# Projet 4 : Anticipez les besoins en consommation électrique de bâtiments.

Ilham NOUMIR | Parcours Data Science | Date : 15/10/2021

# Contexte du projet :

Relevés de consommation ont été effectués pour les années 2015 et en 2016.

Cependant, ces relevés sont coûteux à obtenir, et difficile à obtenir

Mission :

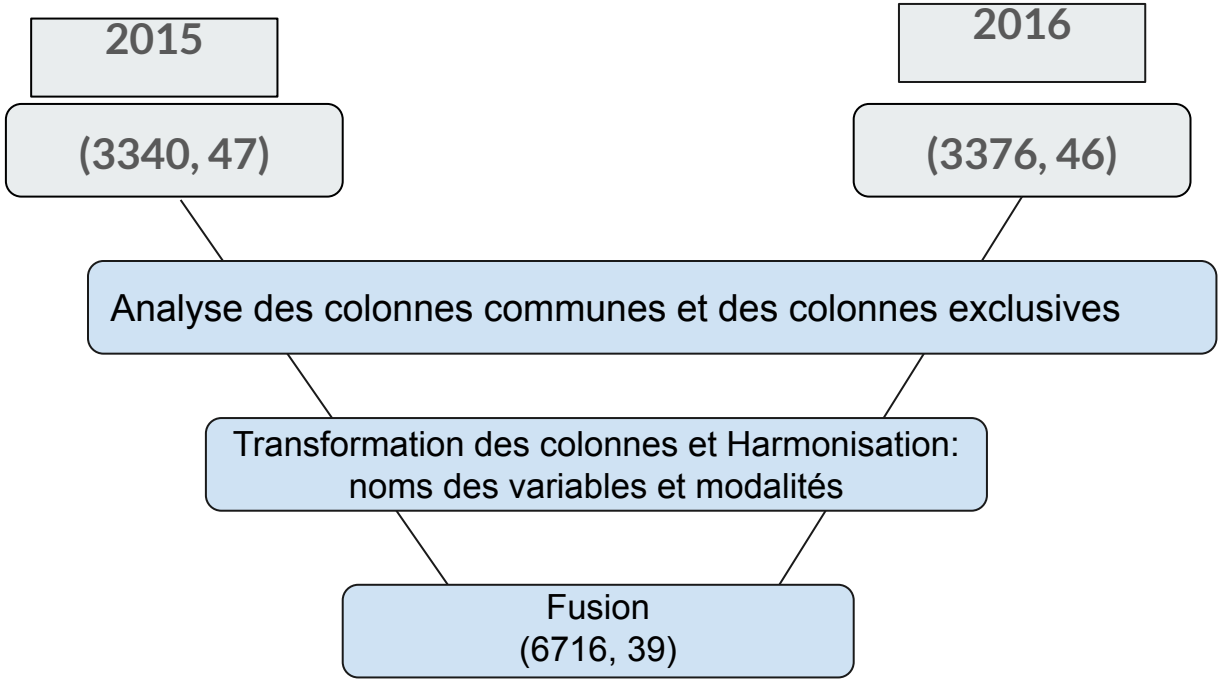
- Prédire les émissions de CO2 et la consommation totale d'énergie
- Evaluer l'intérêt de l'"ENERGY STAR Score" pour la prédiction d'émissions

Tâches :

- Réaliser une courte analyse exploratoire.
- Tester différents modèles de prédiction afin de répondre au mieux à la problématique.

# Préparation et analyse exploratoire des données :

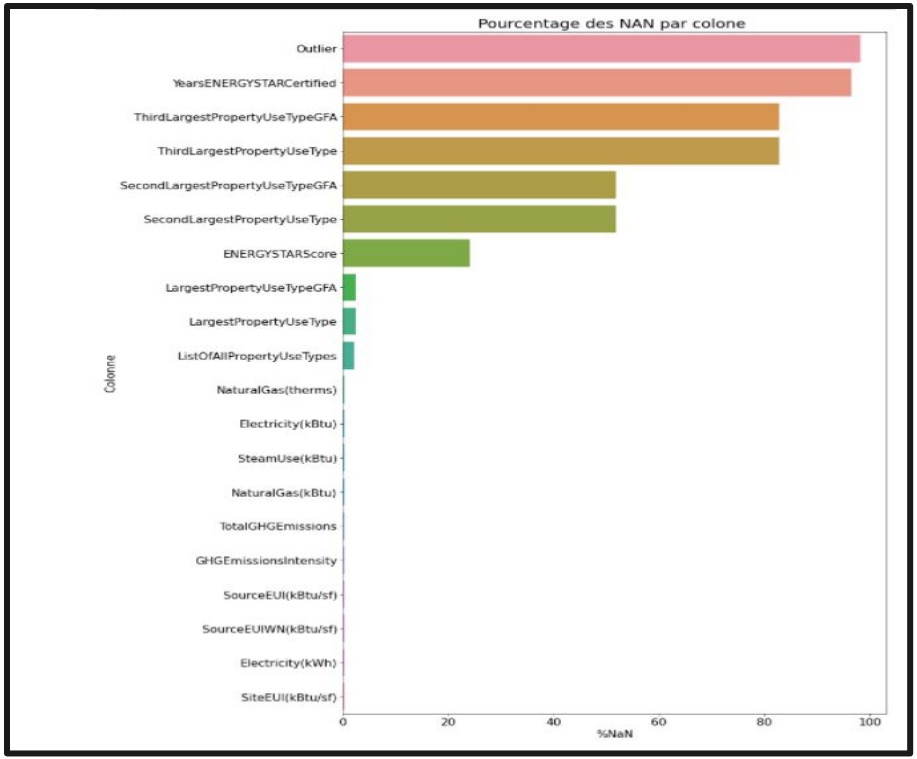
Le jeu de données comporte deux fichiers csv :



# Nettoyage des données :

Traitement des valeurs manquantes : par colonne

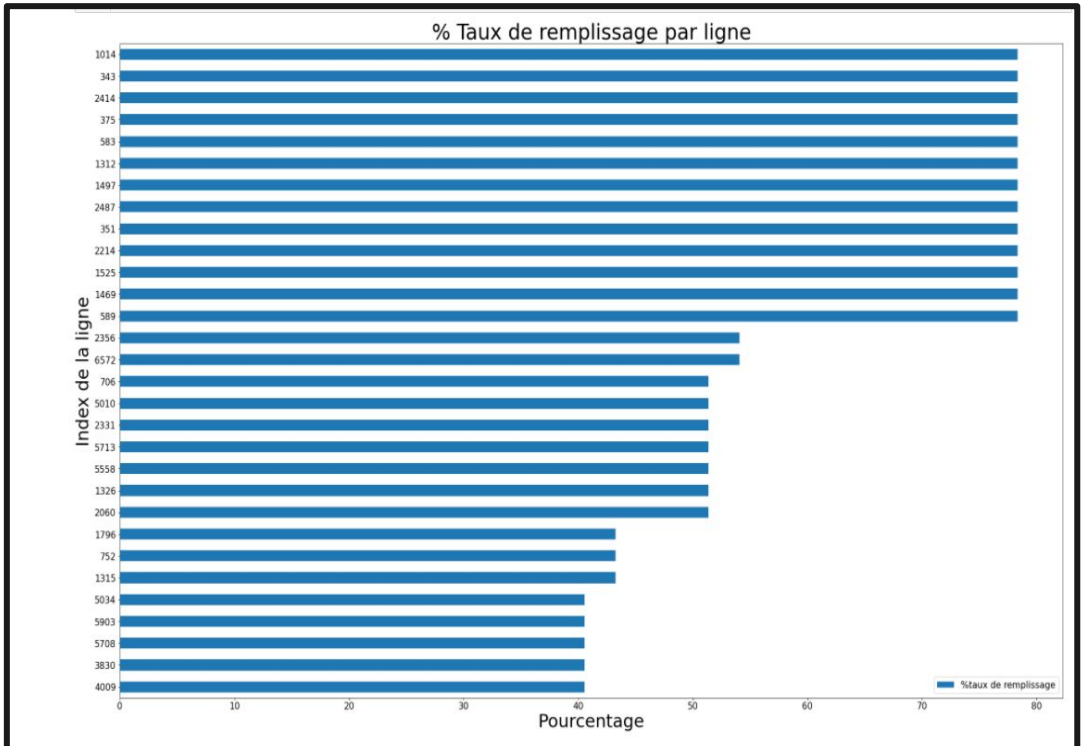
| | Colonne | %NaN |
|---|---|---|
| 0 | Outlier | 98.272781 |
| 1 | YearsENERGYSTARCertified | 96.590232 |
| 2 | ThirdLargestPropertyUseTypeGFA | 82.787373 |
| 3 | ThirdLargestPropertyUseType | 82.787373 |
| 4 | SecondLargestPropertyUseTypeGFA | 51.786778 |
| 5 | SecondLargestPropertyUseType | 51.786778 |
| 6 | ENERGYSTARScore | 24.166170 |
| 7 | LargestPropertyUseTypeGFA | 2.322811 |
| 8 | LargestPropertyUseType | 2.322811 |
| 9 | ListOfAllPropertyUseTypes | 2.025015 |
| 10 | NaturalGas(therms) | 0.282906 |
| 11 | Electricity(kBtu) | 0.282906 |
| 12 | SteamUse(kBtu) | 0.282906 |
| 13 | NaturalGas(kBtu) | 0.282906 |
| 14 | TotalGHGEmissions | 0.282906 |
| 15 | GHGEmissionsIntensity | 0.282906 |
| 16 | SourceEUI(kBtu/sf) | 0.282906 |
| 17 | SourceEUIWN(kBtu/sf) | 0.282906 |
| 18 | Electricity(kWh) | 0.282906 |
| 19 | SiteEUI(kBtu/sf) | 0.253127 |
| 20 | SiteEUIWN(kBtu/sf) | 0.238237 |
| 21 | SiteEnergyUseWN(kBtu) | 0.238237 |
| 22 | SiteEnergyUse(kBtu) | 0.223347 |
| 23 | NumberofFloors | 0.119119 |
| 24 | NumberofBuildings | 0.119119 |
| 25 | TaxParcelIdentificationNumber | 0.029780 |
| 26 | DefaultData | 0.014890 |
| 27 | ComplianceStatus | 0.000000 |
| 28 | PropertyName | 0.000000 |



# Nettoyage des données :

Traitement des valeurs manquantes : par ligne

| | %taux de remplissage par ligne |
|---|---|
| 4009 | 40.540541 |
| 3830 | 40.540541 |
| 5708 | 40.540541 |
| 5903 | 40.540541 |
| 5034 | 40.540541 |
| 1315 | 43.243243 |
| 752 | 43.243243 |
| 1796 | 43.243243 |
| 2060 | 51.351351 |
| 1326 | 51.351351 |
| 5558 | 51.351351 |
| 5713 | 51.351351 |
| 2331 | 51.351351 |
| 5010 | 51.351351 |
| 706 | 51.351351 |
| 6572 | 54.054054 |
| 2356 | 54.054054 |
| 589 | 78.378378 |
| 1469 | 78.378378 |
| 1525 | 78.378378 |
| 2214 | 78.378378 |
| 351 | 78.378378 |
| 2487 | 78.378378 |
| 1497 | 78.378378 |
| 1312 | 78.378378 |
| 583 | 78.378378 |
| 375 | 78.378378 |
| 2414 | 78.378378 |
| 343 | 78.378378 |
| 1014 | 78.378378 |



# Nettoyage des données :

Traitement des valeurs négatifs :

| | YearBuilt | NumberofBuildings | NumberofFloors | PropertyGFATotal | PropertyGFAParking | PropertyGFABuilding(s) | LargestPropertyUseTypeGFA |
|---|---|---|---|---|---|---|---|
| count | 6560.000000 | 6560.000000 | 6552.000000 | 6.560000e+03 | 6560.000000 | 6.560000e+03 | 6.441000e+03 |
| mean | 1968.408841 | 1.072256 | 4.710928 | 9.340017e+04 | 9591.081098 | 8.380909e+04 | 7.744464e+04 |
| std | 32.889595 | 1.610576 | 5.516997 | 1.893317e+05 | 34094.228659 | 1.751053e+05 | 1.694550e+05 |
| min | 1900.000000 | 0.000000 | 0.000000 | 1.128500e+04 | -3.000000 | -5.055000e+04 | 5.656000e+03 |
| 25% | 1948.000000 | 1.000000 | 2.000000 | 2.849200e+04 | 0.000000 | 2.728800e+04 | 2.509400e+04 |
| 50% | 1974.000000 | 1.000000 | 4.000000 | 4.419500e+04 | 0.000000 | 4.229200e+04 | 3.960000e+04 |
| 75% | 1997.000000 | 1.000000 | 5.000000 | 9.000000e+04 | 0.000000 | 8.163900e+04 | 7.484000e+04 |
| max | 2015.000000 | 111.000000 | 99.000000 | 9.320156e+06 | 512608.000000 | 9.320156e+06 | 9.320156e+06 |

# Nettoyage des données :

Filtrages des données : Filtrage des catégories des bâtiments

```
--------------------------------------------------------------------------------
Le nombre de valeurs que peuvent prendre la colone : BuildingType
8
Les valeurs que peuvent prendre la colone : BuildingType


['NonResidential' 'Nonresidential COS' 'Multifamily MR (5-9)'
 'SPS-District K-12' 'Multifamily LR (1-4)' 'Campus'
 'Multifamily HR (10+)' 'Nonresidential WA']
--------------------------------------------------------------------------------
Le nombre de valeurs que peuvent prendre la colone : PrimaryPropertyType
30
Les valeurs que peuvent prendre la colone : PrimaryPropertyType


['Hotel' 'Other' 'Mid-Rise Multifamily' 'Mixed Use Property' 'K-12 School'
 'College/University' 'Small- and Mid-Sized Office'
 'Self-Storage Facility' 'Distribution Center' 'Large Office'
 'Retail Store' 'Low-Rise Multifamily' 'Senior Care Community'
 'Medical Office' 'Hospital' 'Residence Hall/Dormitory'
```

# Nettoyage des données :

Filtrages des données : Filtrage des catégories des bâtiments

```
usetype_dict = {'Retail Store' : 'Retail',
                'Supermarket/Grocery Store' : 'Retail',
                'Repair Services (Vehicle, Shoe, Locksmith, etc)' : 'Retail',
                'Automobile Dealership': 'Retail',
                'Convenience Store without Gas Station': 'Retail',
                'Personal Services': 'Retail',
                'Enclosed Mall': 'Retail',
                'Strip Mall': 'Retail',
                'Wholesale Club/Supercenter': 'Retail',
                'Other - Mall': 'Retail',
                'Supermarket / Grocery Stor': 'Retail',

                'Food Sales': 'Leisure',
                'Restaurant' : 'Leisure',
                'Other - Restaurant/Bar' : 'Leisure',
                'Food Service': 'Leisure' ,
                'Worship Facility': 'Leisure',
                'Other - Recreation': 'Leisure',
                'Other - Entertainment/Public Assembly': 'Leisure',
                'Performing Arts': 'Leisure',
                'Bar/Nightclub': 'Leisure',
                'Movie Theater': 'Leisure',
                'Museum': 'Leisure',
                'Social/Meeting Hall': 'Leisure',
                'Fitness Center/Health Club/Gym': 'Leisure',
                'Lifestyle Center ': 'Leisure',
                'Fast Food Restaurant': 'Leisure',

                'Multifamily Housing' : 'Hotel/Senior Care/Housing',
                'Other - Lodging/Residential': 'Hotel/Senior Care/Housing',
                'Residence Hall/Dormitory': 'Hotel/Senior Care/Housing',
                'Hotel' : 'Hotel/Senior Care/Housing',
                'Senior Care Community' : 'Hotel/Senior Care/Housing',
                'Residential Care Facility' : 'Hotel/Senior Care/Housing',
                'High-Rise Multifamily': 'Hotel/Senior Care/Housing',

                'Medical Office' : 'Health',

                'Other - Services' : 'Office',
                'Bank Branch' : 'Office',
                'Financial Office' : 'Office',
                'Other - Public Services': 'Office',

                'K-12 School' : 'Education',
                'Other - Education': 'Education',
                'Vocational School' : 'Education',
                'Adult Education' : 'Education',
                'Pre-school/Daycare': 'Education',
                'University': 'Education',
                'College/University': 'Education',
                'Library': 'Education'
               }
```

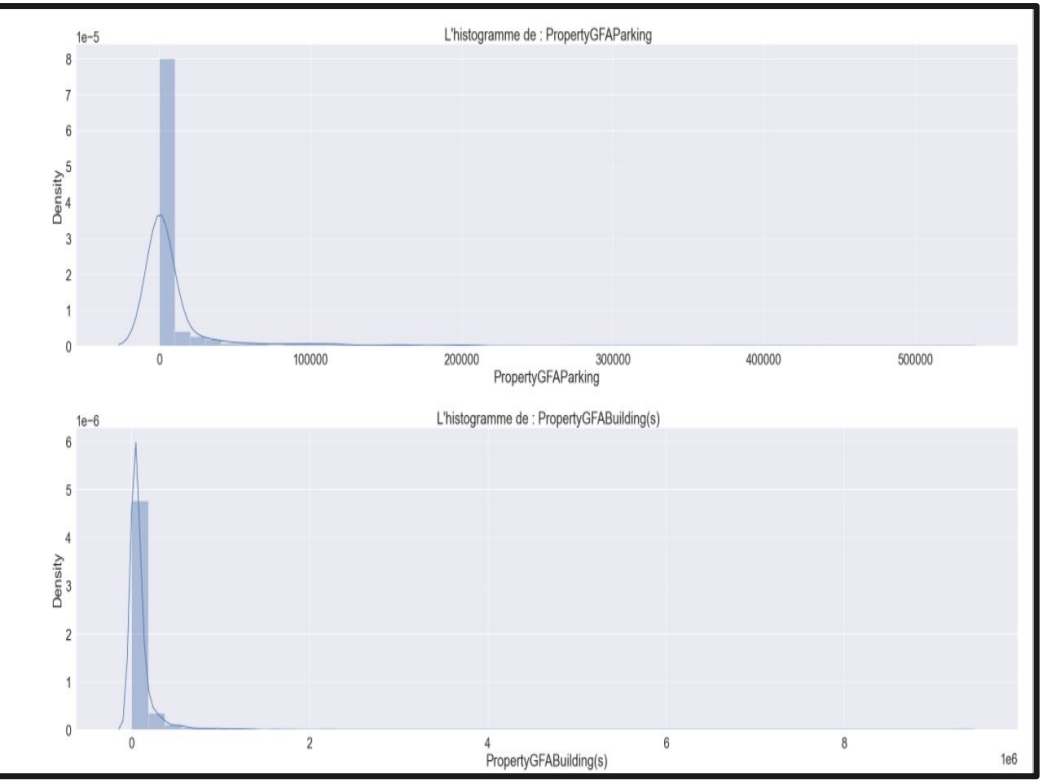
```
BuildingType 8
LargestPropertyUseType 57
PrimaryPropertyType 32
SecondLargestPropertyUseType 50
ThirdLargestPropertyUseType 45
```

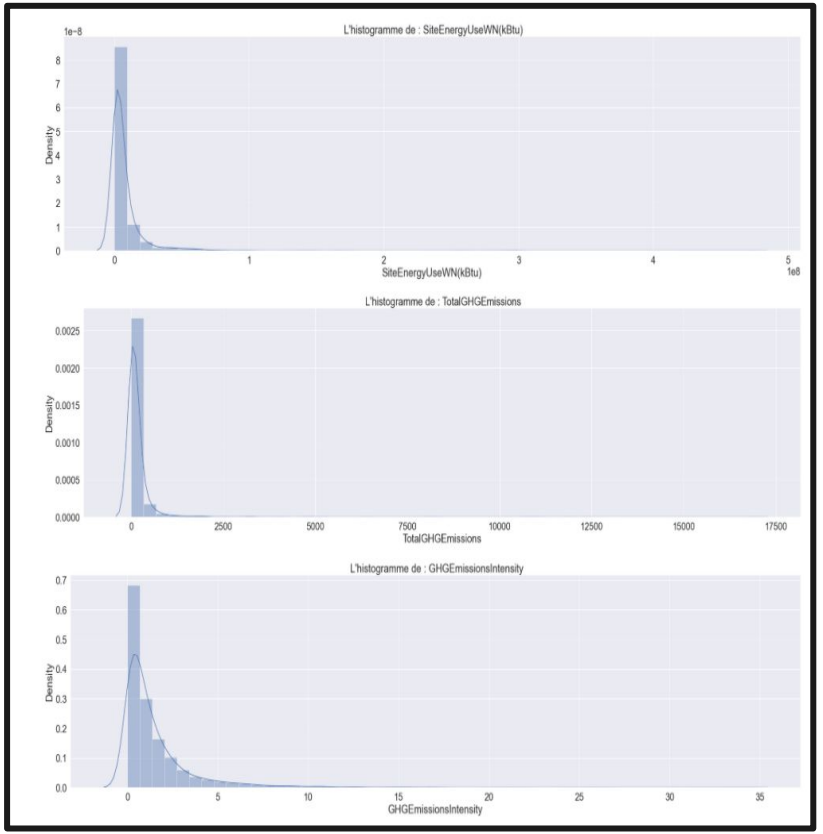
```
BuildingType 5
LargestPropertyUseType 27
PrimaryPropertyType 18
SecondLargestPropertyUseType 19
ThirdLargestPropertyUseType 20
```

# Nettoyage des données :

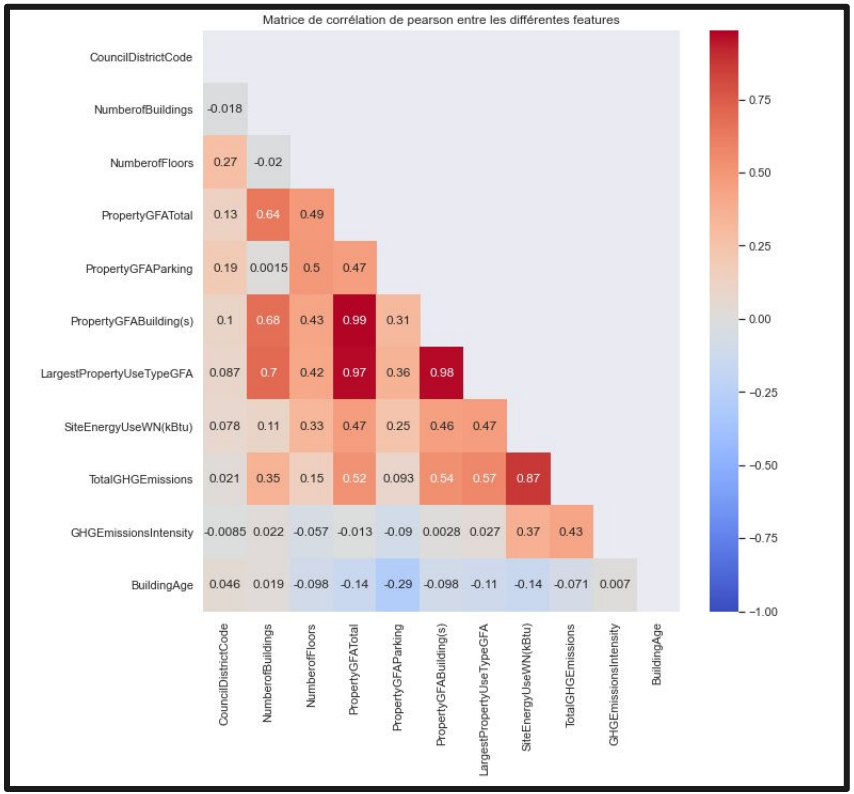
1. Fusion des deux jeu de données : data-> **(6716, 39)**
2. Traitement des valeurs manquantes par ligne et par colonnes -> **(6458, 37)**
3. Filtrage des catégories bâtiments : (Bâtiments non résidentiels et réduction de la variabilité des catégories -> **(3241, 37)**
4. Suppression des colonnes avec des données énergétiques: data-> **(3162, 27)**
5. Création d'une nouvelle variable BuildingAge
6. Suppression des colonnes inutiles à l'analyse : Comment ,OSEBuildingID , DataYear , YearBuilt …. : data -> **(3162, 21)**

# Exploration des données :



# Exploration des données : Matrice de corrélation




Corrélations fortes :

- PropertyGFATotal
- PropertyGFABuilding(s)
- LargestPropertyUseTypeGFA

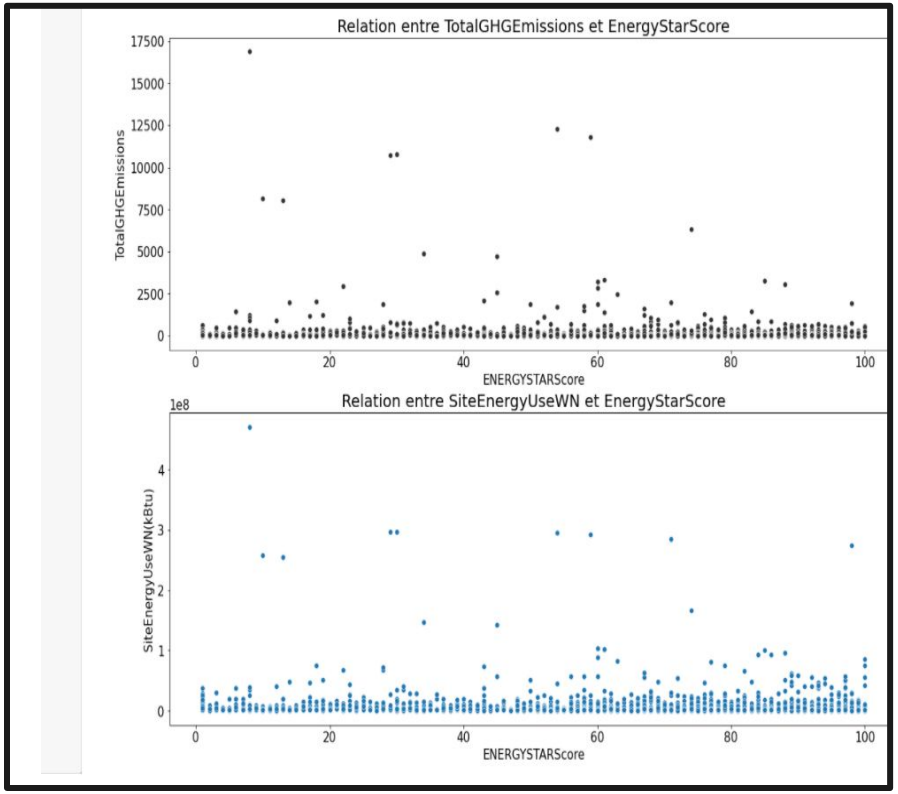
Variable de la consommation :SiteEnergyUse
Corrélation avec les variables :

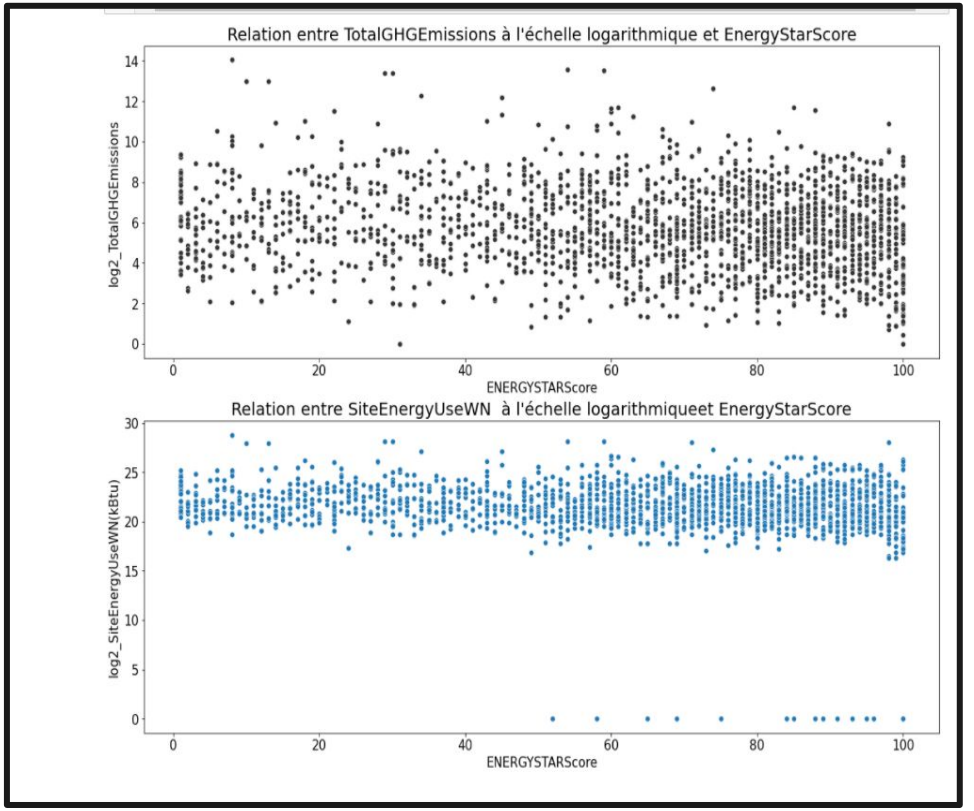
- PropertyGFATotal
- PropertyGFABuilding(s)
- LargestPropertyUseTypeGFA

Variable de l'émission : TotalGHGEmissions
Corrélation avec les variables :

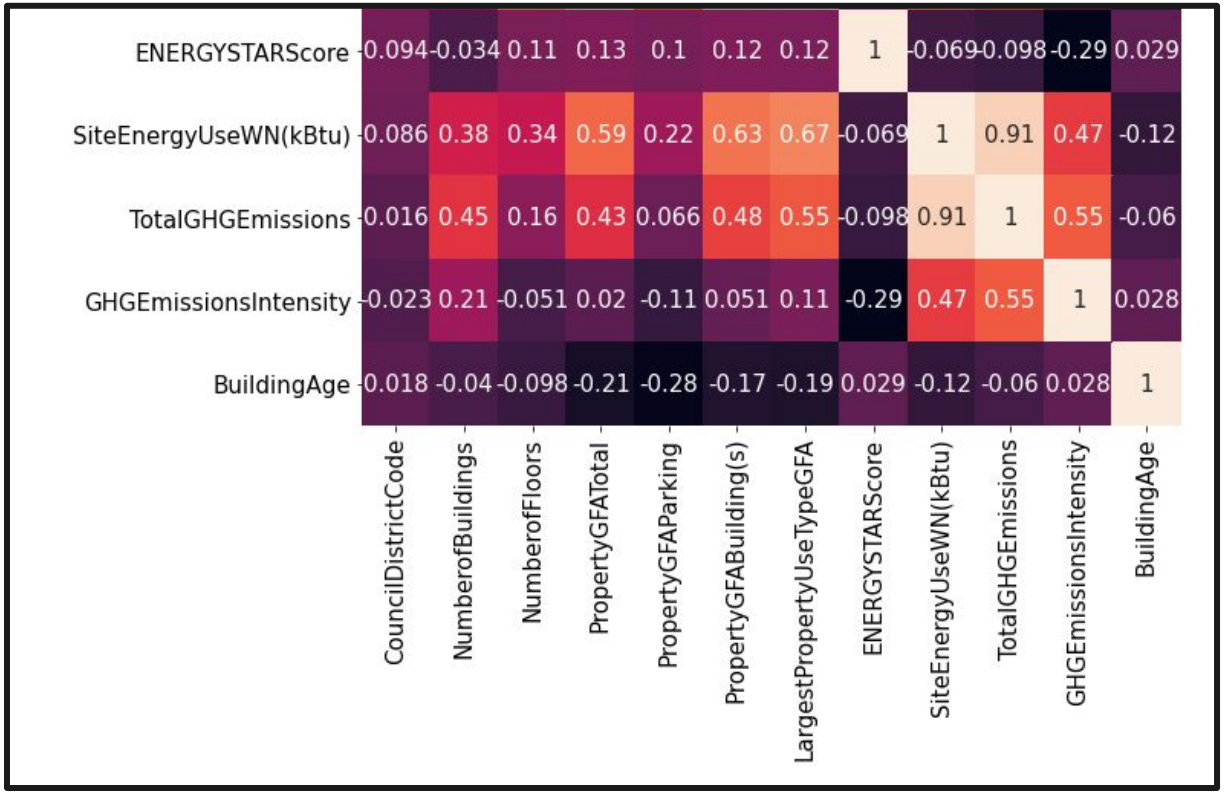
- PropertyGFATotal
- PropertyGFABuilding(s)
- LargestPropertyUseTypeGFA

# Relation entre les variables cibles et EnergyStarScore



# Exploration des données : Matrice de corrélation




EnergyStarScore : Aucune corrélation remarquable avec les autres variables

# Sélection des variables pour les modèles :

Quantitatives :

- NumberofBuildings
- NumberofFloors
- BuildingAge
- PropertyGFATotal
- PropertyGFAParking
- PropertyGFABuilding(s)

Qualitatives :

- BuildingType
- PrimaryPropertyType
- LargestPropertyUseType
- Neighborhood
- DefaultData
- ComplianceStatus

# Préparation des données :

## Démarche de modélisation

1. Séparation des outputs et des inputs
2. Encodage et Normalisation des variables d'entrée (features)
3. Séparation du jeu de données : Train /Test
4. Cross Validation : ShuffleSplit
5. GridSearchCV : Optimisation des hyperparamètres.
6. Calcul des métriques et choix du modèle le plus performant

# Préparation des données :
## Encodage et transformation

### Variables quantitatives :

- Normalisation selon la méthode de RobustScaler .

### Variables qualitatives :

- Encodage selon la méthode de OneHotEncoder .

### Variables de sortie:

- Initialement : Variable de sortie initiales .
- Transformation Box Cox et transformation logarithmique ->Évaluation des performances des modèles avant et  aprés transformation.

# Modélisation :

## GridSearchCV

```
    'linear_regression' : {
        'model': LinearRegression(),
        'params': {
            'normalize': [True, False]
        }
    },
    'lasso': {
        'model': Lasso(),
        'params': {
            'alpha': np.logspace(-5,1,20),
            'selection': ['random', 'cyclic']
        }
    },
    'Ridge': {
        'model': Ridge(),
        'params': {
            'alpha': np.logspace(-5,5,20),
        }
    },
    'Elasticnet': {
        'model': ElasticNet(),
        'params': {
            'alpha': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100],
            'l1_ratio' :np.arange(0.0, 1.0, 0.1),
            'tol': [0.1,0.01,0.001,0.0001]
        }
    },
    'XGBRegressor': {
        'model': XGBRegressor(),
        'params': {
            'n_estimators' : [100,500,1000,2000]
        }},
    'Random Forest Regressor': {
        'model': RandomForestRegressor(),
        'params': {
             'n_estimators' : [10,50,100,300,500],
             'min_samples_leaf' : [1,3,5,10],
              'max_features': ['auto', 'sqrt']
        }
    }
}
```

| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | -1.609336e+21 | {'normalize': False} |
| 1 | lasso | 4.674439e-01 | {'alpha': 1.0, 'selection': 'random'} |
| 2 | Ridge | 5.085698e-01 | {'alpha': 6.158482110660254} |
| 3 | Elasticnet | 5.085501e-01 | {'alpha': 0.01, 'l1_ratio': 0.6000000000000001, 'tol': 0.1} |
| 4 | XGBRegressor | 7.056387e-01 | {'n_estimators': 2000} |
| 5 | Random Forest Regressor | 6.964435e-01 | {'max_features': 'sqrt', 'min_samples_leaf': 1, 'n_estimators': 500} |

# GridSearchCV

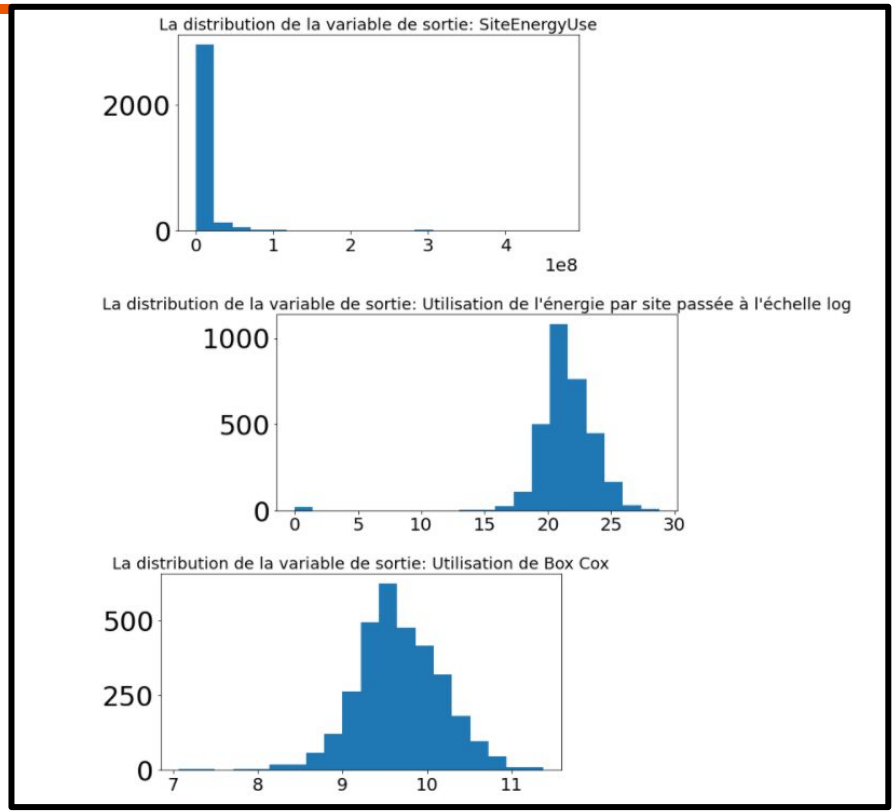
## SiteEnergyUse

| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | -1.609336e+21 | {'normalize': False} |
| 1 | lasso | 4.674439e-01 | {'alpha': 10.0, 'selection': 'random'} |
| 2 | Ridge | 5.085698e-01 | {'alpha': 6.1584821106602545} |
| 3 | Elasticnet | 5.085501e-01 | {'alpha': 0.01, 'l1_ratio': 0.6000000000000001, 'tol': 0.1} |
| 4 | XGBRegressor | 7.056387e-01 | {'n_estimators': 2000} |
| 5 | Random Forest Regressor | 6.964435e-01 | {'max_features': 'sqrt', 'min_samples_leaf': 1, 'n_estimators': 500} |

## TotalGHGEmissions

| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | -3.454103e+21 | {'normalize': False} |
| 1 | lasso | 2.891551e-01 | {'alpha': 4.832930238571752, 'selection': 'random'} |
| 2 | Ridge | 3.054837e-01 | {'alpha': 20.6913808111479} |
| 3 | Elasticnet | 3.093421e-01 | {'alpha': 0.01, 'l1_ratio': 0.30000000000000004, 'tol': 0.0001} |
| 4 | XGBRegressor | 6.198236e-01 | {'n_estimators': 2000} |
| 5 | Random Forest Regressor | 6.408416e-01 | {'max_features': 'sqrt', 'min_samples_leaf': 1, 'n_estimators': 500} |

# Transformation des variables cibles :



# Les scores aprés le passage au log

## SiteEnergyUse

| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | -6.410666e+20 | {'normalize': False} |
| 1 | lasso | 5.515492e-01 | {'alpha': 0.00018329807108324357, 'selection': 'random'} |
| 2 | Ridge | 5.509921e-01 | {'alpha': 0.5455594781168515} |
| 3 | ElasticNet | 5.514251e-01 | {'alpha': 0.0001, 'l1_ratio': 0.9, 'tol': 0.1} |
| 4 | XGBRegressor | 8.063267e-01 | {'n_estimators': 500} |
| 5 | Random Forest Regressor | 7.992095e-01 | {'max_features': 'auto', 'min_samples_leaf': 1, 'n_estimators': 500} |

## TotalGHGEmissions

| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | -9.074591e+20 | {'normalize': False} |
| 1 | lasso | 4.239548e-01 | {'alpha': 0.00018329807108324357, 'selection': 'random'} |
| 2 | Ridge | 4.240486e-01 | {'alpha': 0.5455594781168515} |
| 3 | ElasticNet | 4.240177e-01 | {'alpha': 0.0001, 'l1_ratio': 0.8, 'tol': 0.1} |
| 4 | XGBRegressor | 7.095348e-01 | {'n_estimators': 500} |
| 5 | Random Forest Regressor | 7.058417e-01 | {'max_features': 'auto', 'min_samples_leaf': 1, 'n_estimators': 500} |

# GridSearchCV

Avant transformation

Aprés transformation

## SiteEnergyUse

| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | -1.609336e+21 | {'normalize': False} |
| 1 | lasso | 4.674439e-01 | {'alpha': 10.0, 'selection': 'random'} |
| 2 | Ridge | 5.085698e-01 | {'alpha': 6.1584821106602545} |
| 3 | Elasticnet | 5.085501e-01 | {'alpha': 0.01, 'l1_ratio': 0.6000000000000001, 'tol': 0.1} |
| 4 | XGBRegressor | 7.056387e-01 | {'n_estimators': 2000} |
| 5 | Random Forest Regressor | 6.964435e-01 | {'max_features': 'sqrt', 'min_samples_leaf': 1, 'n_estimators': 500} |

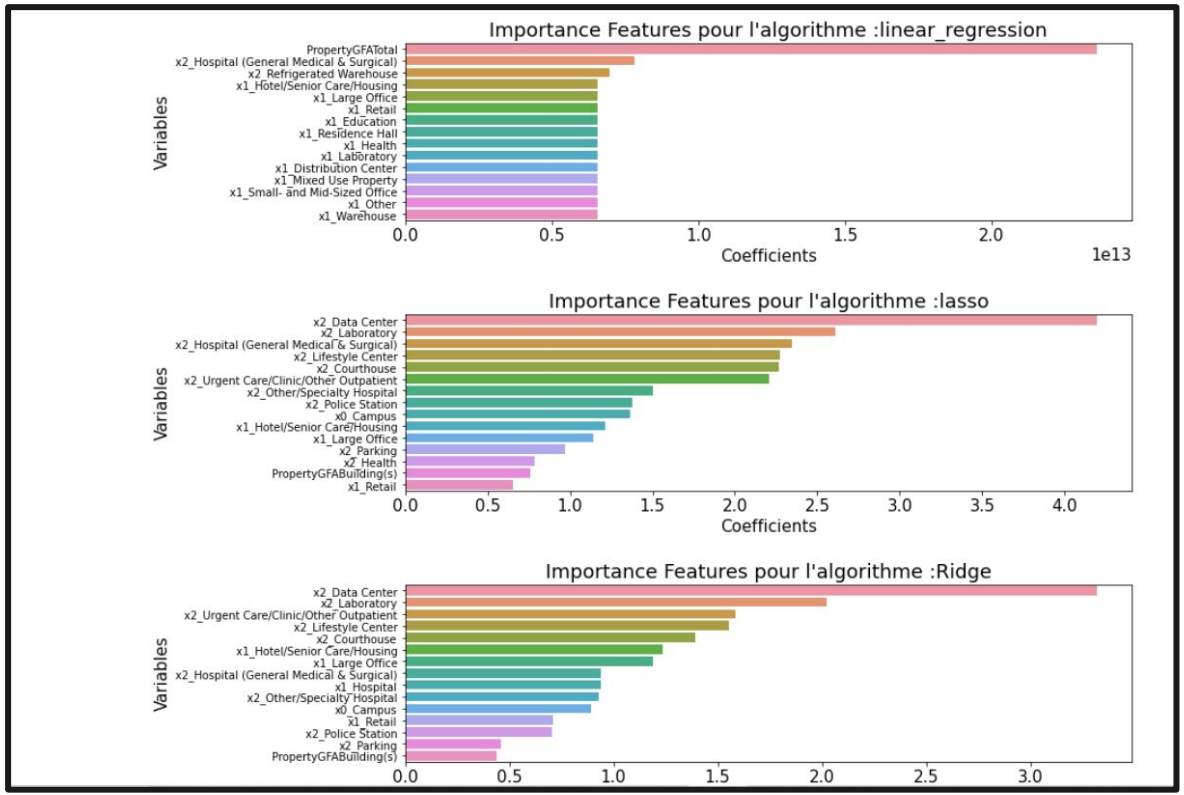
| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | -6.410666e+20 | {'normalize': False} |
| 1 | lasso | 5.515492e-01 | {'alpha': 0.0001832980710832435 7, 'selection': 'random'} |
| 2 | Ridge | 5.509921e-01 | {'alpha': 0.5455594781168515} |
| 3 | Elasticnet | 5.514251e-01 | {'alpha': 0.0001, 'l1_ratio': 0.9, 'tol': 0.1} |
| 4 | XGBRegressor | 8.063267e-01 | {'n_estimators': 500} |
| 5 | Random Forest Regressor | 7.992095e-01 | {'max_features': 'auto', 'min_samples_leaf': 1, 'n_estimators': 500} |

## TotalGHGEmissions

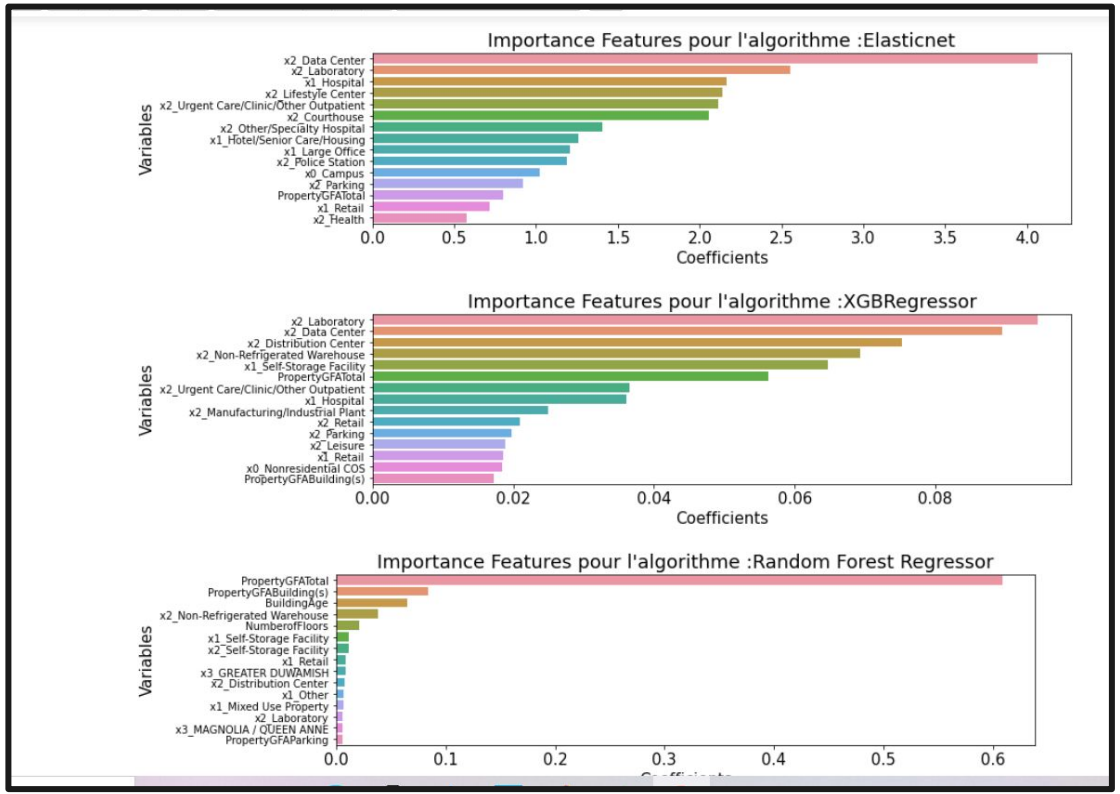
| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | -3.454103e+21 | {'normalize': False} |
| 1 | lasso | 2.891551e-01 | {'alpha': 4.832930238571752, 'selection': 'random'} |
| 2 | Ridge | 3.054837e-01 | {'alpha': 20.6913808111479} |
| 3 | Elasticnet | 3.093421e-01 | {'alpha': 0.01, 'l1_ratio': 0.30000000000000004, 'tol': 0.0001} |
| 4 | XGBRegressor | 6.198236e-01 | {'n_estimators': 2000} |
| 5 | Random Forest Regressor | 6.408416e-01 | {'max_features': 'sqrt', 'min_samples_leaf': 1, 'n_estimators': 500} |

| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | -9.074591e+20 | {'normalize': False} |
| 1 | lasso | 4.239548e-01 | {'alpha': 0.0001832980710832435 7, 'selection': 'random'} |
| 2 | Ridge | 4.240486e-01 | {'alpha': 0.5455594781168515} |
| 3 | Elasticnet | 4.240177e-01 | {'alpha': 0.0001, 'l1_ratio': 0.8, 'tol': 0.1} |
| 4 | XGBRegressor | 7.095348e-01 | {'n_estimators': 500} |
| 5 | Random Forest Regressor | 7.058417e-01 | {'max_features': 'auto', 'min_samples_leaf': 1, 'n_estimators': 500} |

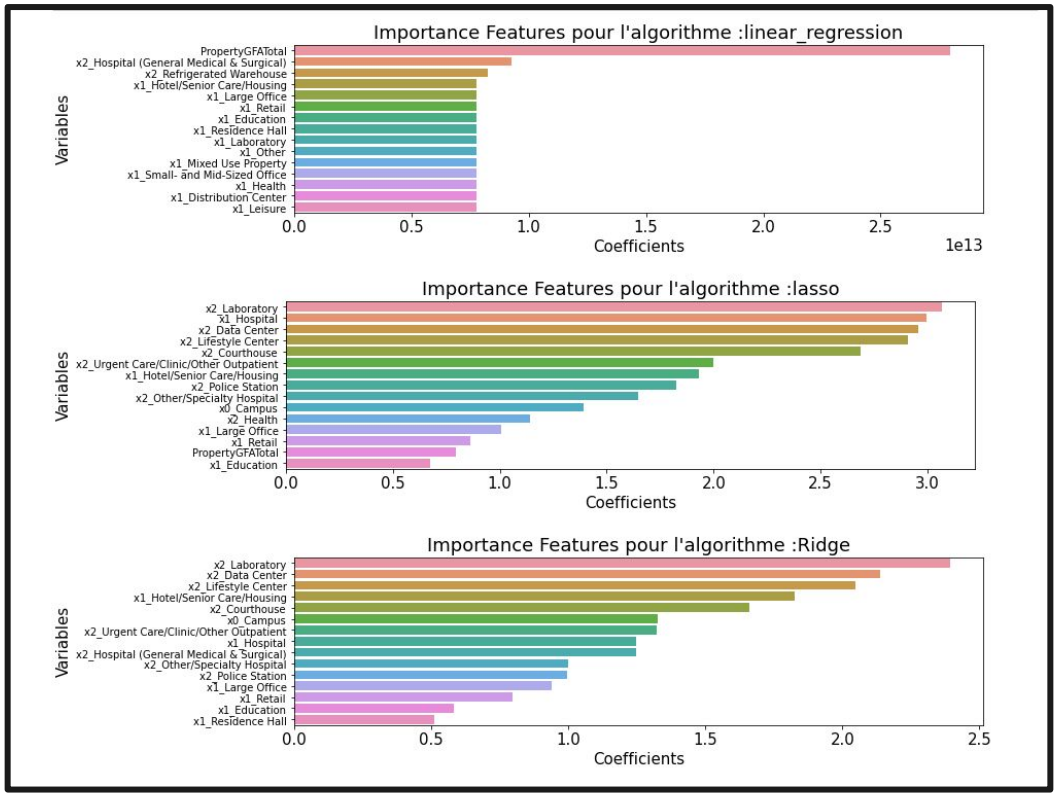
# Importance des variables : features Importance / SiteEnergyUse



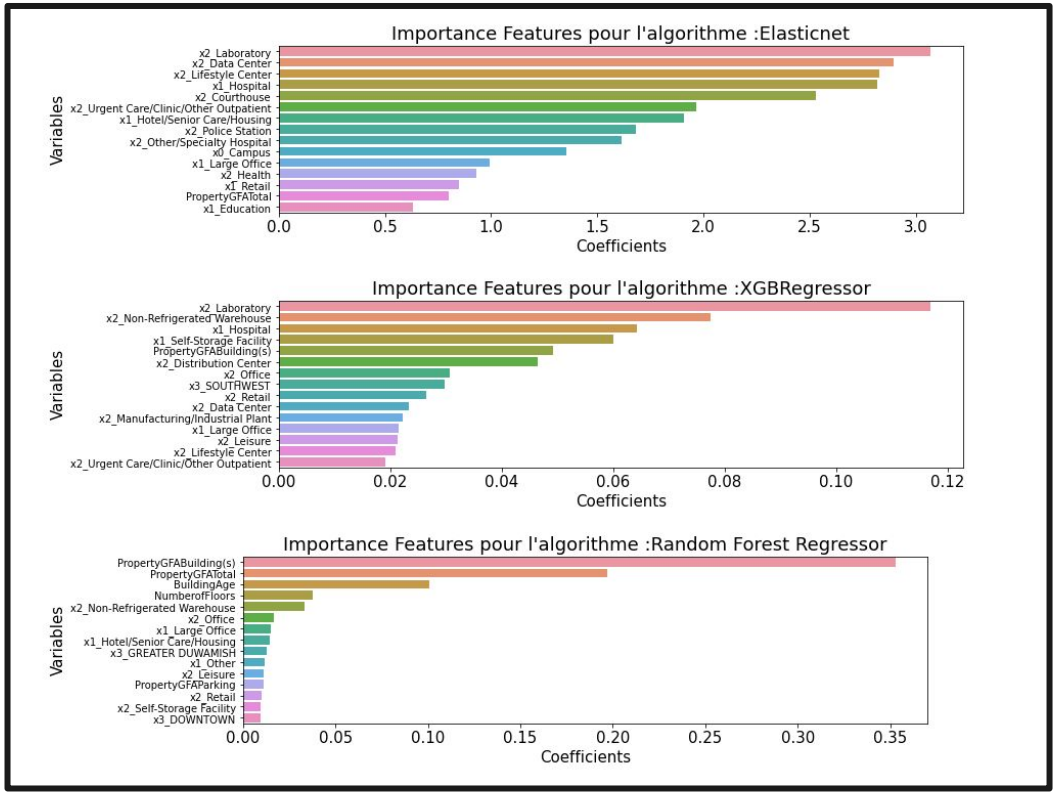
# Importance des variables : features Importance / SiteEnergyUse



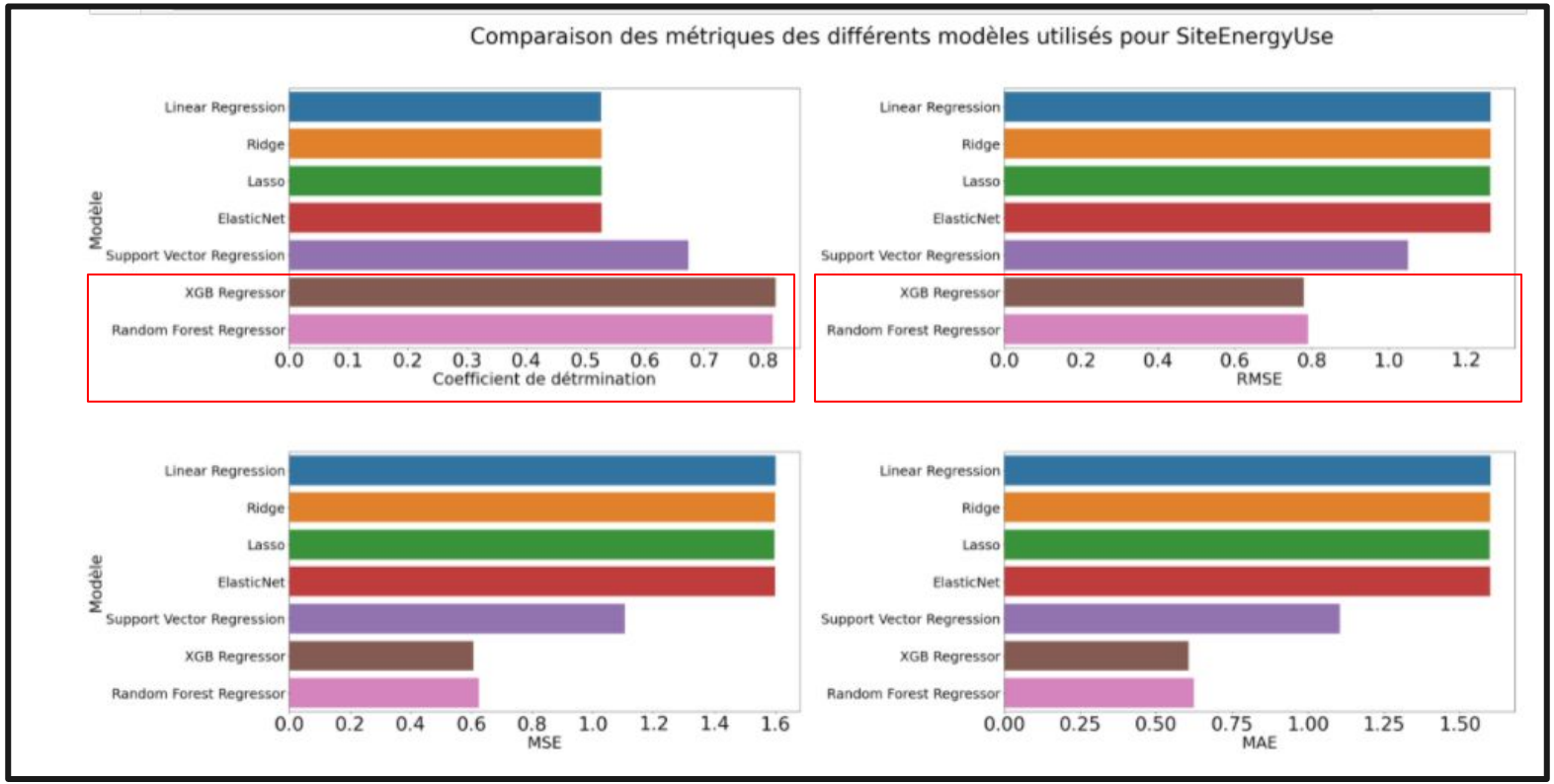
# Importance des variables : features Importance / TotalGHGEmissions



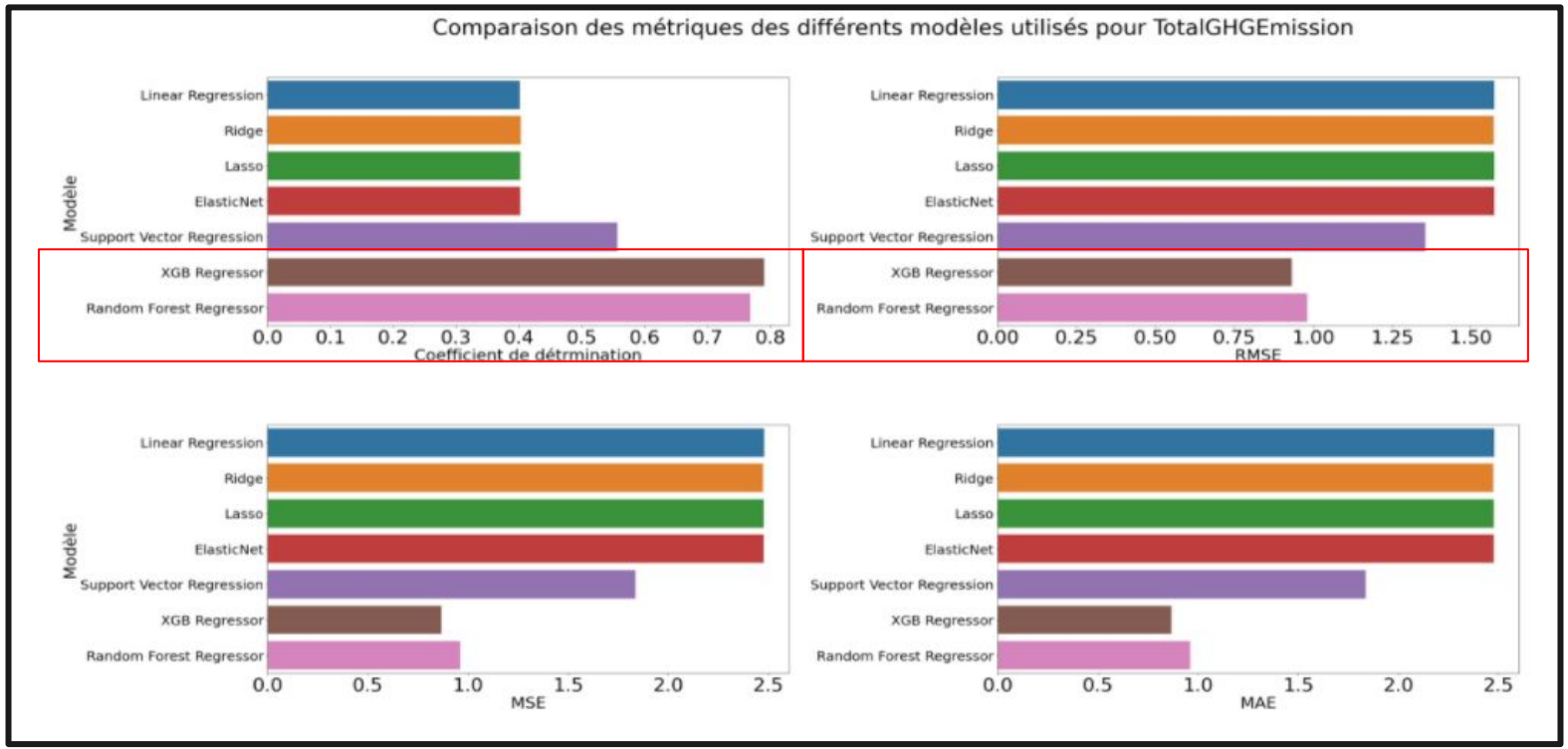
# Importance des variables : features Importance / TotalGHGEmission



# Choix du meilleur modèle pour SiteEnergyUSE



# Choix du meilleur modèle pour TotalEnergyUse



# Evaluation de l'intérêt de EnergyStarScore

## Sans EnergyStarScore

| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | -6.410666e+20 | {'normalize': False} |
| 1 | lasso | 5.514685e-01 | {'alpha': 0.0001832980710832435 7, 'selection': 'random'} |
| 2 | Ridge | 5.509921e-01 | {'alpha': 0.5455594781168515} |
| 3 | Elasticnet | 5.514251e-01 | {'alpha': 0.0001, 'l1_ratio': 0.9, 'tol': 0.1} |
| 4 | XGBRegressor | 8.063267e-01 | {'n_estimators': 500} |
| 5 | Random Forest Regressor | 7.994960e-01 | {'max_features': 'auto', 'min_samples_leaf': 1, 'n_estimators': 500} |

## Avec EnergyStarScore

| model | best_score | best_params |
|---|---|---|
| linear_regression | -7.087228e+22 | {'normalize': False} |
| lasso | 5.261095e-01 | {'alpha': 0.0007847599703514606, 'selection': 'cyclic'} |
| Ridge | 5.253796e-01 | {'alpha': 0.5455594781168515} |
| Elasticnet | 5.262648e-01 | {'alpha': 0.001, 'l1_ratio': 0.9, 'tol': 0.01} |
| XGBRegressor | 7.558192e-01 | {'n_estimators': 2000} |
| Random Forest Regressor | 7.559343e-01 | {'max_features': 'auto', 'min_samples_leaf': 1, 'n_estimators': 500} |

Pour certains modèles améliore très légèrement les performances des modèles  Pour d'autres modèle comme XGBRegressor et Random Forest regressor , il affecte les performances des modèles

# Conclusion :

Le XGBRegresson et le RandomForestRegressor sont nos deux algorithmes les plus performants. Ils obtiennent des résultats très satisfaisants. Avec des r2_score les plus élevés et des RMSE les plus faibles