

# “Anticipez les besoins en consommation de bâtiments”

Projet n°4

Léa ZADIKIAN  
Parcours Data Scientist  
26 Décembre 2022

# Objectif ville neutre en 2050 pour Seattle

- Objectif de la ville de Seattle : **ville neutre** en émissions de carbone en 2050.
  - Elle s'intéresse à la **consommation et aux émissions de ses bâtiments non destinées à l'habitation**
  - Elle dispose de relevés effectués en 2016, **MAIS** relevés coûteux !
- 
- **Mission** : prédire les émissions de CO<sub>2</sub> et la consommation totale d'énergie de bâtiments non destinés à l'habitation pour lesquels elles n'ont pas encore été mesurées.



**Seattle**

# Anticipez les besoins en consommation de bâtiments

- I. Présentation du jeu de données**
- II. Approche de modélisation et présentation des résultats**

# I. Présentation du jeu de données

# Environnement technique

- Notebook Jupyter 6.4.8
- Python 3.9.12
- Librairies utilisées :
  - pandas
  - numpy
  - missingno
  - matplotlib et seaborn
  - sklearn

# Jeu de données de la ville de Seattle



Seattle

- **3.376 lignes.** Chaque ligne est un **bâtiment**.
- **46 colonnes :**



Objectif du projet :  
**modélisation et prédiction**

- **Localisation du bâtiment** : adresse, quartier, coordonnées...
- **Année de construction**
- **Caractéristiques du bâtiments** : nb d'étages, surface, nombre de bâtiments etc...
- **Les différents usages** du bâtiments et la **surface** associée (bureau, supermarché, hôpital, hôtel, restaurant, parking etc...)
- **Consommation d'énergie**
- **Emission de CO<sub>2</sub>**

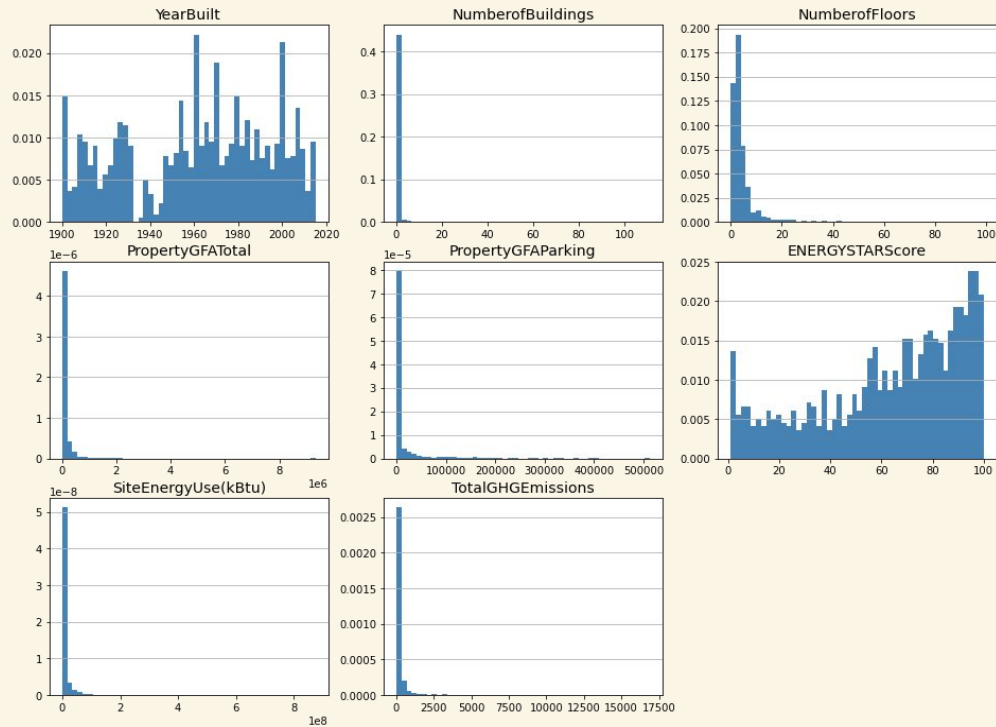
**Features**

**Targets**



# Analyse exploratoire

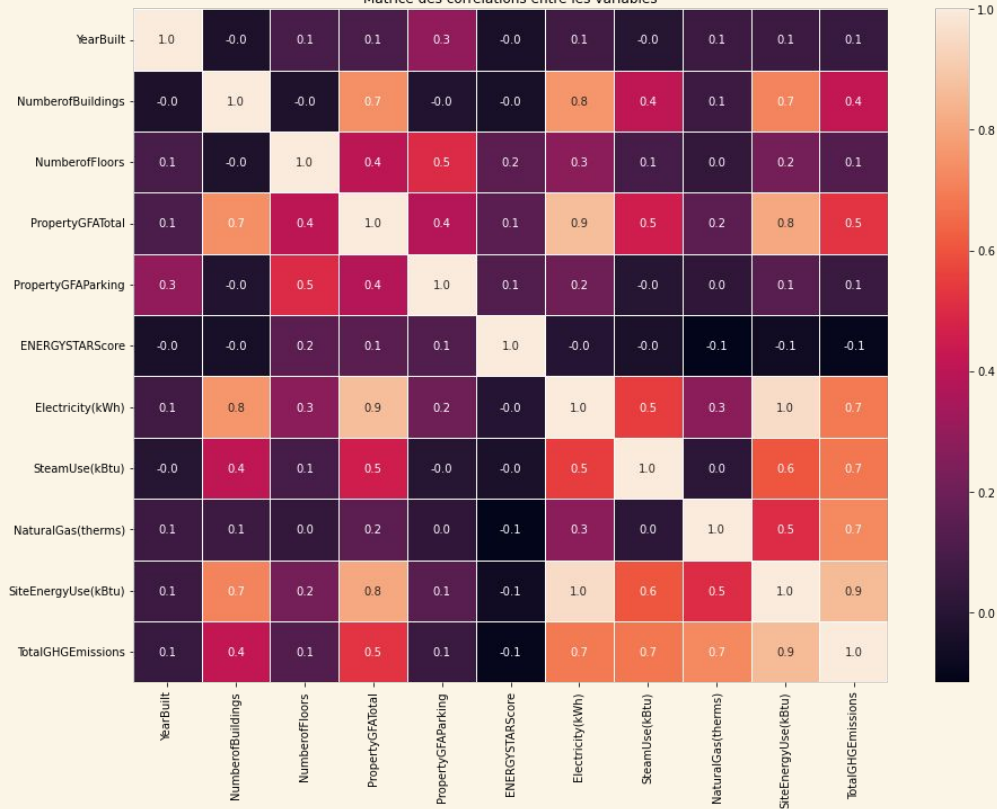
- Distribution des quelques variables principales





# Analyse des corrélations

Matrice des corrélations entre les variables



- Etude du **data leakage** ⇒ Suppression des variables corrélées avec les targets mais susceptibles de ne pas être disponibles dans le futur (risque de bonne performance du modèle d'entraînement, mais mauvaise en production) :
  - SteamUse
  - NaturalGas
  - Electricity...
- Variables corrélées avec la consommation :
  - La surface (0.8)
  - Le nb de bâtiments (0.7)
- Consommation et émission : 0.9
- ENERGY STAR Score : ne semble pas avoir de corrélation avec les autres variables.



## II. Approche de la modélisation et présentation des résultats



# Approche de modélisation

→ Itérations entre le feature engineering et l'entraînement des modèles

## 1. Feature engineering à partir de features "simples"

- 'BuildingType',
- 'PrimaryPropertyType',
- 'Neighborhood',
- 'YearBuilt',
- 'NumberOfBuildings',
- 'NumberOfFloors',
- 'PropertyGFATotal',
- 'PropertyGFAParking',

## 2. Prise en compte plus précise des différents usages d'un bâtiment et de la surface associée

- 'LargestPropertyUseType'
- 'LargestPropertyUseTypeGFA'
- 'SecondLargestPropertyUseType'
- 'SecondLargestPropertyUseTypeGFA'
- 'ThirdLargestPropertyUseType'
- 'ThirdLargestPropertyUseTypeGFA'

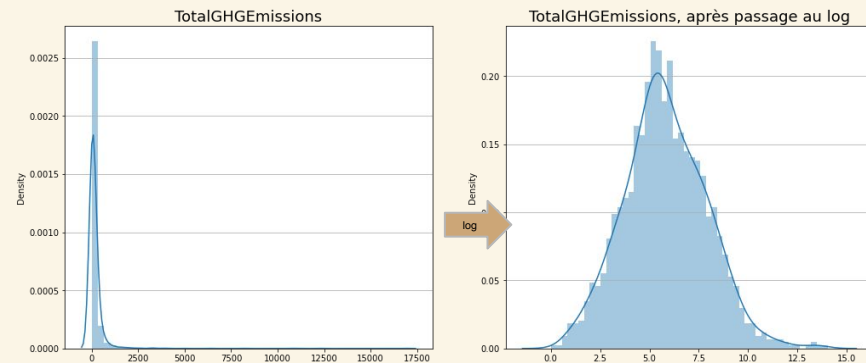
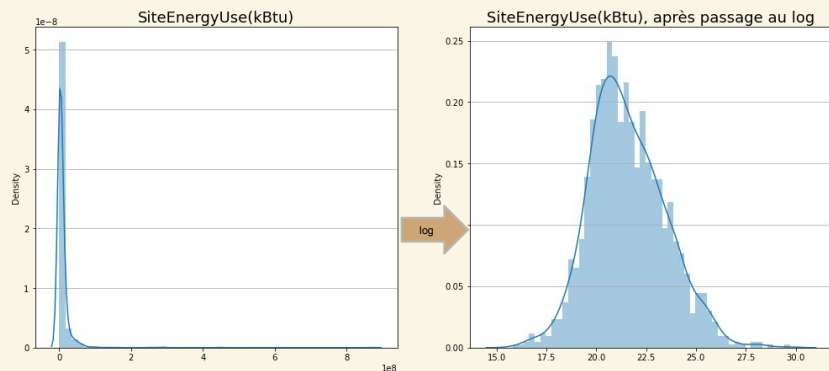
## 3. Intérêt de ENERGY STAR Score

- 'ENERGYSTARScore'
- Sur un sous ensemble du jeu de données sans valeurs manquantes

# 1<sup>ère</sup> itération - Feature engineering

→ **Transformation logarithmique** (*np.log2*) des variables numériques :

- Distributions initiales étalées à droite (skewness>0), type loi de puissance
- Après transformation, les données sont plus normalement distribuées.



→ **Categorical binning** (*pd.cut*) de la variable '**YearBuilt**' (années de construction) :

- Transformation de la variable continue en variable discrète en la catégorisant (par décennie)
- Réduit le risque de surapprentissage et améliore la robustesse du modèle

# 1<sup>ère</sup> itération - Feature engineering

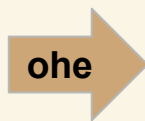
## → Encodage des variables catégorielles avec OneHotEncoder()

- Permet d'intégrer les variables catégorielles à la modélisation.
- Transforme une variable catégorielle en un tableau comportant une colonne binaire par modalité de la variable.
- Appliqué sur : '**BuildingType**', '**PrimaryPropertyType**', '**Neighborhood**', '**YearBuiltLabel**'

Variable catégorielle 'Neighborhood'  
Données d'origine, 10 dernières lignes.

Out[88]:

	Neighborhood
1537	NORTH
1538	BALLARD
1539	BALLARD
1540	EAST
1541	CENTRAL
1542	DELRIDGE
1543	DOWNTOWN
1544	MAGNOLIA / QUEEN ANNE
1545	GREATER DUWAMISH
1546	GREATER DUWAMISH



Variable catégorielle 'Neighborhood'  
Données transformées avec OneHotEncoder, 10 dernières lignes.

Out[85]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
1537	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
1538	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1539	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1540	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1541	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1542	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1543	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1544	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
1545	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1546	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

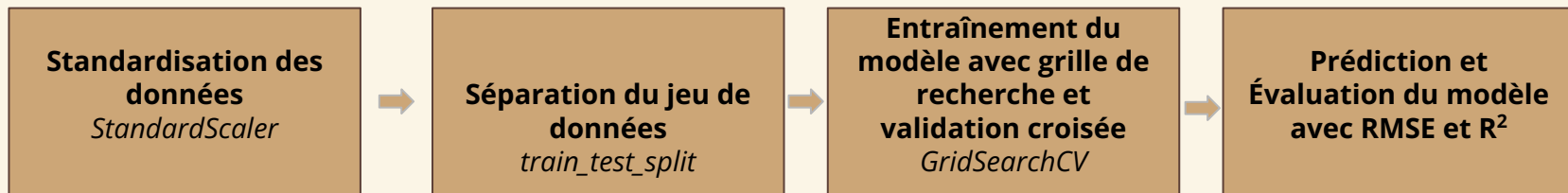
Variable "**Neighborhood**" comportant  
13 modalités

Création d'un tableau de 13 colonnes binaires

# Étapes de modélisation

## → 2 Targets :

- **SiteEnergyUse (KBtu)** (consommation d'énergie)
- **TotalGHGEmissions** (émissions de CO2)



## ■ Modèles utilisés :

- Dummy Regressor (baseline)
- Elastic Net (linéaire)
- Random Forest (non linéaire)

## ■ Métriques utilisées :

- **RMSE** : Root mean squared error
- **R<sup>2</sup>** : coefficient de détermination

# Approche naïve avec DummyRegressor

- Modèle utilisant une stratégie de prédiction très simple : prédiction par une constante ou par la moyenne, la médiane du jeu d'entraînement.
- Les résultats servent de “**baseline**” pour comparer avec des “vraies” régressions
- **Prédiction de la consommation d'énergie :**
  - stratégie de prédiction du Dummy Regressor = moyenne
  - $RMSE = 1,98$
  - $R^2 = 0$  (le modèle est équivalent à prédire par la moyenne)

# Modèle de régression linéaire avec ElasticNet

- L'ElasticNet permet de combiner la régression Ridge et le Lasso
- **Objectif** : Minimiser cette fonction de coût comportant 2 coefficients de régularisation :

$$1 / (2 * n\_samples) * ||y - Xw||_2^2 + \text{alpha} * l1\_ratio * ||w||_1 + 0.5 * \text{alpha} * (1 - l1\_ratio) * ||w||_2^2$$

- **Grille de recherche des 2 hyperparamètres par validation croisée avec GridSearchCV :**

- $\alpha : 10^{-4}, 10^{-3}, \dots, 10, 10^2$
- $l1\_ratio : 0.1, 0.2, 0.3, \dots, 0.9$

Prédiction de la CONSOMMATION (1ère itération)	Modèle
	Elastic Net (Linéaire)
Meilleurs hyperparamètres	<ul style="list-style-type: none"><li>• <math>\alpha = 0.01</math></li><li>• <math>L_1\_ratio = 0.9</math></li></ul>
RMSE	1.50
R <sup>2</sup>	0.50
Temps d'exécution	0.013 s



# Modèle non linéaire avec Random Forest

- **Random forest** : méthode ensembliste qui se base sur l'assemblage d'arbres de décisions (tree bagging+ feature sampling)
- Grille de recherche des hyperparamètres par **validation croisée avec GridSearchCV** :
  - `n_estimators` : 10, 50, 100, 300, 500
  - `min_samples_leaf` : 1, 3, 5, 10

Prédiction de la CONSOMMATION (1ère itération)	Modèle
	Random Forest (Non linéaire)
Meilleurs hyperparamètres	<ul style="list-style-type: none"><li>• <code>min_samples_leaf</code> = 5</li><li>• <code>n_estimators</code> = 100</li></ul>
RMSE	1.48
R <sup>2</sup>	0.53
Temps d'exécution	0.46 s

# 2<sup>ème</sup> itération - Feature engineering

- On complexifie le modèle en prenant en compte de façon plus précise les différents usages d'un même bâtiment et la surface associée.
- On s'appuie sur un encodage binaire avec OneHotEncoder().
- On ne garde que les catégories les plus fréquentes.
- On remplace les 1 par le % d'utilisation par rapport à la surface totale.

## Features concernées :

- *'LargestPropertyUseType'*,
- *'LargestPropertyUseTypeGFA'*,
- *'SecondLargestPropertyUseType'*,
- *'SecondLargestPropertyUseTypeGFA'*
- *'ThirdLargestPropertyUseType'*,
- *'ThirdLargestPropertyUseTypeGFA'*

# 2<sup>ème</sup> itération - Résultats pour la CONSOMMATION

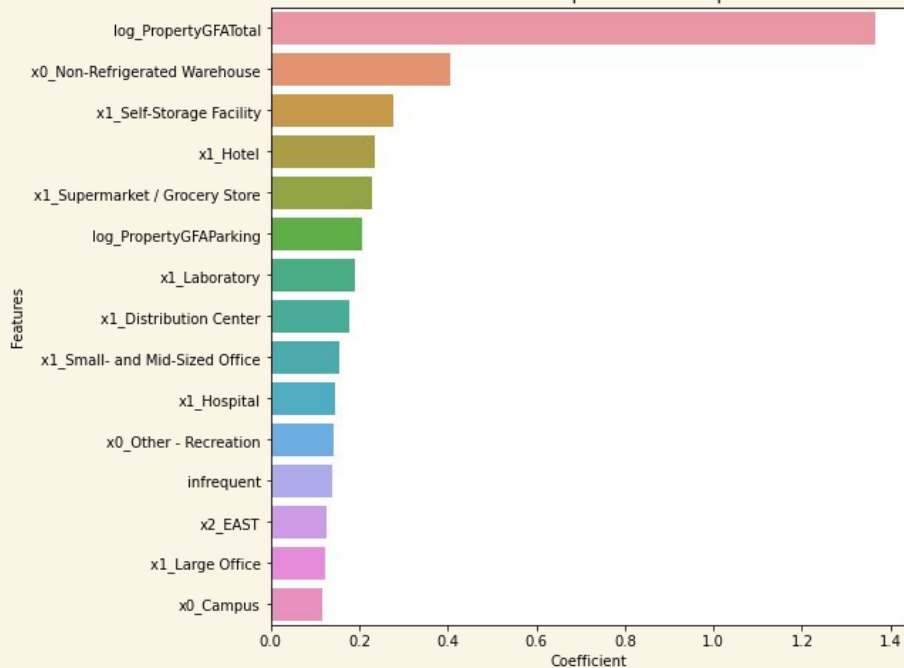
→ Résultats issus du 2<sup>ème</sup> feature engineering pour la CONSOMMATION

Prédiction de la CONSOMMATION (2 <sup>ème</sup> itération)	Modèle	
	Elastic Net ( <i>Linéaire</i> )	Random Forest ( <i>Non linéaire</i> )
Meilleurs hyperparamètres	<ul style="list-style-type: none"><li>• <math>\alpha = 0.01</math></li><li>• <math>L_1\text{-ratio} = 0.9</math></li></ul>	<ul style="list-style-type: none"><li>• <math>\text{min\_samples\_leaf} = 1</math></li><li>• <math>n\text{-estimators} = 100</math></li></ul>
RMSE	1.02	1.12
R <sup>2</sup>	0.58	0.64
Temps d'exécution	0.015 s	0.57s

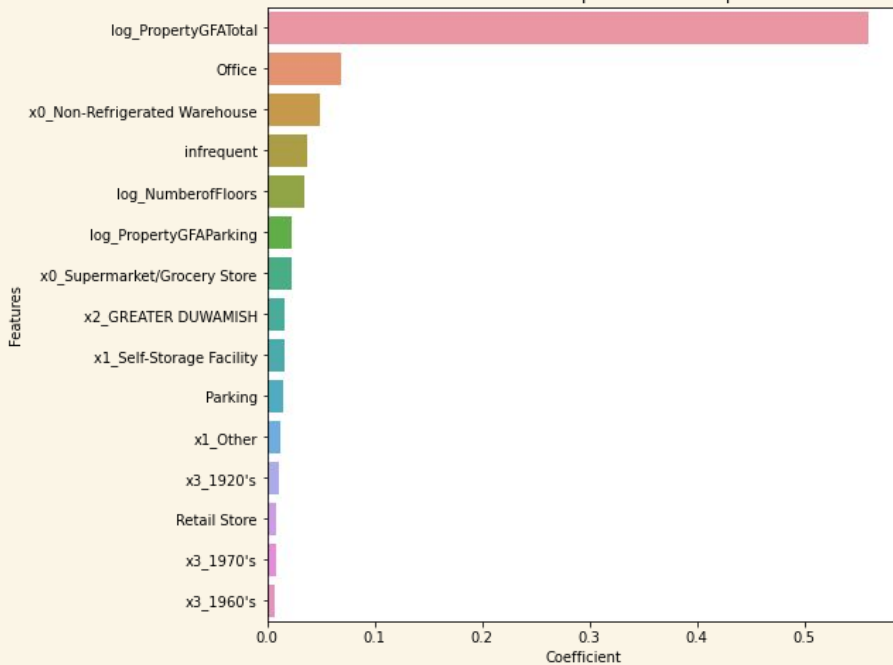
⇒ Le 2<sup>ème</sup> feature engineering a amélioré les performances de chaque modèle.

# 2<sup>ème</sup> itération - Feature importance CONSOMMATION

Elastic Net - Consommation - Importance des 15 premières Features



RandomForest - Consommation - Importance des 15 premières Features

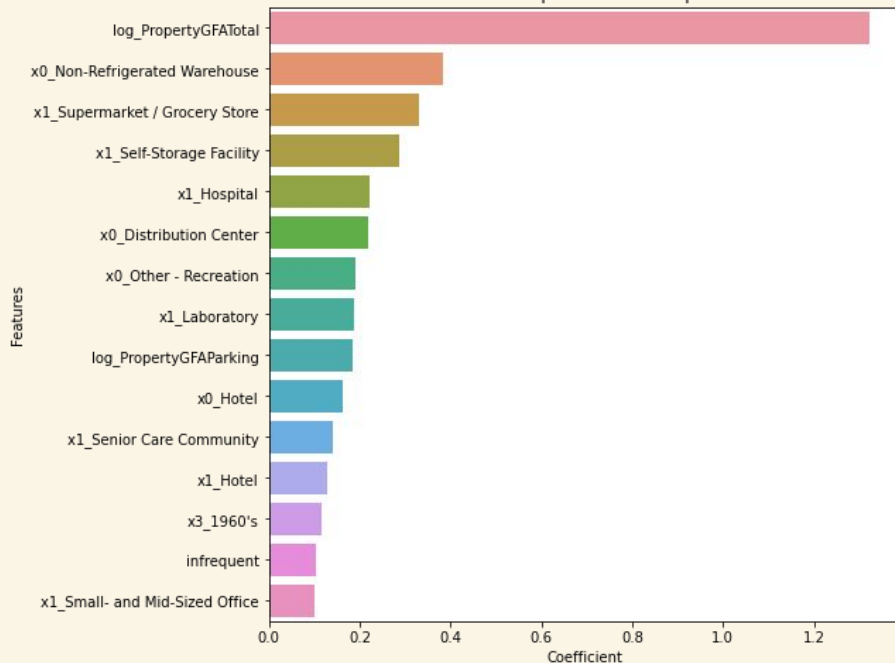


# Prédiction des émissions

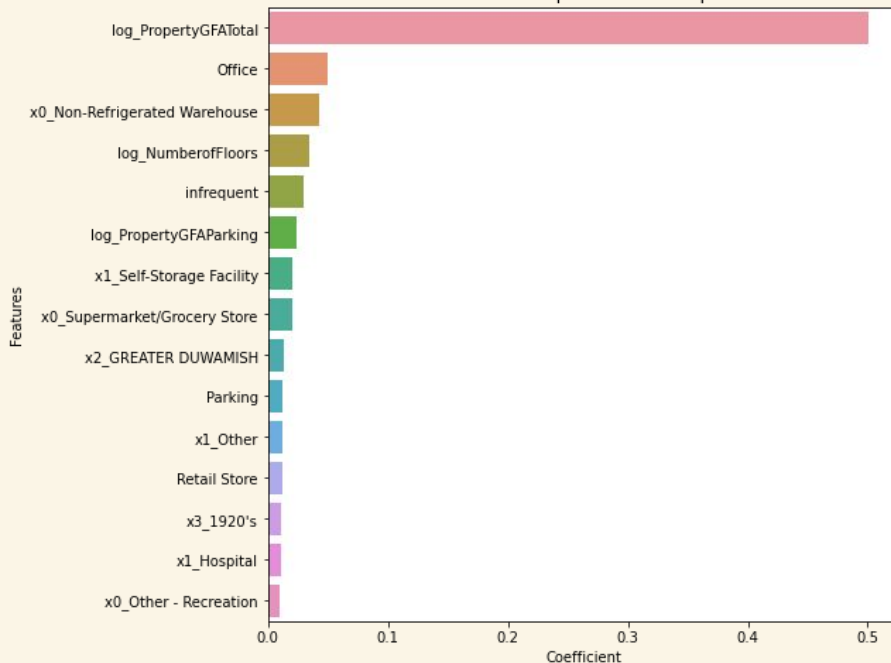
Prédiction les EMISSIONS	Modèle			
	Elastic Net (Linéaire)		Random Forest (Non linéaire)	
	1 <sup>ère</sup> itération	2 <sup>ème</sup> itération	1 <sup>ère</sup> itération	2 <sup>ème</sup> itération
Meilleurs hyperparamètres	<ul style="list-style-type: none"><li>• <math>\alpha = 0.01</math></li><li>• <math>L_1\text{-ratio} = 0.9</math></li></ul>	<ul style="list-style-type: none"><li>• <math>\alpha = 0.01</math></li><li>• <math>L_1\text{-ratio} = 0.9</math></li></ul>	<ul style="list-style-type: none"><li>• <math>\text{min\_samples\_leaf} = 3</math></li><li>• <math>\text{n\_estimators} = 500</math></li></ul>	<ul style="list-style-type: none"><li>• <math>\text{min\_samples\_leaf} = 3</math></li><li>• <math>\text{n\_estimators} = 500</math></li></ul>
RMSE	1.49	1.44	1.48	1.37
$R^2$	0.51	0.57	0.49	0.59
Temps d'exécution	0.013 s	0.014 s	0.48 s	0.60 s

# 2<sup>ème</sup> itération - Feature importance ÉMISSIONS

Elastic Net - Emission - Importance des 15 premières Features

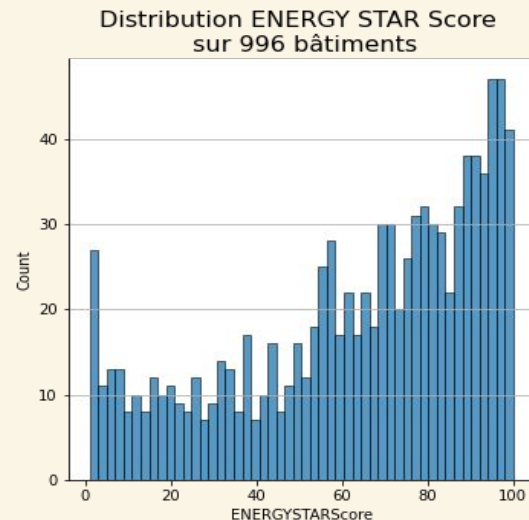


RandomForest - Emission - Importance des 15 premières Features



# 3ème itération : Intérêt de ENERGY STAR Score

- **ENERGY STAR Score** : score allant de 1 à 100 de la performance énergétique d'un bâtiment, prenant en compte ses caractéristiques physiques, son utilisation et le comportement de ses occupants.
  - **Score fastidieux à calculer** avec l'approche utilisée
  - On cherche donc à **évaluer son intérêt pour la prédiction d'émissions** ⇒ intégration dans la modélisation
  - L'ENERGY STAR Score **présente des valeurs manquantes** ⇒ on travaille sur un **sous-ensemble sans valeur manquante du jeu de données** de 996 bâtiments.
  
- Entraînement du modèle Random Forest :
  - R2 : 0.61 amélioration légère
  - ⇒ compromis à faire entre performance du modèle et complexité à calculer le score

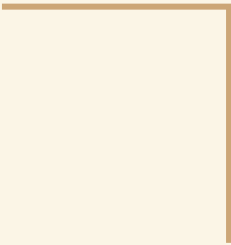


# Conclusion

Ce projet a été une opportunité de :

- Appréhender **l'approche de modélisation itérative** courante en Data Science : feature engineering / modèle d'apprentissage.
- Mettre en œuvre différentes **techniques de feature engineering**.
- Mettre en œuvre des **algorithmes de régression linéaire et non linéaire**, à l'aide de **grille de recherche et validation croisée**.
- **Évaluer les performances** de l'algorithme.





Merci pour votre attention !  
Des questions ?

