Projet n°4 : “Analyse de données de Consommation Electrique”

Soutenance de Jury - Projet n°4 - Octobre 2021

Ali Naama

Sommaire

I. Description de la problématique et exploration du jeu de données

Description de la problématique

Description :

La ville de Seattle cherche à connaître le niveau des émissions de CO2 et la consommation totale d'énergie électrique annuel des bâtiments de la ville.

Contraintes :

- Coût important d'obtention des relevés électriques
- Complexité de la collecte des relevés

Objectifs de l'étude :

Analyser les données de consommation électrique et prédire les émissions de CO2 ainsi que la consommation totale d'énergie sans accès aux relevés annuels :

- Mettre en place un modèle de prédiction réutilisable

❑ Pour ce projet, nous mettrons en place un pré-traitement des données et des prédictions de consommation électriques et d'émissions en C02.

Présentation du jeu de données

Informations sur les variables :

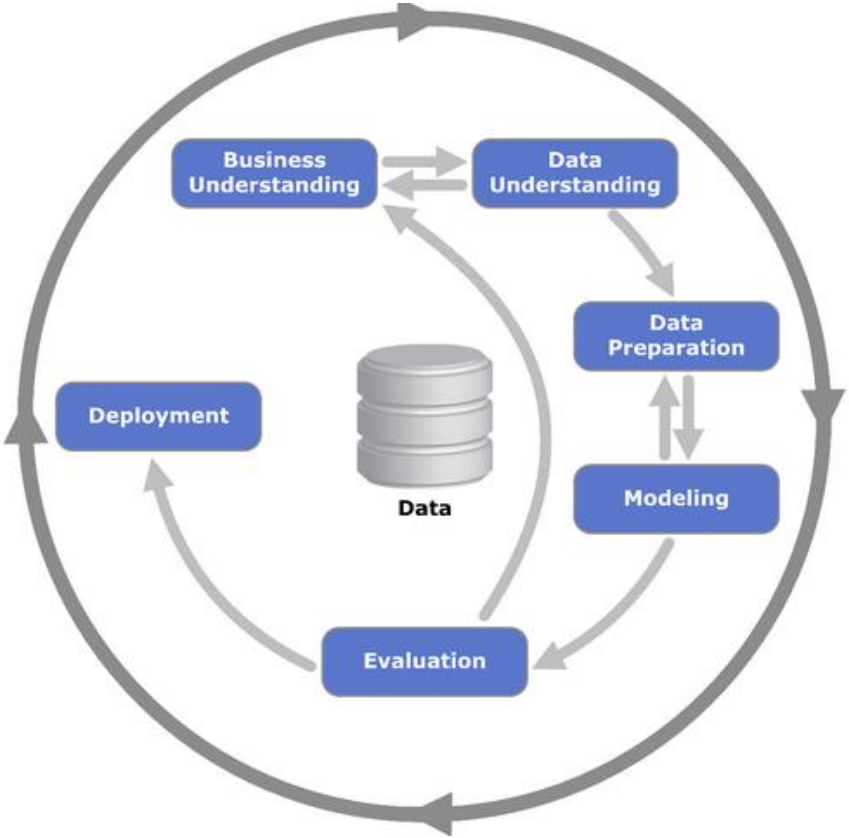
- Source des données : https://www.kaggle.com/city-of-seattle/sea-building-energy-benchmarking#2015-building-energy-benchmarking.csv
- Lien vers Score Star Energy : https://www.energystar.gov/buildings/facility-owners-and-managers/existing-buildings/use-portfolio-manager/interpret-your-results/what
- Lien vers la description des colonnes : https://data.seattle.gov/dataset/2016-Building-Energy-Benchmarking/2bpz-gwpy

2015-building-energy-benchmarking.csv

2016-building-energy-benchmarking.csv

- On dispose de deux fichiers csv

Méthodologie de Data Science « CRISP DM »



Présentation du jeu de données

Fichier	Nb lignes	Nb colonnes	Taux remplissage moyen	Doublons	Description
2015-building-energy-benchmarking.csv	3340	47	83.1%	0	2015-building-energy-benchmarking
2016-building-energy-benchmarking.csv	3376	46	87.2%	0	2015-building-energy-benchmarking

❑ On notera que l'on dispose d'un bon taux de remplissage des données qui est supérieur à 83%

Présentation du jeu de données – colonnes différentes : fichier 2015 et 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'], ['Address', 'City', 'State', 'ZipCode', 'Latitude',
 'Longitude', 'Comments', 'TotalGHGEmissions', 'GHGEmissionsIntensity'])
{'latitude': '47.61219025', 'longitude': '-122.33799744', 'human_address': '{"address": "405 OLIVE WAY", "city": "SEATTLE", "state": "WA", "zip": "98101"}'}
```

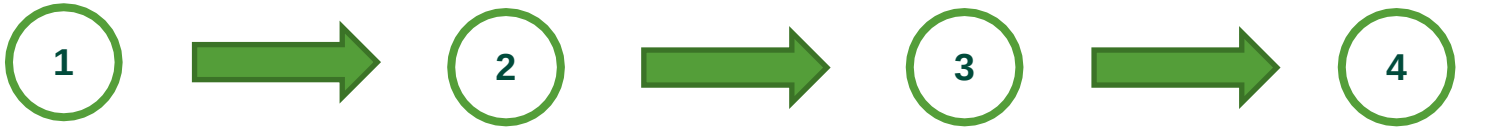
- On notera que le nombre de colonnes diffère

Outils utilisés pour l'analyse

Nom	Utilisation	Fonctions spécifiques
Anaconda 1.7.2	Gestion de package Gestion d'environnement virtuel	Conda : installation de package via le terminal
Jupyter Notebook 6.1.4	Structurer la démarche Executer code pas à pas Expliquer la démarche (markdown)	
PyCharm 2021.1	Test et développement	IDE Community Edition, Debug, Synchro Git
Python 3.9.5	Moteur Python Gestionnaire de librairies	Moteur d'exécution
Pandas 1.4.0	Librairie de manipulation de données Représentation des données	Manipulation de Dataframe : création, copie, filtres, tris, description, concaténation, pivotage, autre
Matplotlib 3.5.1 **Seaborn 0.11.2** **Numpy 1.22** **scikit-learn 1.0.2** **xgboost 1.5.2**	Génération de graphiques Gestion des densités de probabilité Machine Learning	Barplot, Scatterplot, lineplot, distplot, heatmap Calcul statistique

II. Analyse Exploratoire

Stratégie d’analyse



Découvrir les données

Analyse des données des 2 fichiers

Données manquantes ?

Comparaison des colonnes des datasets

Travaux de Nettoyage et d’harmonisation des colonnes

Analyse Exploratoire / Feature Engineering

Les types de Bâtiments

Les années de Construction

Les corrélations Age / Consommation / Emission CO2

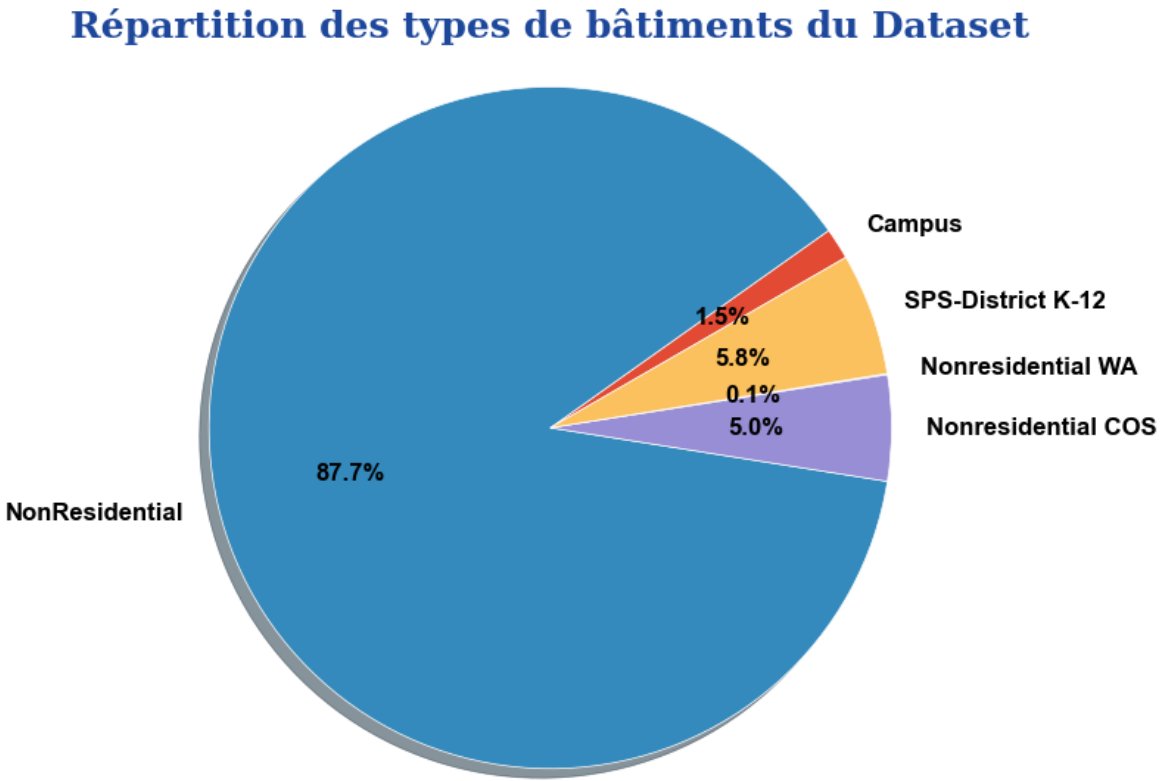
Comparer

Modélisation et Test des Modèles les plus pertinents
Prédictions et Résultats

Modélisation et constats

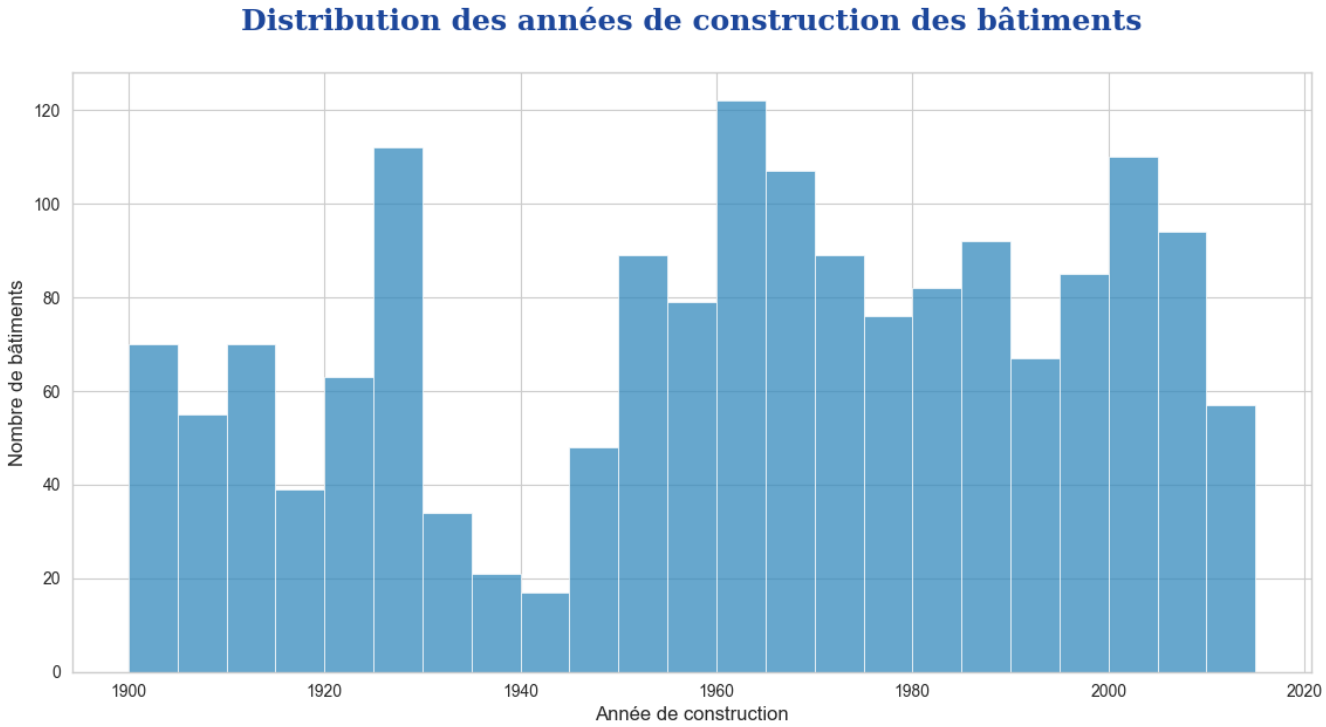
Synthèse des prédictions et de l’analyse

Type de Bâtiments identifiés



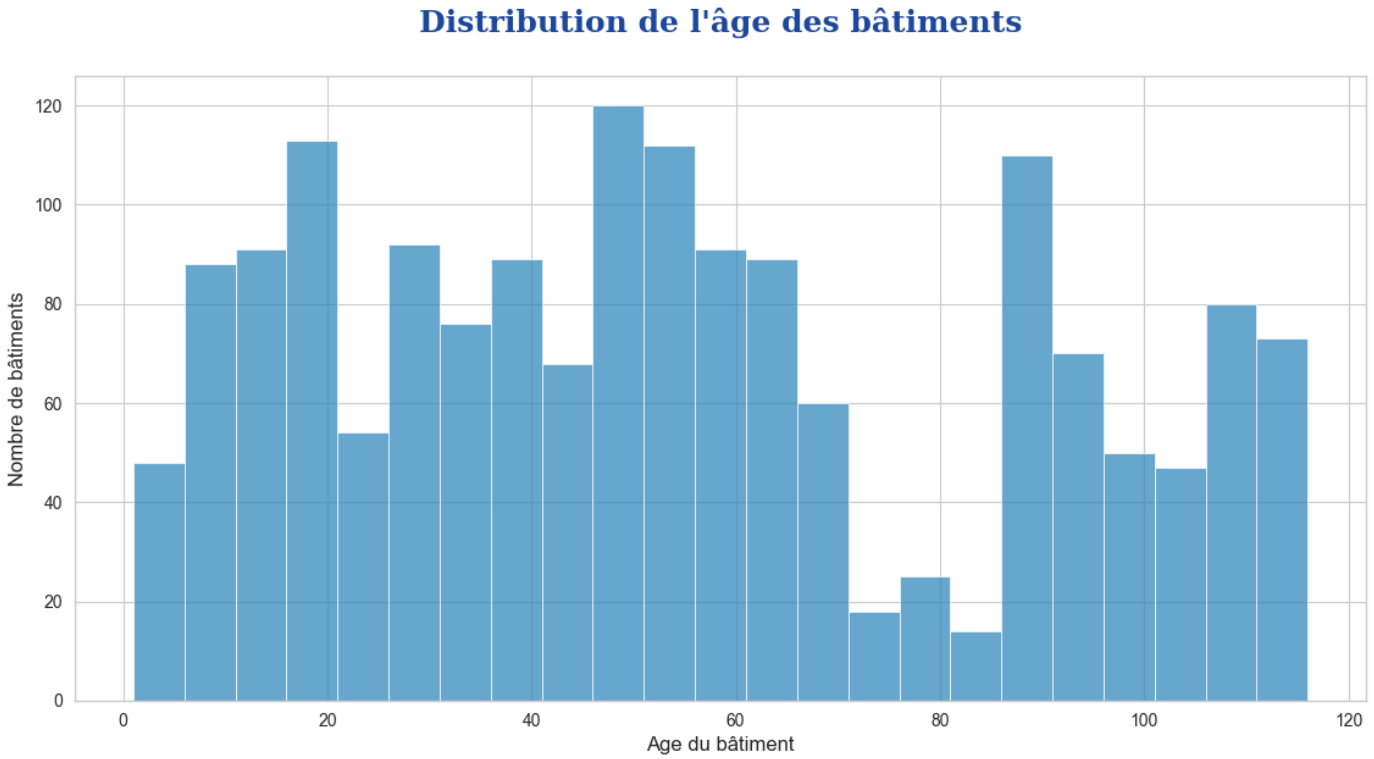
Années de construction

Seattle



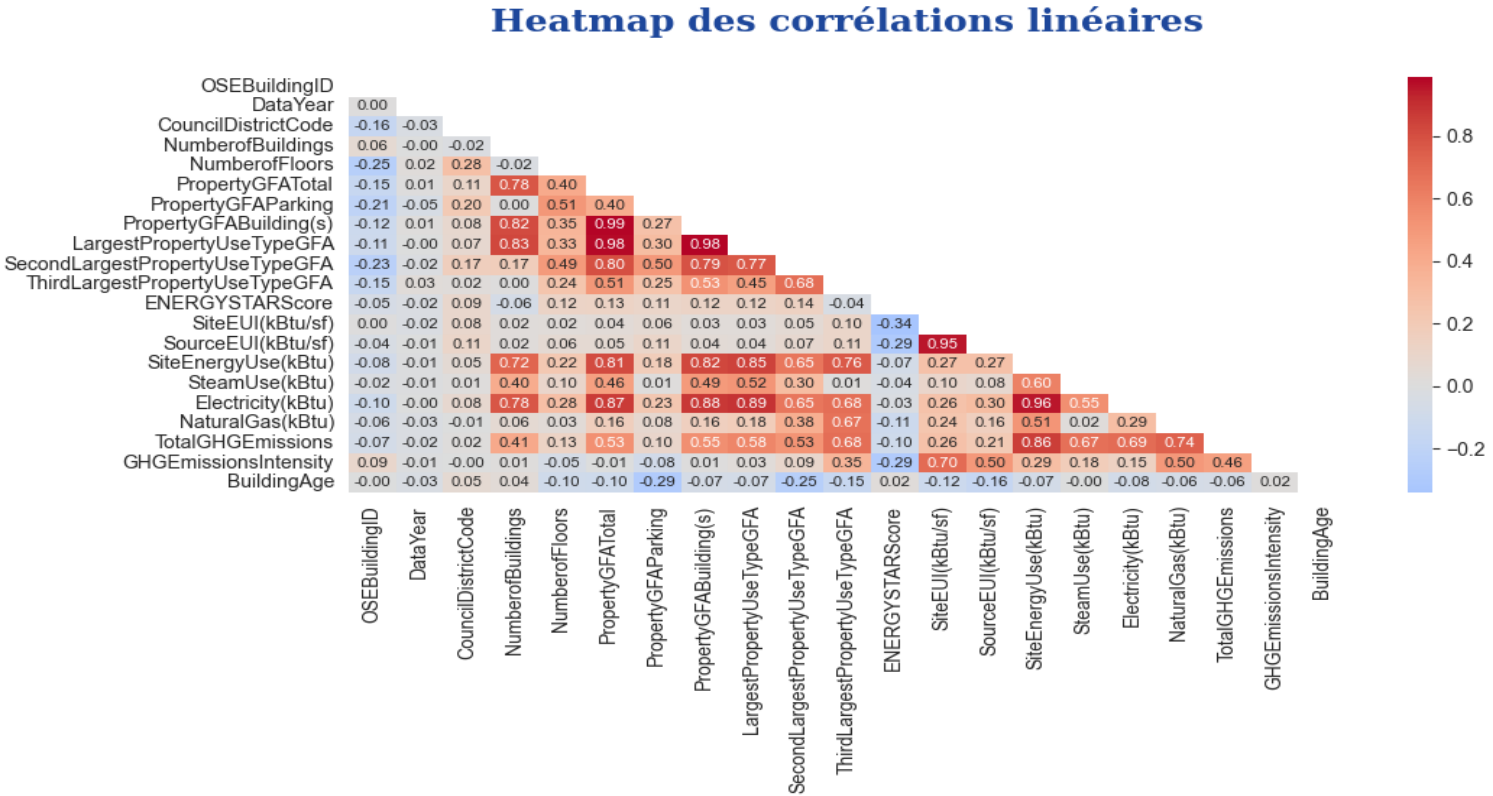
Age des Bâtiments

Seattle



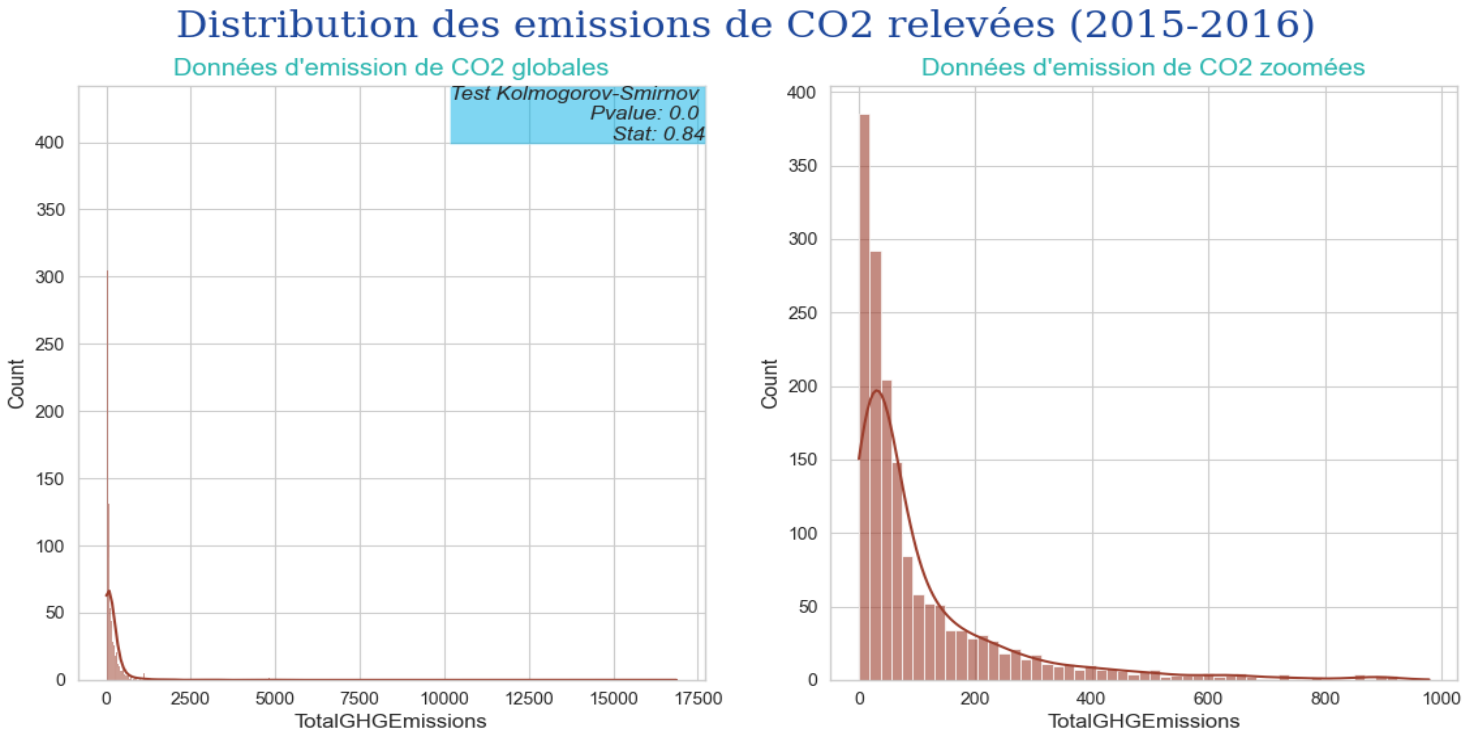
Carte thermique des corrélations

Seattle



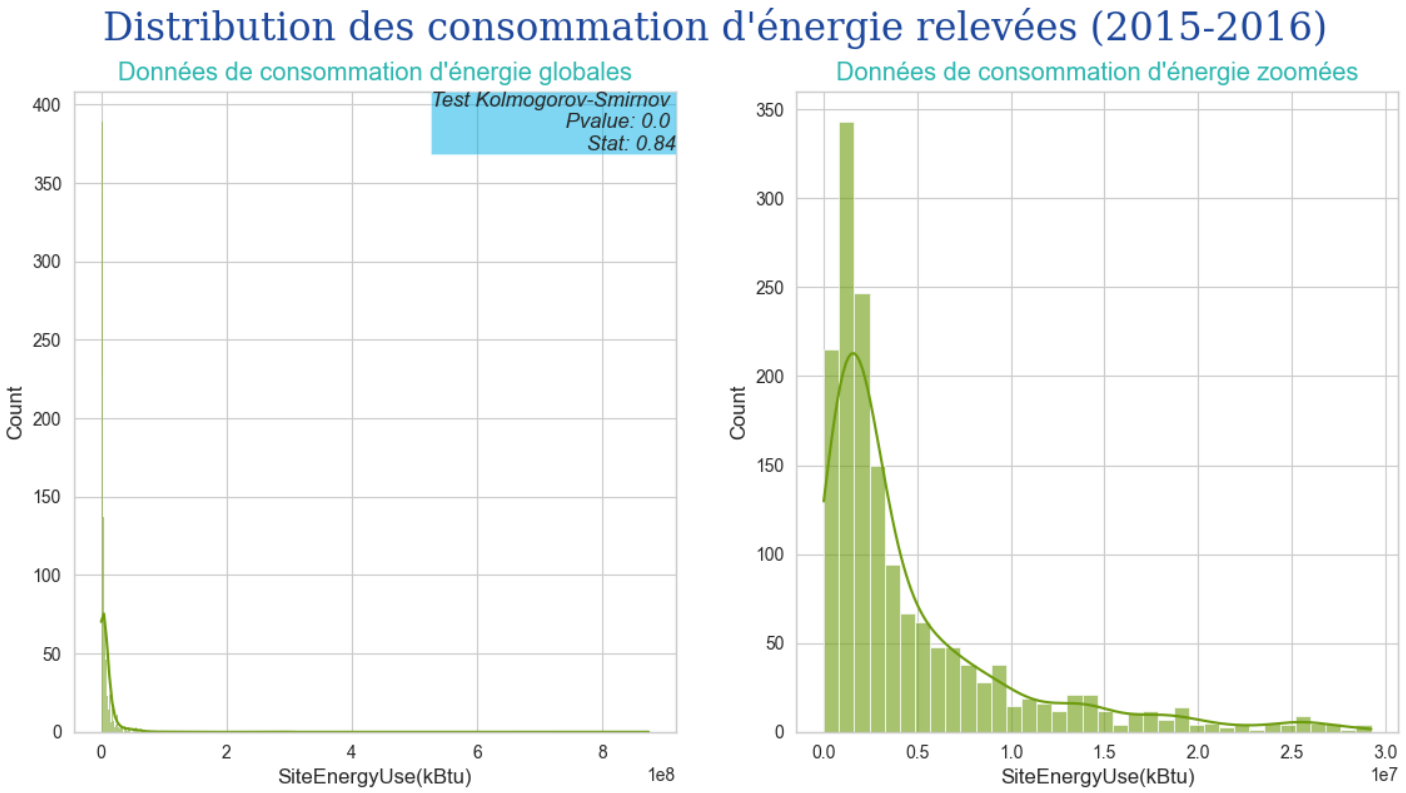
- On notera que pour les variables à prédire **TotalGHGEmissions** et **SiteEnergyUse(kBtu),** il y a des corrélations linéaires quasi similaires avec les variables de relevés (les consommations) mais également avec le nombre de bâtiments ou d'étages ains que les surfaces au sol.

Emissions de CO2



Consommation d'énergie

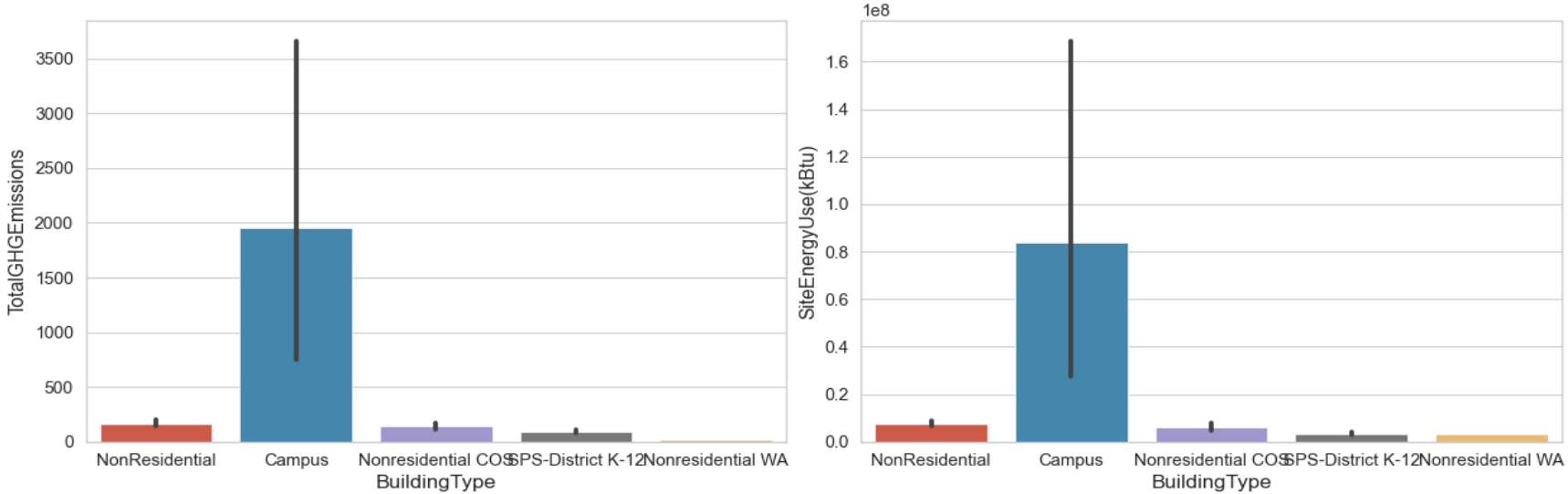
Seattle



Consommation et Emissions / Type de Bâtiment

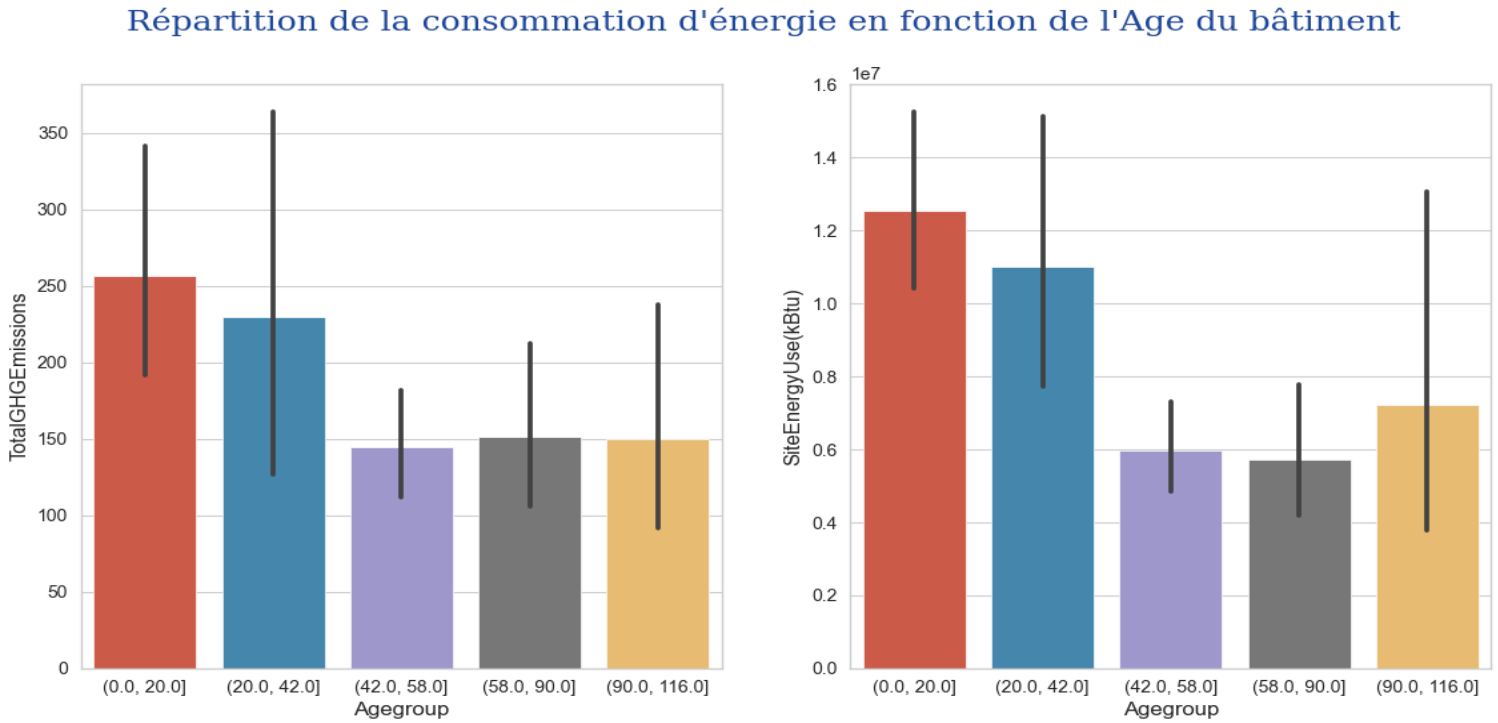
Seattle

Répartition de la consommation d'énergie et emissions de CO2 en fonction du type de bâtiment



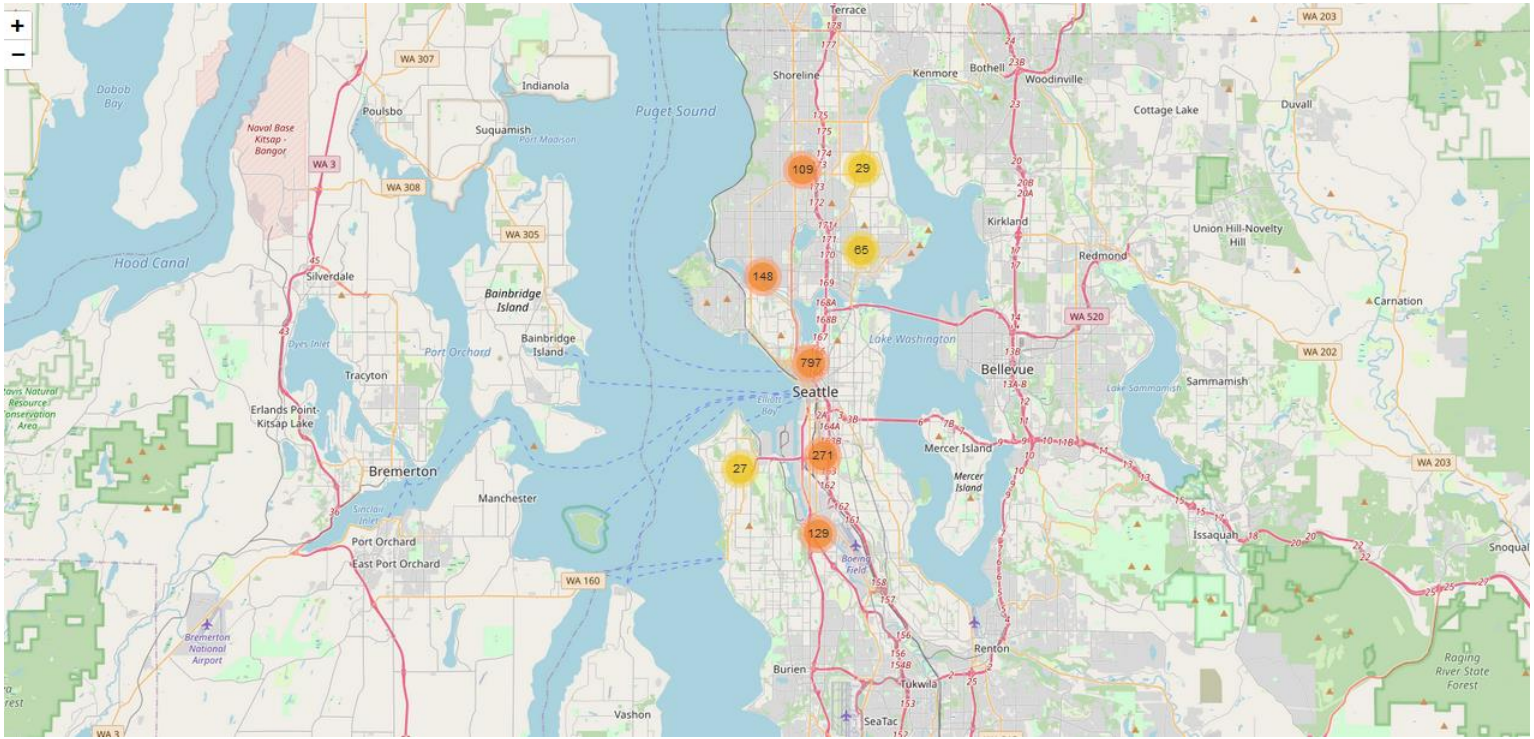
Consommation et Emissions / Age des Bâtiments

Seattle



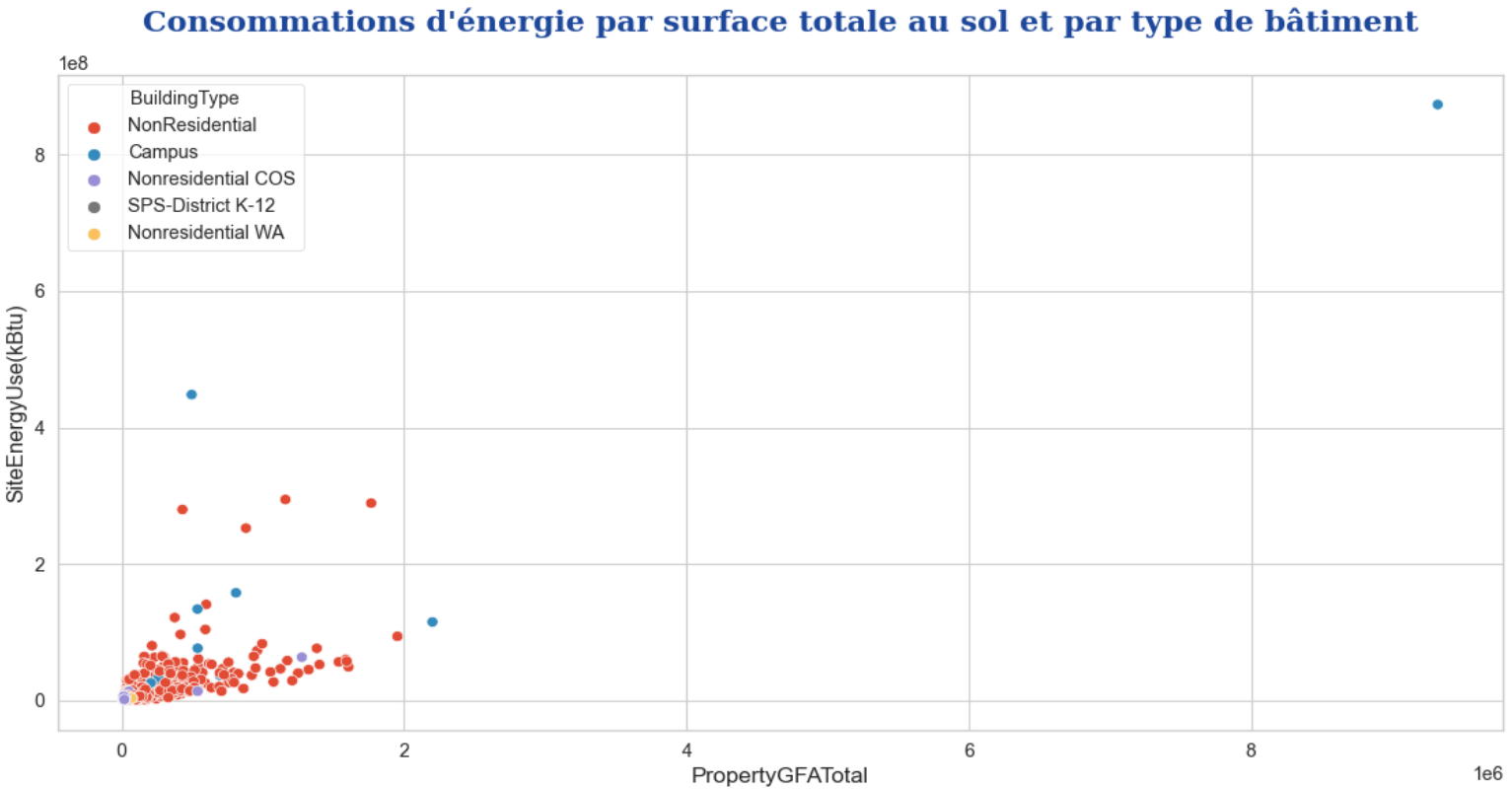
❑ On notera que les nouveaux bâtiments sont plus consommateur que les anciens bâtiments.

Carte des établissements sur la carte de Seattle

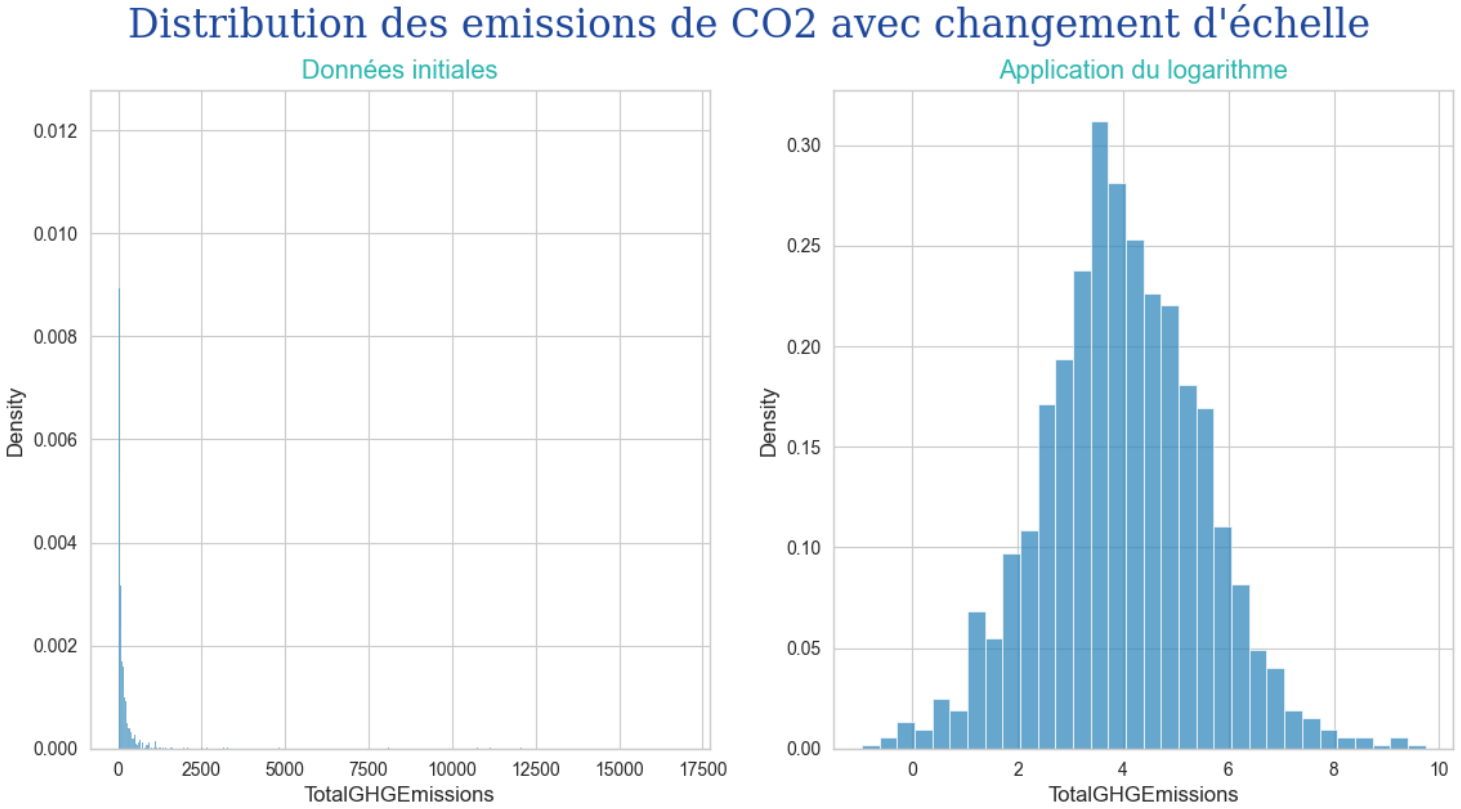


III. Analyse

Analyse de la consommation d'énergie par surface totale / par type de bâtiment



Distribution des émissions de CO2



Résultats des prédictions par Modèle

Model	RMSE with Train Data	RMSE with Test data	Tps d'exécution en seconde
Random Forest Model	485.9249807359694	626.4899003714745	0.036 s.
Linear Model	626.5869694012588	1537.626376198886	0.004 s.
ElasticNet	629.3093290712233	1755.2699957263153	0.004 s.
Support Vector Regression	664.1398510167052	1756.0641292602022	0.0039 s.
XGBoost	783.8531484762003	811.4033899367168	0.014 s.

```
# 5. Modèle non-linéaires : RandomForestRegressor

from sklearn.ensemble import RandomForestRegressor

regr = RandomForestRegressor(max_depth=2, random_state=0)
model_RFS = regr.fit(X_train, Y_train['TotalGHGEmissions'])

# predict the target on train and test data
predict_train_rfs = model_RFS.predict(X_train)
predict_test_rfs = model_RFS.predict(X_test)

# Root Mean Squared Error on train and test date
print('Random Forest Model :')
print('RMSE on train data: - RFS: ', mean_squared_error(Y_train['TotalGHGEmissions'],
predict_train_rfs)**(0.5))
print('RMSE on test data : - RFS: ',  mean_squared_error(Y_test['TotalGHGEmissions'],
predict_test_rfs)**(0.5))
start_time = time.time()
XGB_pred = model_XGB.predict(X_test)
print("Temps d'execution de l'agorithme model_XGB : {:.2} s.".format((time.time() -
start_time)))
```

Sélection de Modèle

- **Sélection des meilleurs modèles**

- Sur les 4 modèles testés, les modèles linéaires retournent de moins bonnes métriques en général pour l'erreur **quadratique moyenne** : **RMSE**.

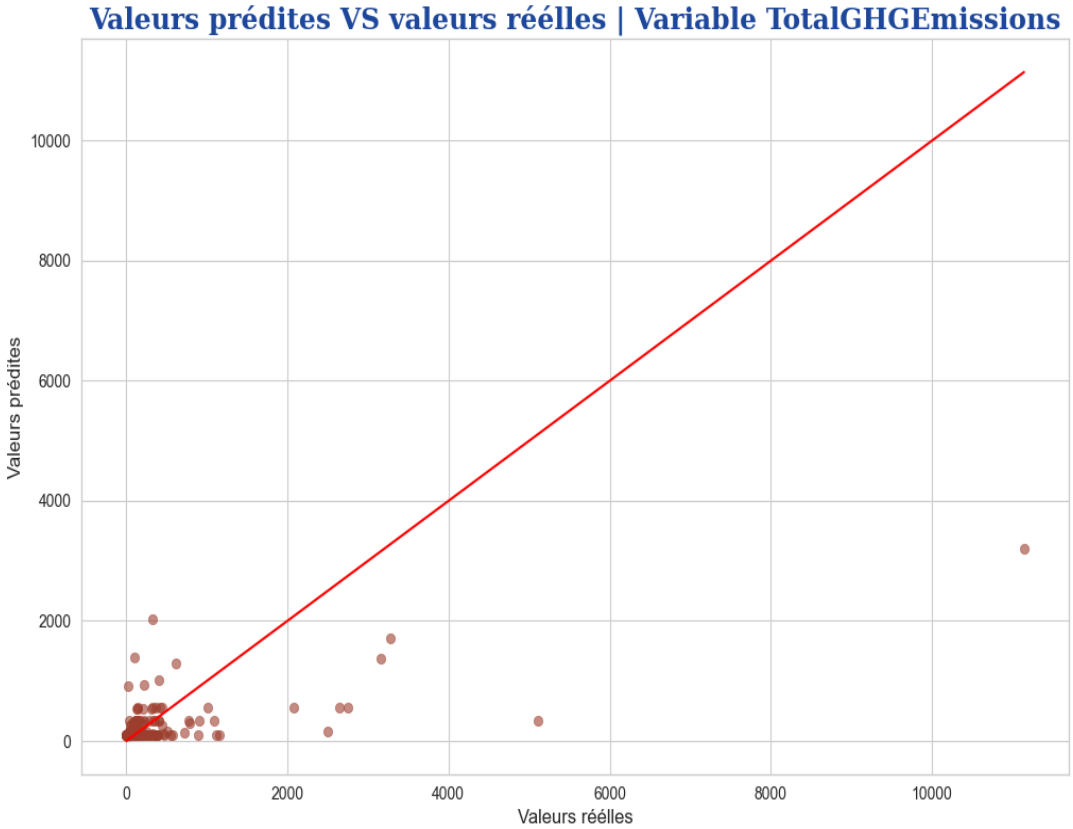
- Si nous prenons en considération le score MAE, qui aura du sens sur les modèles linéaires et non-linéaires, **les algorithmes XGBoost et RandomForestRegressor offrent des performances à peu près similaires pour la qualité des prédictions** mais les temps de calculs sont meilleurs sur le modèle RandomForestRegressor.

Prédictions

```
#Calcul des métriques pour les émissions de CO2
SEUmetrics =
metrics_model(Y_test['SiteEnergyUse(kBtu)'],RFM_pred)
print(SEUmetrics)

#Affichage des valeurs prédites vs valeurs réélles pour
émissions de CO2
plot_pred_true(Y_test['TotalGHGEmissions'],RFM_pred,
color="#9C3E2D", title="TotalGHGEmissions")
```

	Métrique	Résultats
0	MAE	1.101020e+07
1	R^2	-4.672975e-02



- Le modèle est performant en entrainement mais ne parvient pas à généraliser sur le jeu de test. Il faut affiner les analyses sur une autre variable, Type de bâtiment , nombre d'étage

V. Résultats et recommandations

Recommandations

Il faudra travailler avec les Campus afin de les sensibiliser sur la réduction des consommations Energétiques et en terme de production de CO2.

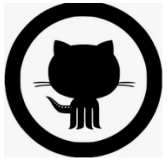
VI. Conclusion

Conclusion

Cette étude m’a permis de :

- Transformer les variables pertinentes d'un modèle d'apprentissage supervisé
- Mettre en place des modèles d'apprentissage supervisé adapté au problème métier
- Comparer les performances d’un modèle d'apprentissage supervisé par rapport à un autre
- Adapter les hyperparamètres d'un algorithme d'apprentissage supervisé afin de l'améliorer

Git



Merci de votre attention