Anticipez les besoins en consommation électrique de bâtiments

PROJET 04/ Openclassrooms

Gulsum Kapanoglu

Dans ce Project..

- ✓Problématique
- ✓Nettoyage des données
- ✓Exploration des données
- ✓Feature Engineering
- ✓Model de Prédiction pour consommation total d'Energy
- ✓Model de Prédiction émission Co2
- ✓L'effet du score ENERGY STAR
- ✓Conclusion

Problématique

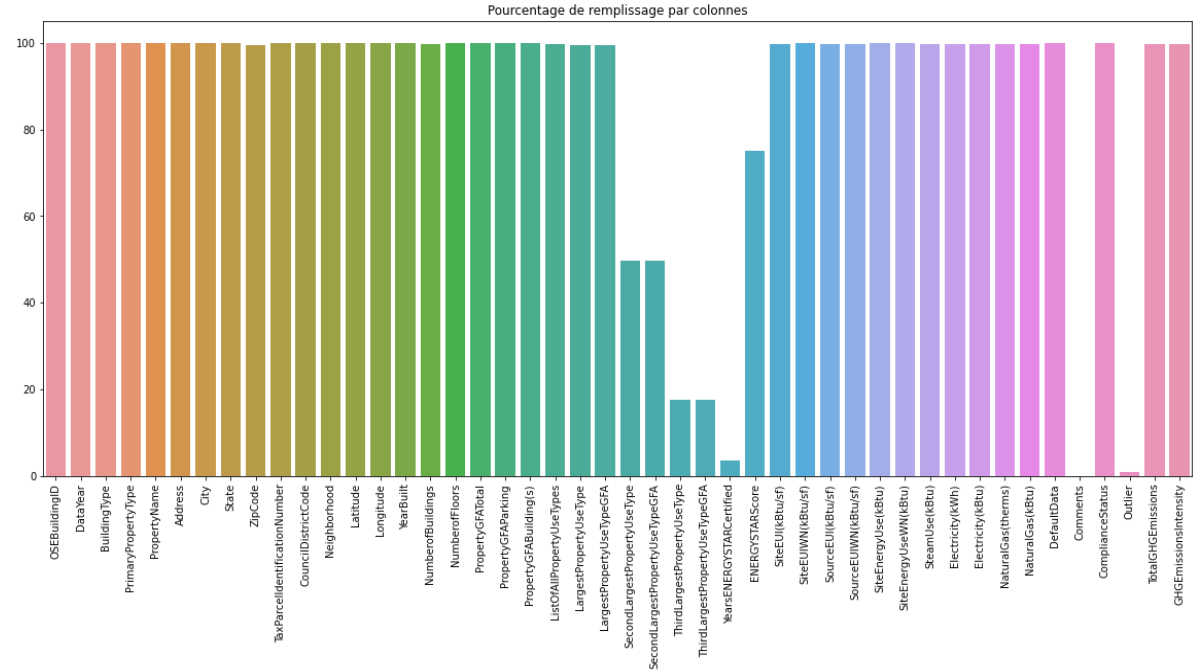
- **Objectif:** Être une ville neutre en émissions de gaz à effets de serre en 2050

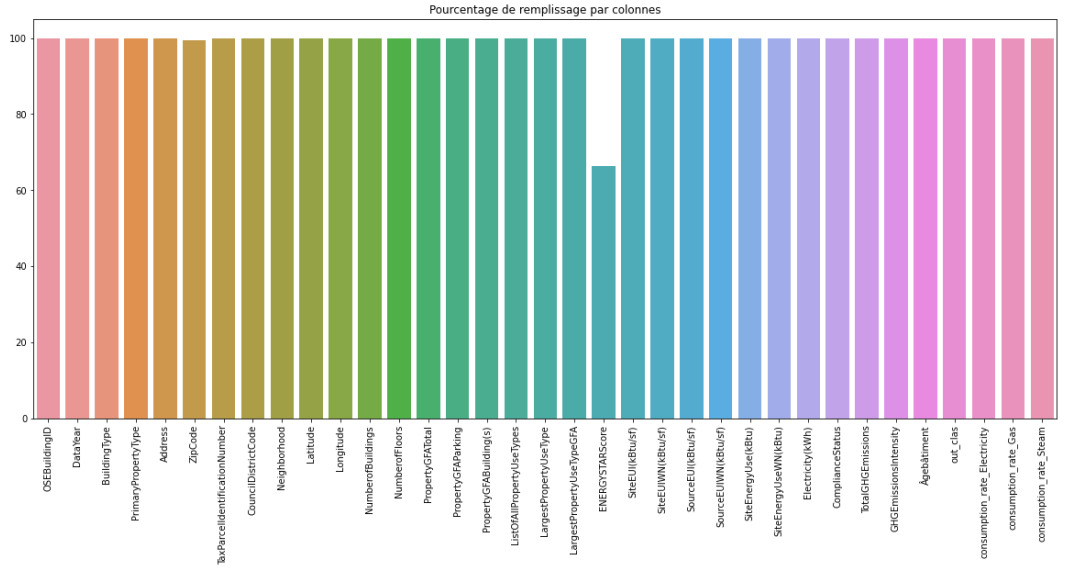
- **Données:** La consommation et aux émissions des **bâtiments non destinés à l'habitation**.

Ces relevés sont coûteux à obtenir, et à partir de ceux déjà réalisés

Identifier l'impact de l'ENERGYSTARSCORE

Nettoyage & Exploration des données





Nettoyage

Suppression des variables non intéressantes

Variables existants sous une autre unité standard

SiteEUI(kBtu/sf)

SourceEUI(kBtu/sf)

SiteEnergyUse(kBtu)

GHGEmissionsIntensity

CouncilDistrictCode

SourceEUIWN(kBtu/sf)

SiteEUIWN(kBtu/sf)

Variables non nécessaires

Address,
City,
State,
Zip code,
ListOfAllPropertyUseTypes,
SecondLargestPropertyUseType,
SecondLargestPropertyUseTypeGFA,
ThirdLargestPropertyUseType,
ThirdLargestPropertyUseTypeGFA,
YearsENERGYSTARCertified,
DefaultData,
Comments,

Feature Engineering

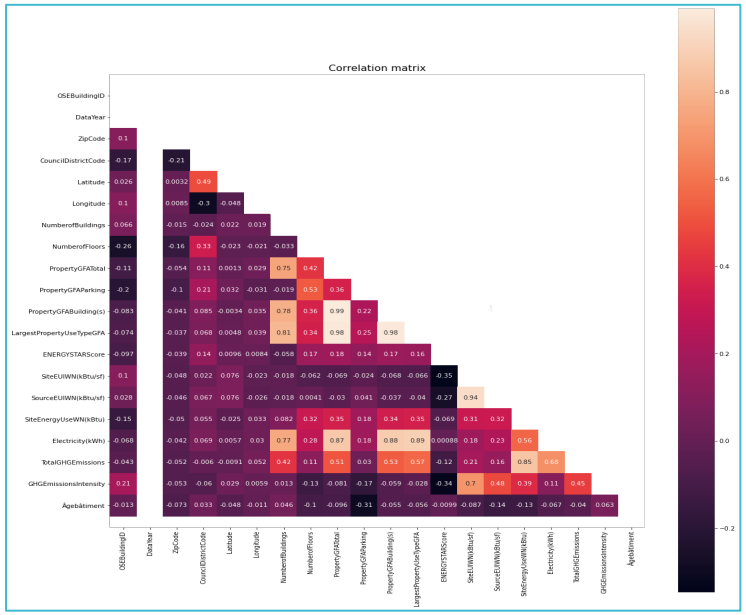
- ✓ Conversion des surfaces (Buildings et Parking) en pourcentage de la surface totale
- ✓ Surface moyenne par bâtiment et par étage
- ✓ Age des bâtiments au lieu de l'année de construction
- ✓ Source principale d'énergie

Feature Engineering

Source principale d'énergie

Electricity(kBtu).		Consumption_rate_Electricity
NaturalGas(kBtu)		Consumption_rate_Gas
SteamUse(kBtu)		Consumption_rate_Steam

Feature Engineering



In [54]:
```
threshold = 0.7
corr_pairs = corr_matrix.unstack().sort_values(kind="quicksort")
strong_corr = (pd.DataFrame(corr_pairs[(abs(corr_pairs) > threshold)])
               .reset_index().rename(columns={0:'corr_coeff'}))
strong_corr = strong_corr[(strong_corr.index%2 == 0) & (strong_corr['level_0'] != strong_corr['level_1'])]
strong_corr.sort_values('corr_coeff', ascending=False)
```

Out[54]:

	level_0	level_1	corr_coeff
22	PropertyGFATotal	PropertyGFABuilding(s)	0.990346
20	PropertyGFABuilding(s)	LargestPropertyUseTypeGFA	0.982501
18	LargestPropertyUseTypeGFA	PropertyGFATotal	0.977860
16	SiteEUIWN(kBtu/sf)	SourceEUIWN(kBtu/sf)	0.942960
14	Electricity(kWh)	LargestPropertyUseTypeGFA	0.887797
12	Electricity(kWh)	PropertyGFABuilding(s)	0.876833
10	PropertyGFATotal	Electricity(kWh)	0.865489
8	TotalGHGEmissions	SiteEnergyUseWN(kBtu)	0.853192
6	LargestPropertyUseTypeGFA	NumberofBuildings	0.808038
4	NumberofBuildings	PropertyGFABuilding(s)	0.783293
2	NumberofBuildings	Electricity(kWh)	0.765437

Feature Engineering

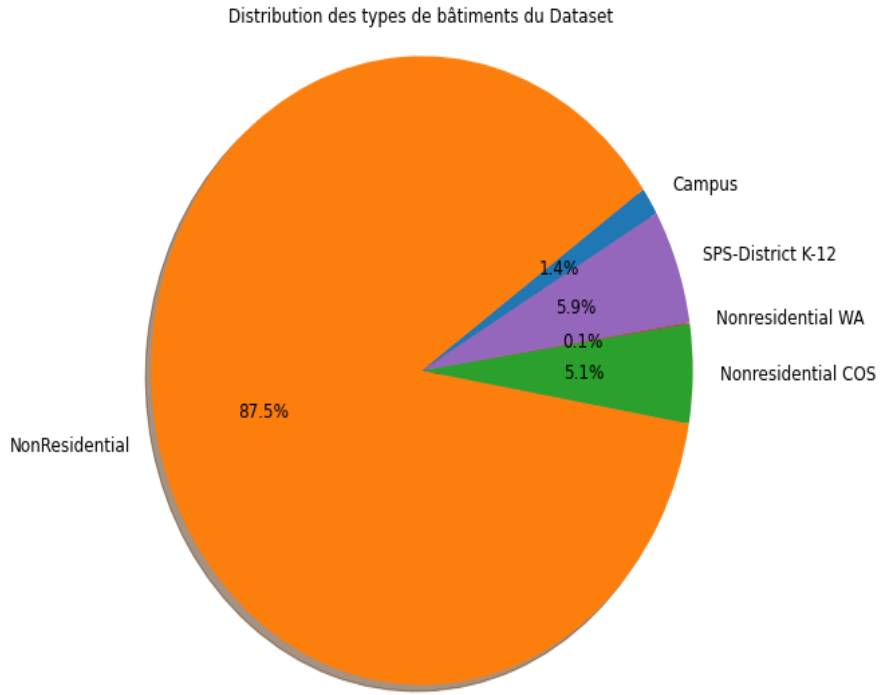
On remarque que les variables suffixées GFA présentent de fortes corrélations avec plusieurs autres variables. Nous allons donc créer de nouvelles variables pour tenter de gommer ces corrélations linéaires :

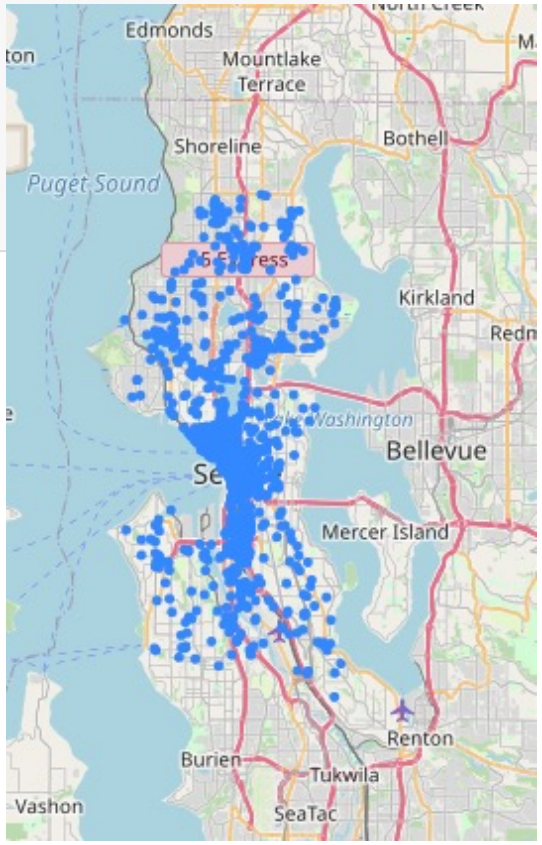
- GFABuildingRate. PropertyGFABuilding(s)/PropertyGFATotal
- GFAParkingRate. PropertyGFAParking/PropertyGFATotal

- GFAPerBuilding PropertyGFATotal
- GFAPerFloor PropertyGFATotal

Exploration des données

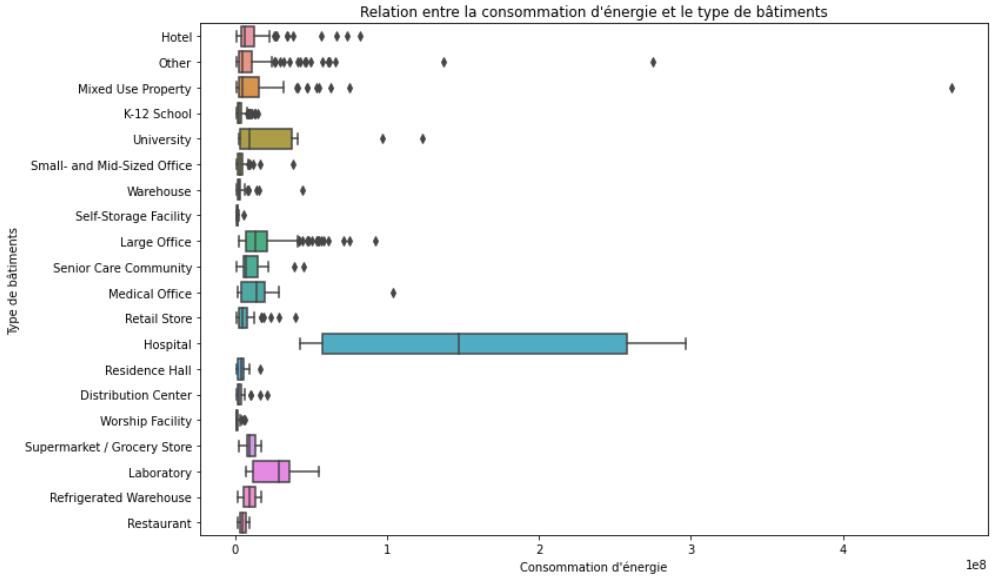
```
In [12]: building_type = energy.groupby(by='BuildingType')['OSEBuildingID'].nunique()

fig, ax = plt.subplots(figsize=(8,8))
ax.pie(building_type.values, labels=building_type.index,
       autopct='%1.1f%%', shadow=True, startangle=30,
       textprops=dict(color="black",size=12))
ax.axis('equal')
ax.set_title("Distribution des types de bâtiments du Dataset")
plt.show()
```

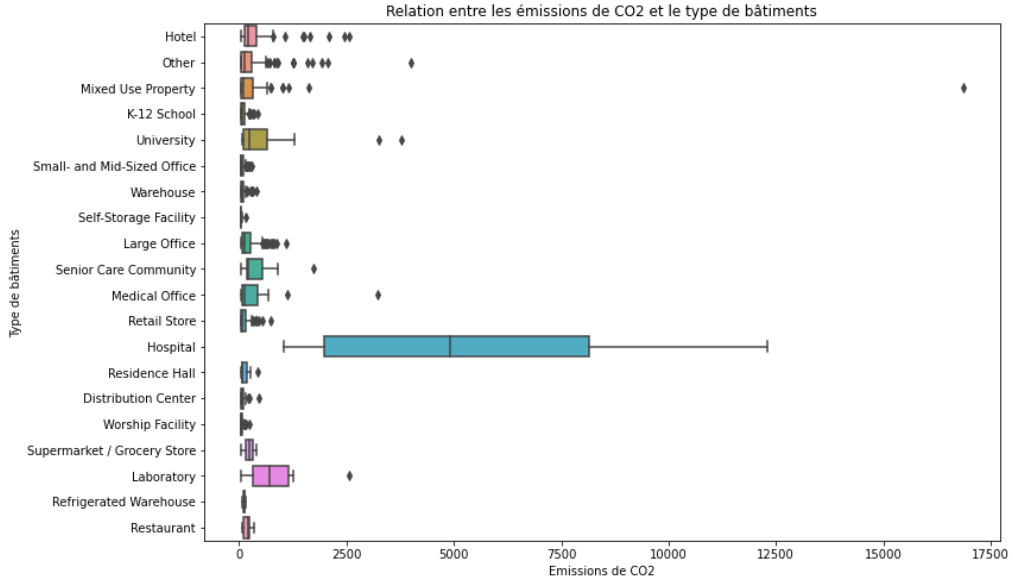




Vérification que tous les bâtiments sont localisés en Seattle

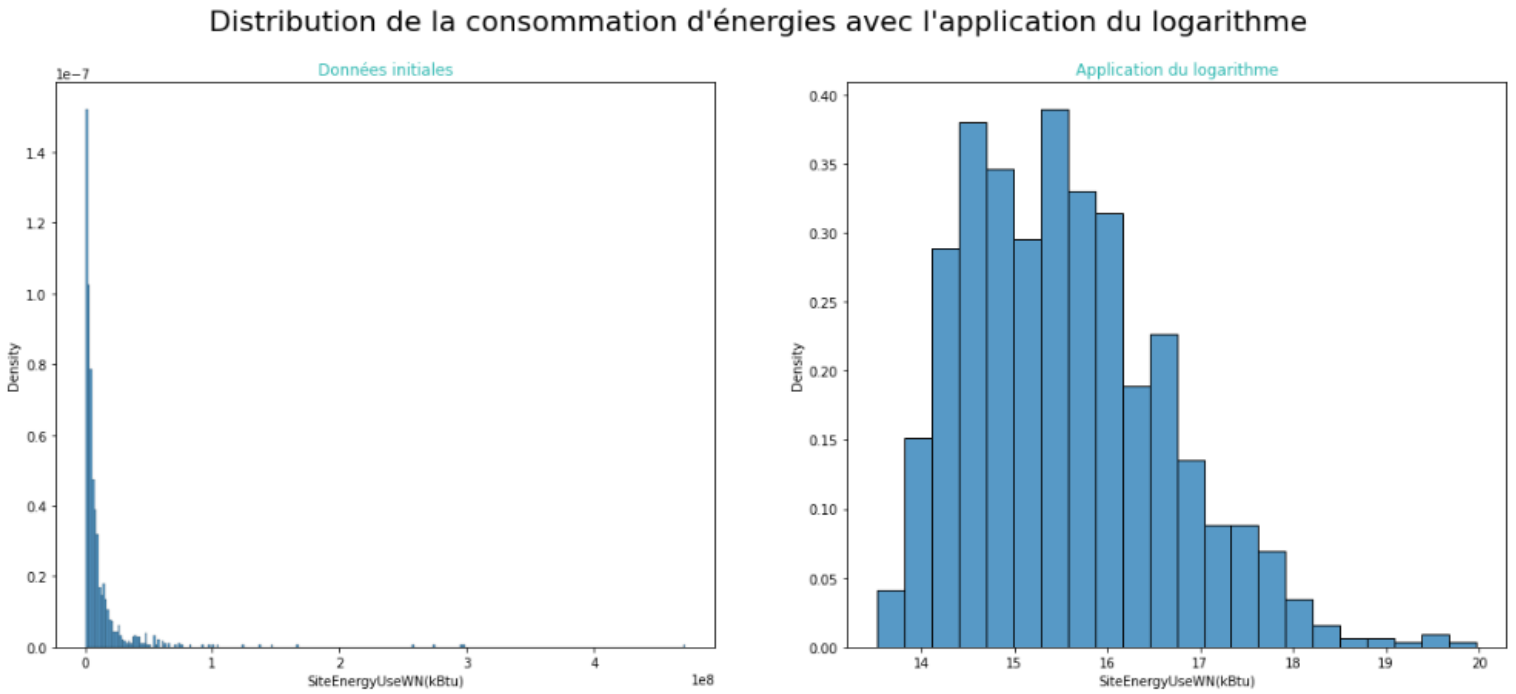
Prédiction Consommation Totale d'énergy





Normalisation & Distribution

Analyse exploratoire



Model de Prédiction

Prédiction de la consommation d'énergie

Comparaison entre Y et log(Y)

Sans log

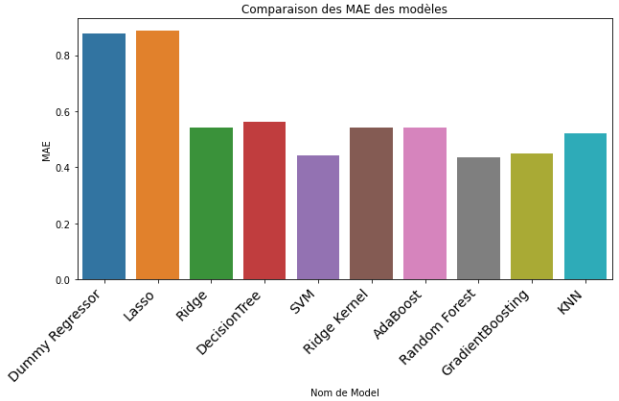
	Dummy Regressor	Lasso	Ridge	DecisionTree	SVM	Ridge Kernel	AdaBoost	Random Forest	GradientBoosting	KNN
MAE	9.882214e+06	6.921841e+06	6.986702e+06	7.039676e+06	9.882192e+06	6.991724e+06	1.684577e+07	6.014202e+06	6.131018e+06	7.031609e+06
MSE	1.099042e+15	8.500710e+14	8.527586e+14	8.720105e+14	1.099041e+15	8.535716e+14	9.216402e+14	7.234253e+14	7.554722e+14	7.880508e+14
RMSE	3.315181e+07	2.915598e+07	2.920203e+07	2.952982e+07	3.315179e+07	2.921595e+07	3.035853e+07	2.689657e+07	2.748585e+07	2.807224e+07
R²	-4.900000e-02	1.880000e-01	1.860000e-01	1.670000e-01	-4.900000e-02	1.850000e-01	1.200000e-01	3.090000e-01	2.790000e-01	2.480000e-01
time	1.338940e-04	2.340823e-03	2.113030e-04	4.138690e-04	5.639792e-02	3.836831e-03	7.455705e-03	1.377393e-02	7.919580e-04	7.438980e-03

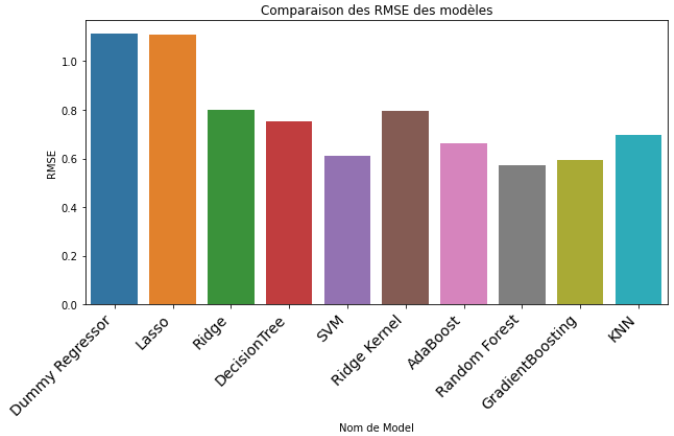
Passage au log

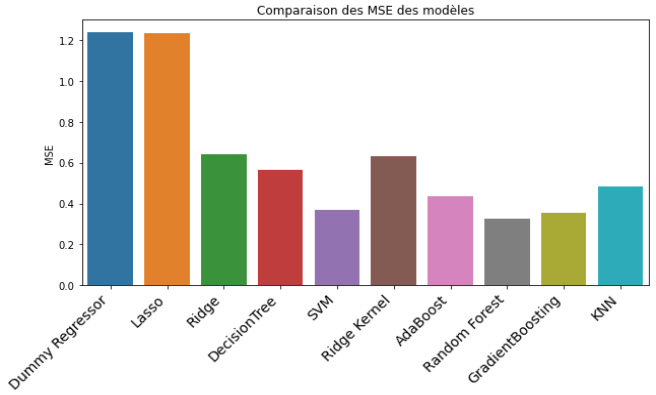
	Dummy Regressor	Lasso	Ridge	DecisionTree	SVM	Ridge Kernel	AdaBoost	Random Forest	GradientBoosting	KNN
MAE	0.878000	0.886000	0.541000	0.562000	0.443000	0.540000	0.542000	0.435000	0.448000	0.52100
MSE	1.237000	1.234000	0.643000	0.567000	0.371000	0.630000	0.438000	0.326000	0.353000	0.48500
RMSE	1.112000	1.111000	0.802000	0.753000	0.609000	0.794000	0.661000	0.571000	0.595000	0.69700
R²	-0.003000	-0.000000	0.479000	0.540000	0.699000	0.489000	0.645000	0.735000	0.713000	0.60600
time	0.000069	0.000362	0.000585	0.000449	0.052413	0.002466	0.006904	0.013172	0.000695	0.00463

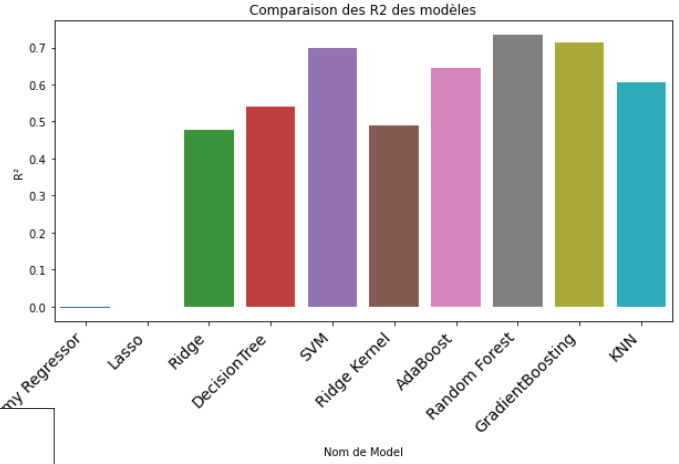
Résultats sont meilleurs avec l'application du logarithme

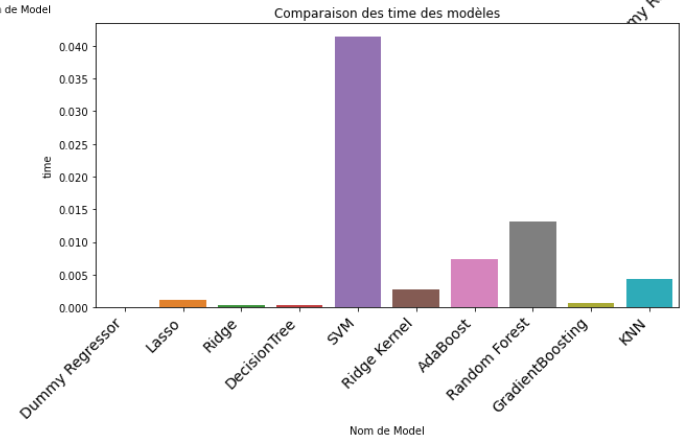
Model de Prédiction







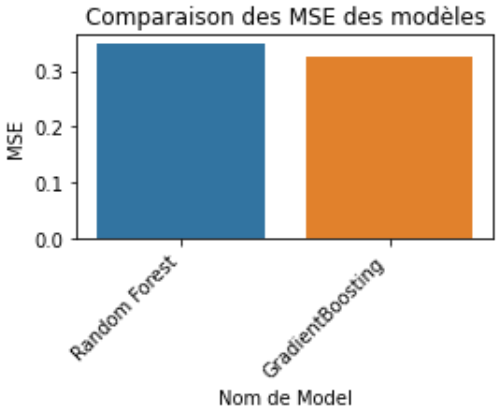


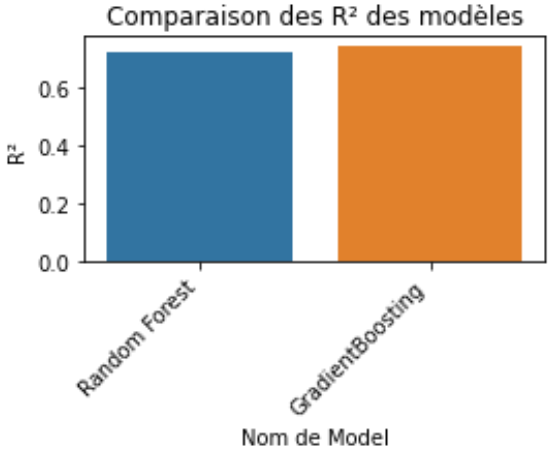


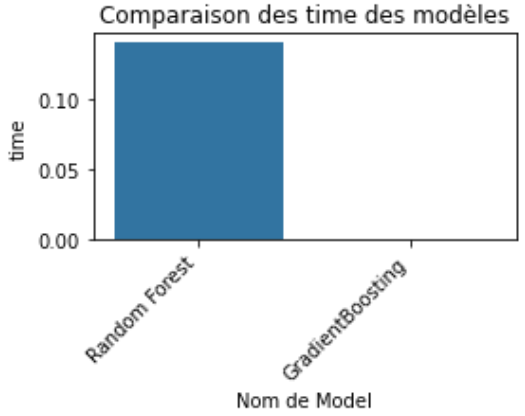
Random Forest et GradientBoosting

Pour aller plus loin

	Random Forest	GradientBoosting
MAE	0.446000	0.427000
MSE	0.347000	0.325000
RMSE	0.589000	0.570000
R²	0.719000	0.736000
time	0.140761	0.000667



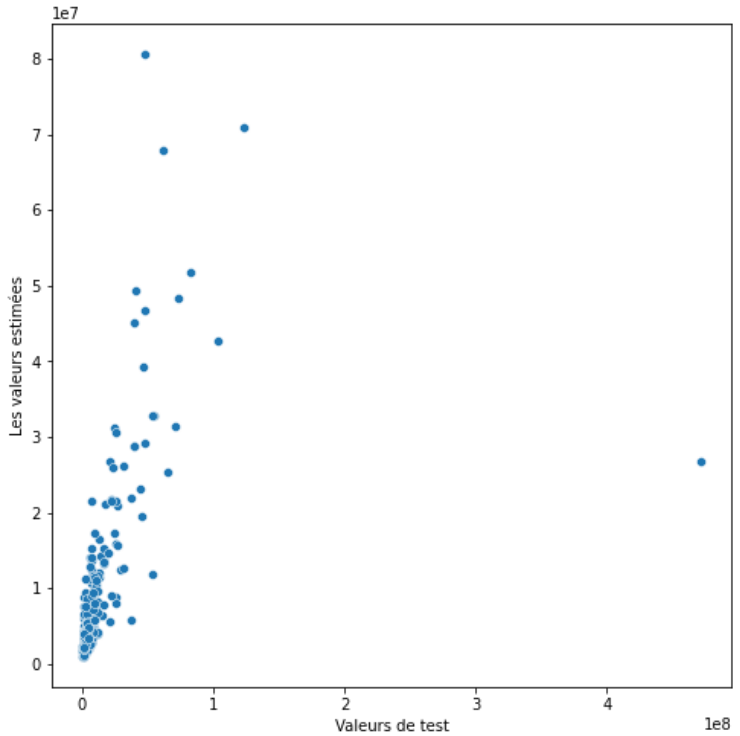




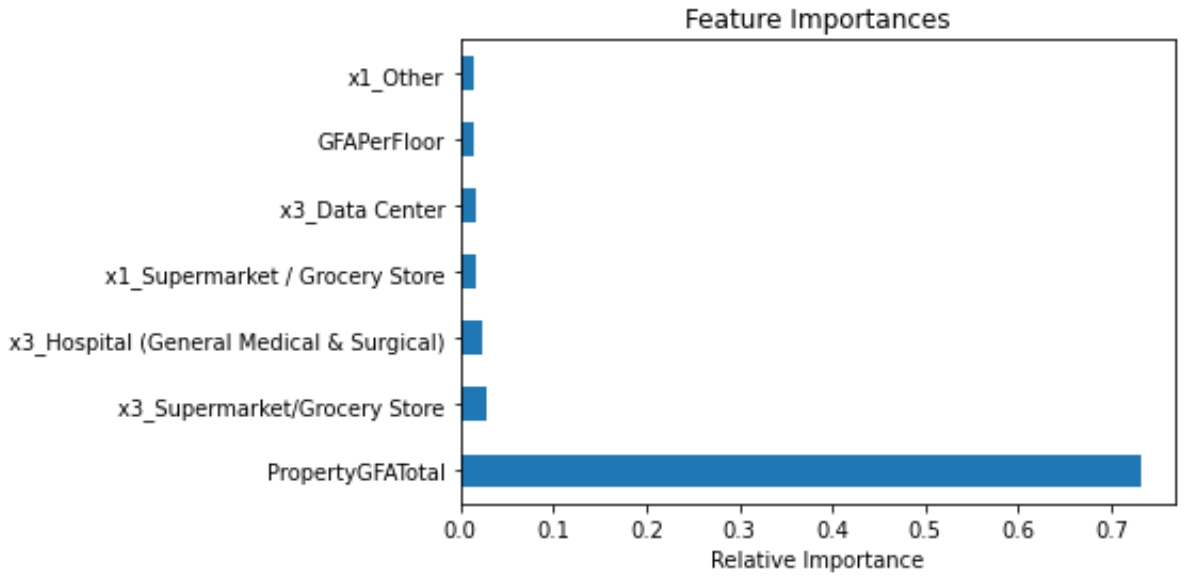
Sélection de model est Gradient Boosting Regressor

	Random Forest	GradientBoosting
MAE	0.446000	0.427000
MSE	0.347000	0.325000
RMSE	0.589000	0.570000
R²	0.719000	0.736000
time	0.140761	0.000667

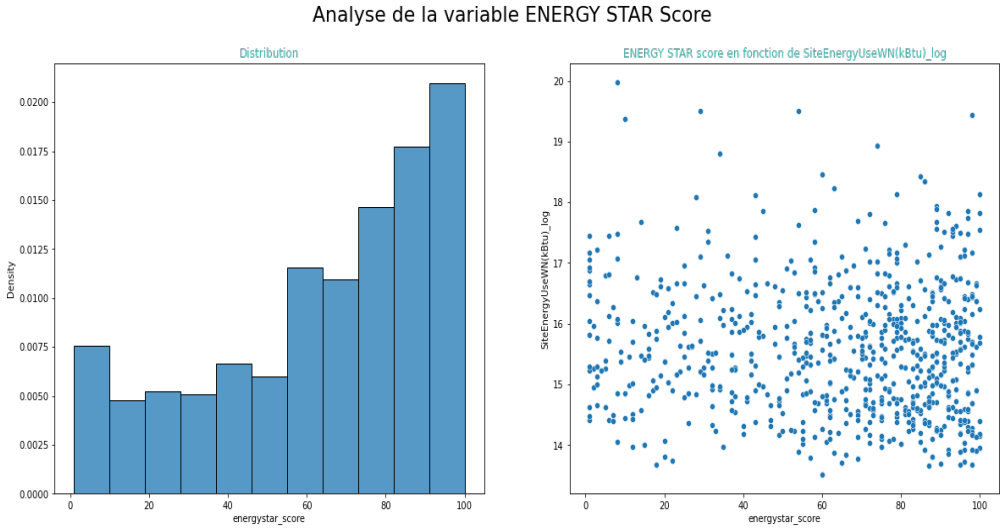
Comparaison de la consommation de énergie prédite avec algorithme Gradient Boosting versus la consommation reel



Feature importance pour energy



L'effet du score ENERGY STAR



Le score ENERGY STAR ne semble pas avoir de corrélation importante avec consommation energy

Prédiction des consommation Energy:

Entrainement sur le nouveau jeu de données intégrant la variable ENERGYSTARSCORE.
Evaluation des performances et comparaison avec les performances initialement obtenues.

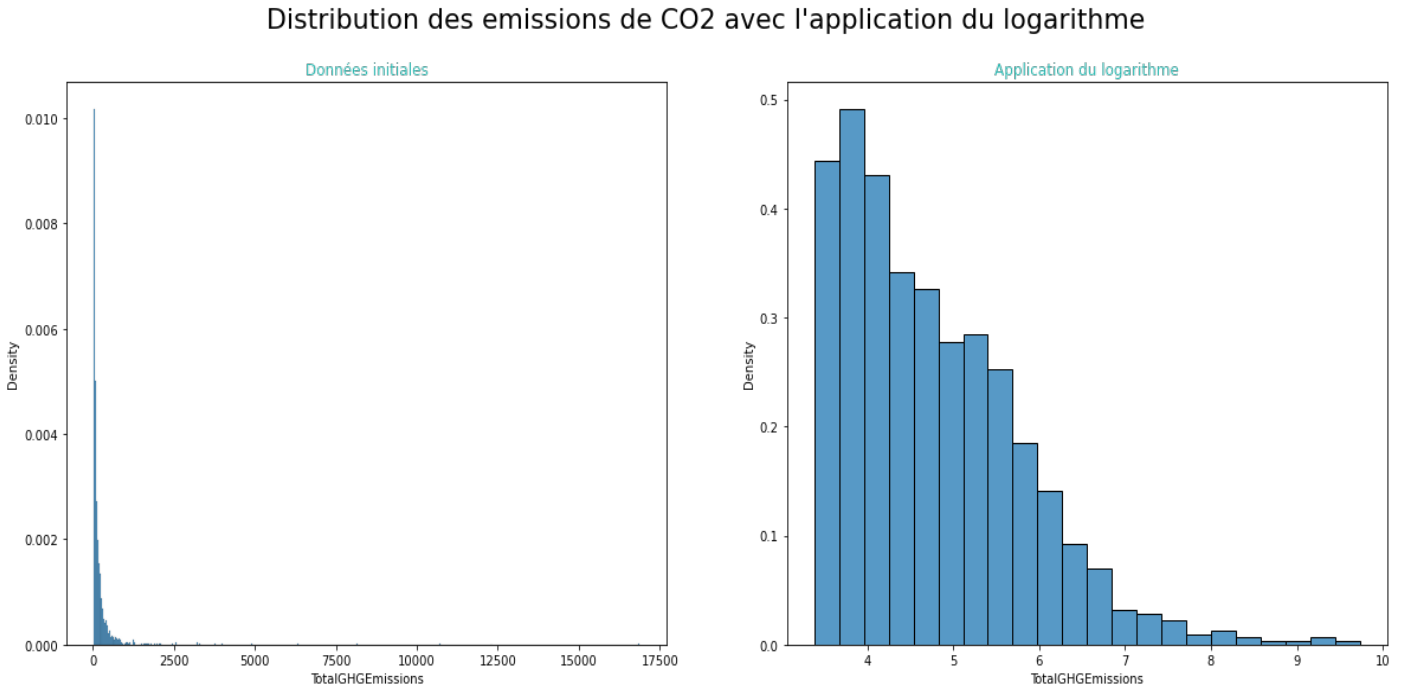
Sans ENERGYSTAR

MAE: 0.43111
MSE: 0.33157
RMSE: 0.5758197600562731
MAPE: 0.02777
R²: 0.73113

Avec ENERGYSTAR

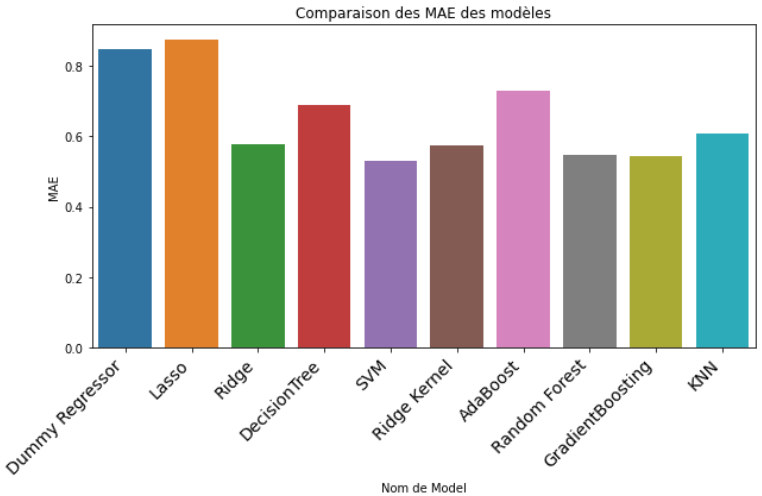
MAE: 0.29396
MSE: 0.18344
RMSE: 0.4282990519386477
MAPE: 0.01871
R²: 0.8283

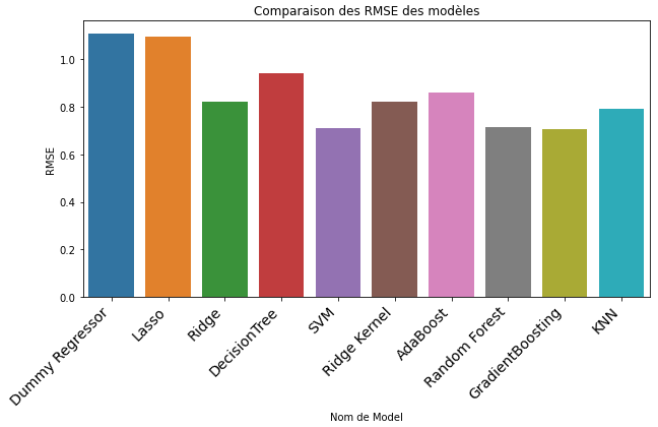
Prediction émissions de CO_2

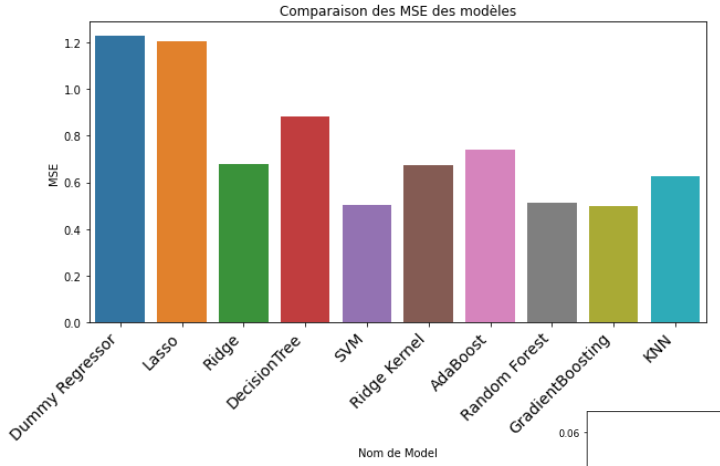


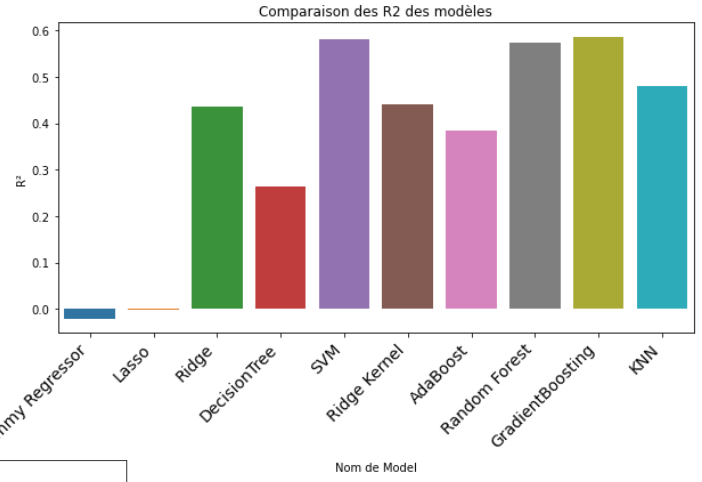
Prédiction des émissions de CO_2

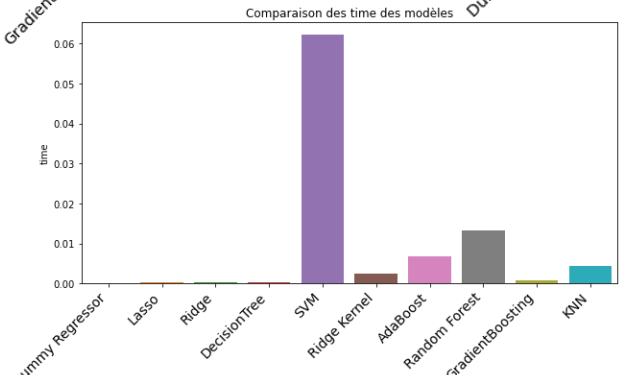
Model de Prédiction





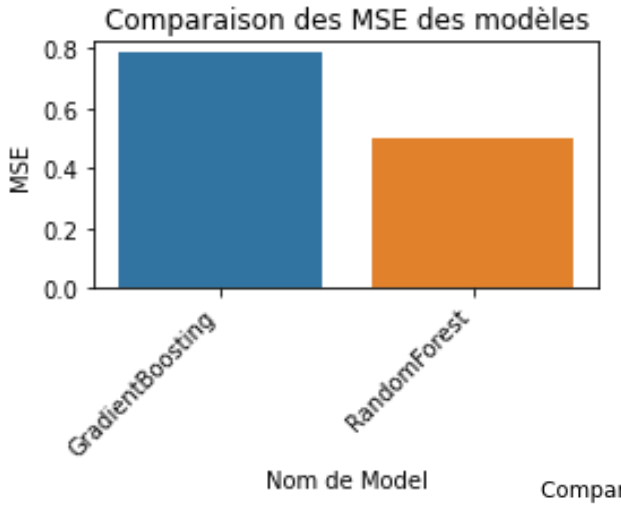


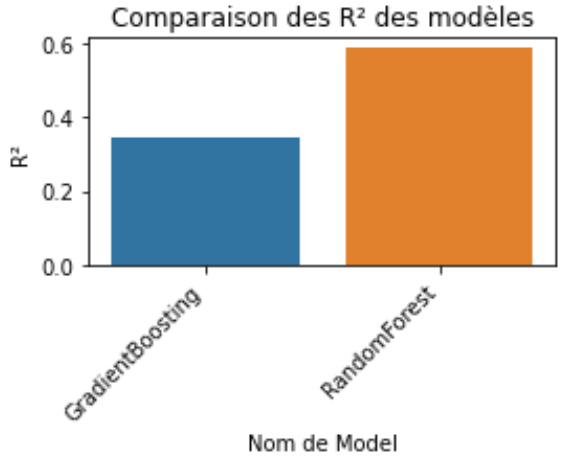


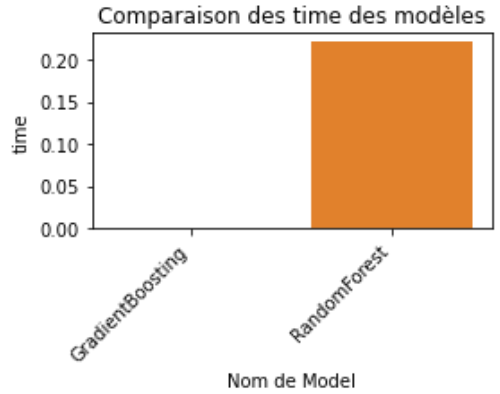


Pour aller plus loin

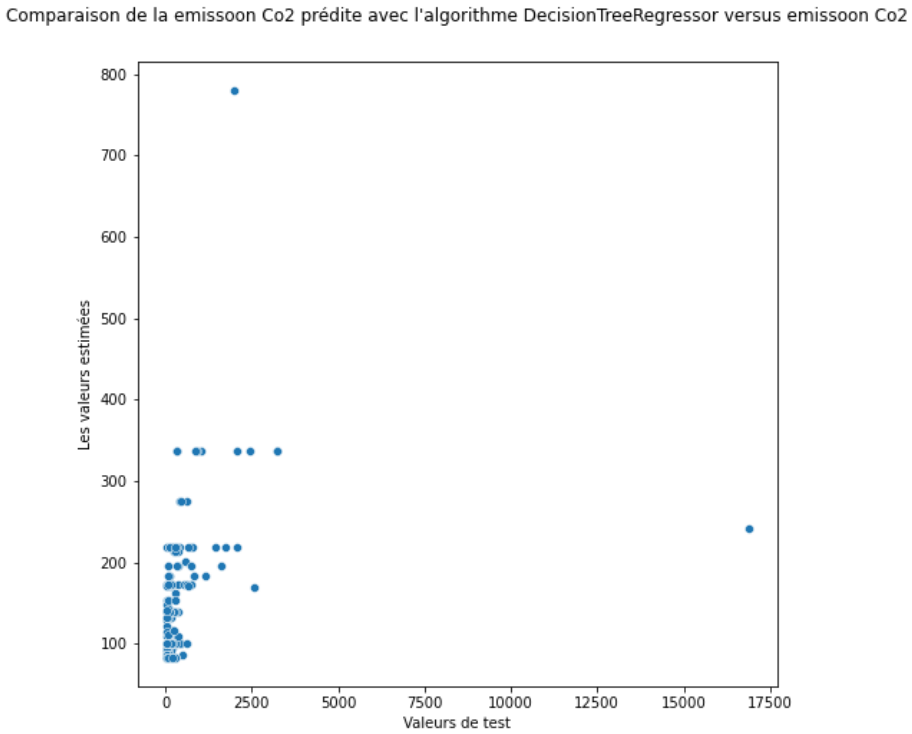
	GradientBoosting	RandomForest
MAE	0.718000	0.547000
MSE	0.784000	0.500000
RMSE	0.886000	0.707000
R²	0.347000	0.584000
time	0.000566	0.220409





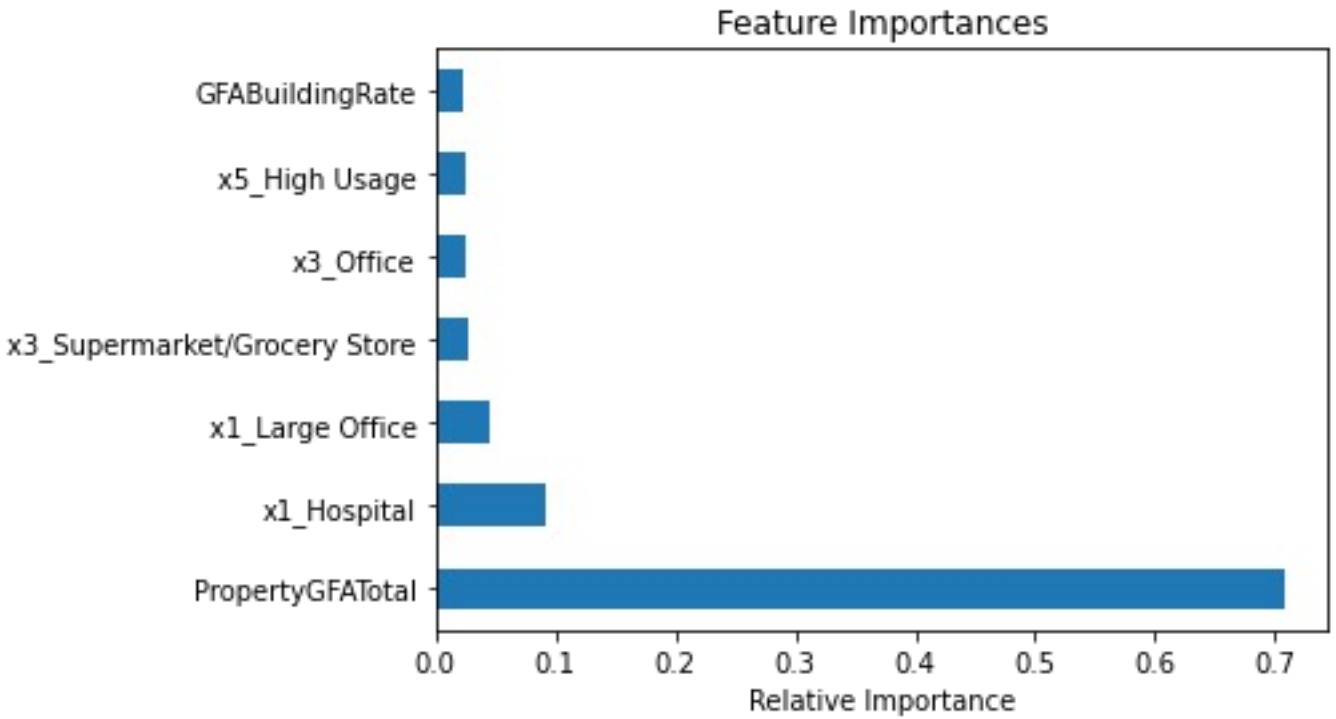


Sélection de model est Gradient Boosting Regressor

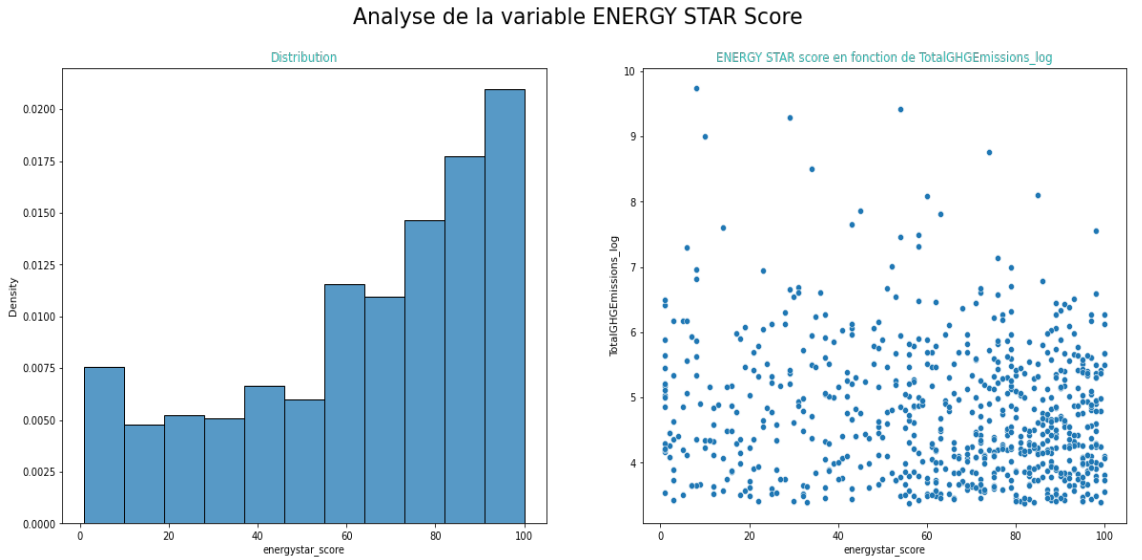



```
MAE:  0.71785
MSE:  0.78441
RMSE:  0.885669689015457
MAPE:  0.15345
R²:  0.34711
```

Feature importance pour Co2



L'effet du score ENERGY STAR



Le score ENERGY STAR ne semble pas avoir de corrélation importante avec émission Co2

Sans ENERGYSTAR

MAE: 0.71785
MSE: 0.78441
RMSE: 0.885669689015457
MAPE: 0.15345
R^2: 0.34711

Avec ENERGYSTAR

MAE: 0.65031
MSE: 0.64078
RMSE: 0.80049000014991371
MAPE: 0.14055
R^2: 0.38434

Conclusion

- Possibilité de prédire la consommation d'énergie de manière fiable (**73%** de variance expliquée)
- Possibilité de prédire les émissions de CO2 mais légèrement moins fiable (**34%** de variance expliquée)
- Score Energy Star : inutile à la prédiction

- Proposition d'amélioration: Rénover ces bâtiments sur le modèle des bâtiments à énergie positive

Merci!