+06	4.806691e+06	0.502797

Comparaison des scores par modèle



Comparaison des prédictions sans et avec energystarscore



Comparaison sans et avec EnergyStar Score

Out [68]:

Métrique	Sans ENERGY STAR	Avec ENERGY STAR
0 MAE	4954495.476102	4679609.574883
1 R ²	[0.4163873169379334]	[0.6109204623415326]

Explicabilité du modèle

On voit que les variables avec suffixe GFA contribuent beaucoup.

Diagramme de "Shap Values"

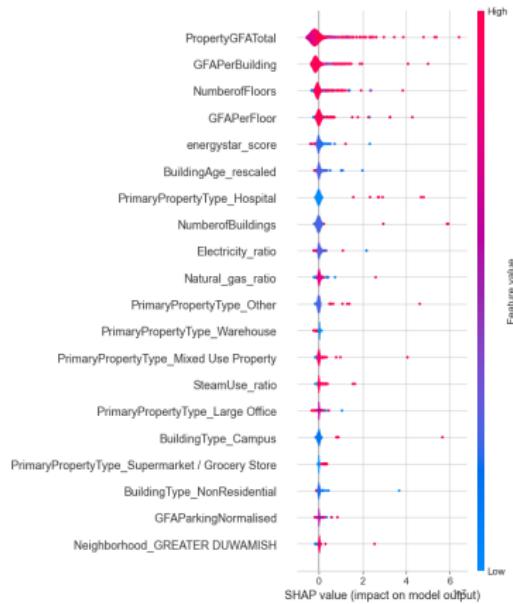
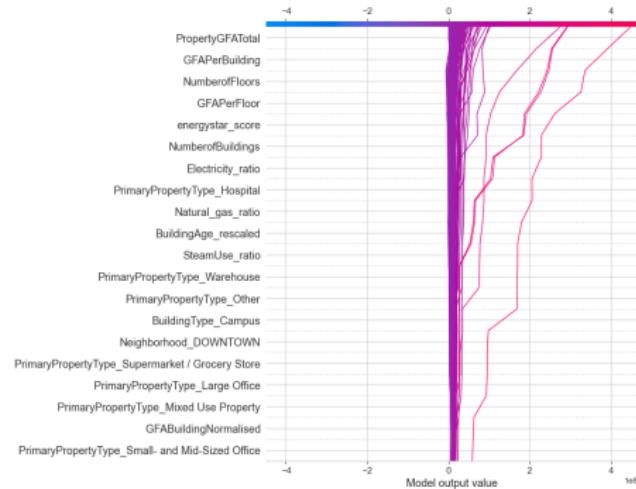
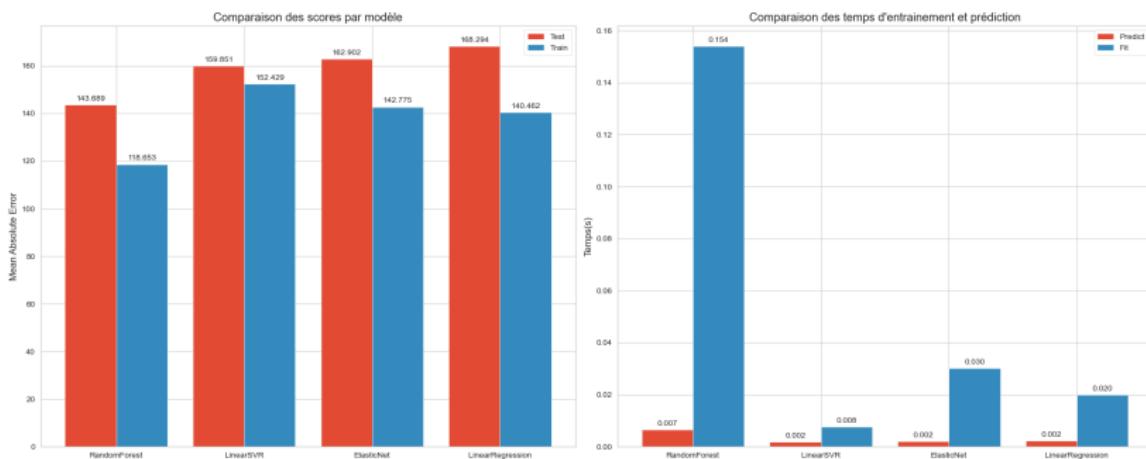


Diagramme de "decision tree"

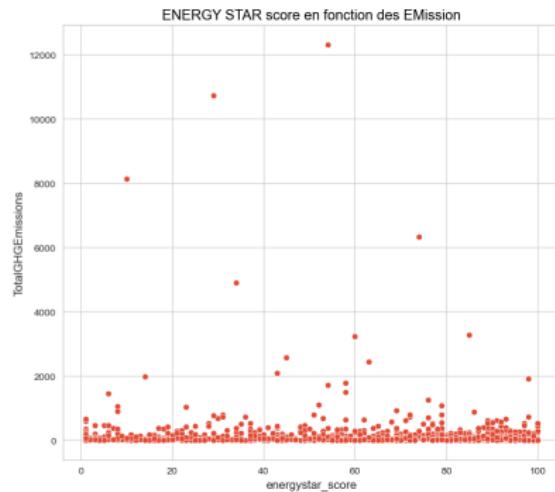
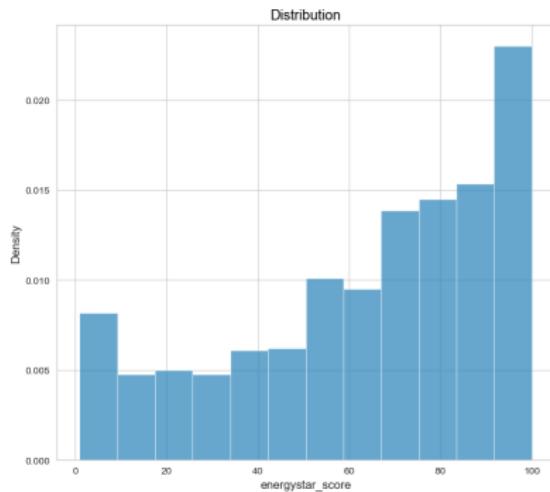


Comparaison des modèles

	mean_fit_time	mean_score_time	mean_test_neg_mean_absolute_error	mean_train_neg_mean_absolute_error	MAE	R2
RandomForest	0.154034	0.006594	-143.688859	-118.653185	101.206855	0.607342
LinearSVR	0.007707	0.001871	-159.850786	-152.429103	133.237417	0.262500
ElasticNet	0.030167	0.002056	-162.902469	-142.775466	152.476535	0.391908
LinearRegression	0.019808	0.002270	-168.294310	-140.461930	168.189578	-0.198132



Comparaison des prédictions sans et avec energystarscore



Out [57] :

Métrique	Sans ENERGY STAR	Avec ENERGY STAR
0	MAE	95.888099
1	R ²	[0.709204484178265] [0.6281993469897802]

Explicabilité du modèle

On voit que les variables avec suffixe GFA contribuent beaucoup.

Diagramme de "Shap values"

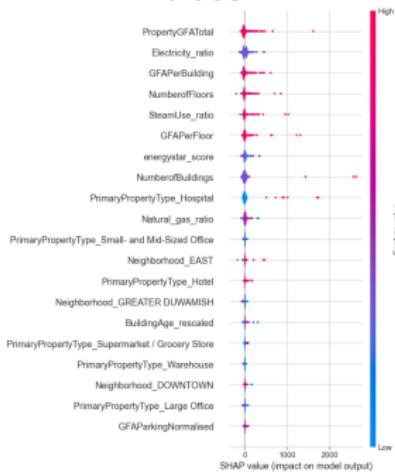
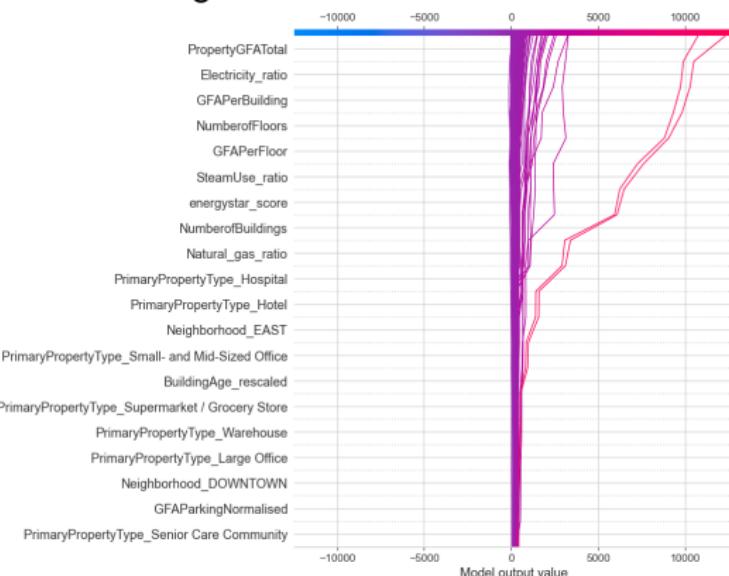


Diagramme de "Decision Tree"



Conclusions

Conclusions

- 1 Dans cette étude, nous avons commencé par faire une analyse exploratoire des données fournies
- 2 Nous avons procédé en sélectionnant des variables importantes pour notre étude et supprimer les autres.
- 3 Nous avons utilisé la technique de **Feature engineering** pour réduire la forte corrélation linéaire de certaines variables sur des variables à prédire.
- 4 pour pouvoir tester les modèles sur plusieurs données nous avons splité nos jeux de données en 80% pour le jeux d'entraînement et le reste pour le jeux de test, en plus de cela nous avons testé nos modèles sur de nombreux ensembles de données en utilisant la technique de validation croisée ce qui nous a permis d'augmenter la précision de nos modèles.
- 5 Pour **ensemble de données nettoyé**, nous avons utilisé différents modèles linéaires et non linéaires et sélectionné le modèle le plus performant.
- 6 Nous avons observé que les modèles non linéaires comme **RandomForestRegressor()** fonctionne mieux que tout autre modèle linéaire.
- 7 Nous avons observé que le **Energystarscore** servait à améliorer la précision de la prédiction dans le cas de **SiteEnergyUse(kBtu)** alors qu'il aggravait la prédiction dans le cas de **TotalGHGEmissions**.
- 8 Pour décomposer notre modèle, nous avons utilisé le **shap**, nous avons réalisé que les variables avec le suffixe **GFA** jouent un rôle important dans la prédiction de **SiteEnergyUse(kBtu)** et de **TotalGHGEmissions**.