ANTICIPEZ LES BESOINS EN CONSOMMATION ÉLECTRIQUE DE BÂTIMENTS

Encadrant : Bertrand Beaufils

Evaluateur : Mohammed Sedki

Parcours Data sciences | ABBOUD Marwa | 29 décembre 2020

SOMMAIRE

Présentation de la problématique de la ville de Seattle

- Données de consommation disponibles pour les bâtiments de la ville de Seattle pour les années **2015** et **2016**

- Objectif :

• Prédire les émissions de CO2 et la consommation en énergie des bâtiments :

- **Selon deux modèles différents**

• Evaluer l'intérêt de **l'ENERGY STAR Score***:

- Comparaison de son intérêt en essayant de modéliser avec et sans

ENERGY STAR Score *:Indicateur qui refléter les performances énergétiques d'un bâtiment

Nettoyage de jeu de données

Vérification des ID de bâtiments

Bâtiments communs aux années 2015 et 2016

Calculons l'écart de consommation énergétique entre les 2 années pour les bâtiments communs

Assemblage du data set 2015 et 2016

Pré-Traitement de données

Observation des valeurs manquantes

Supprimer les Colonnes inutiles

Sélectionner les variables cibles

Filtrage

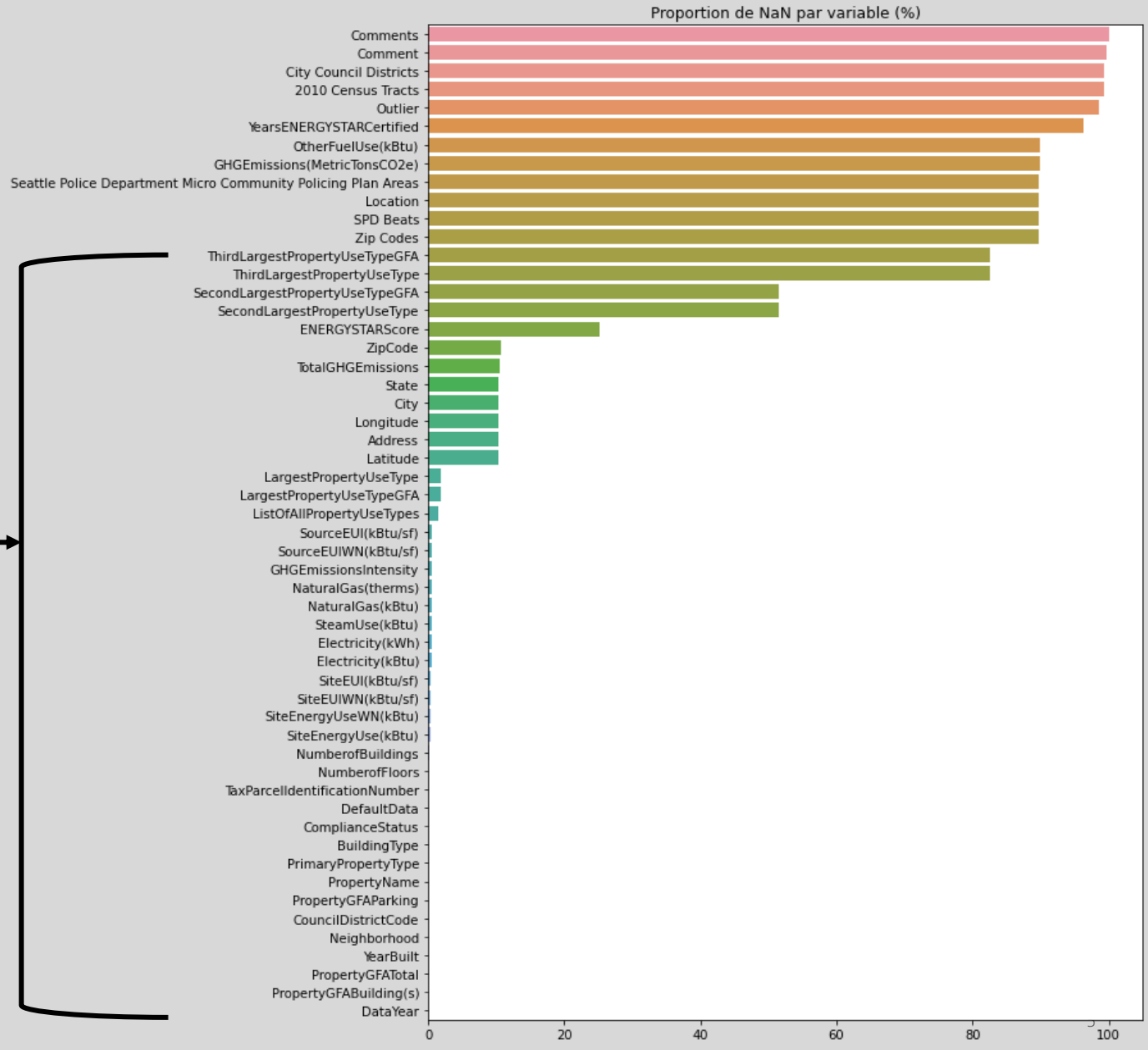
Distribution des variables cibles

Imputation des valeurs manquantes

Encoding pour les variables catégoriels

Etude des valeurs manquantes

Sélectionner les colonnes qui ont **au moins 15% de valeurs renseignées**



DataYear
PropertyGFABuilding(s)
PropertyGFAParking
PropertyGFATotal
YearBuilt
Neighborhood
ComplianceStatus
PropertyName
PrimaryPropertyType
BuildingType
CouncilDistrictCode
DefaultData
TaxParcelIdentificationNumber
NumberofFloors
NumberofBuildings
SiteEnergyUse(kBtu)
SiteEnergyUseWN(kBtu)
SiteEUIWN(kBtu/sf)
SiteEUI(kBtu/sf)
SteamUse(kBtu)
SourceEUIWN(kBtu/sf)
SourceEUI(kBtu/sf)
NaturalGas(kBtu)
Electricity(kWh)
GHGEmissionsIntensity
NaturalGas(therms)
Electricity(kBtu)
ListOfAllPropertyUseTypes
LargestPropertyUseTypeGFA
LargestPropertyUseType
Longitude
City
Latitude
Address
State
TotalGHGEmissions
ZipCode
ENERGYSTARScore
SecondLargestPropertyUseTypeGFA
SecondLargestPropertyUseType
ThirdLargestPropertyUseType
ThirdLargestPropertyUseTypeGFA

Supprimer les features inutiles

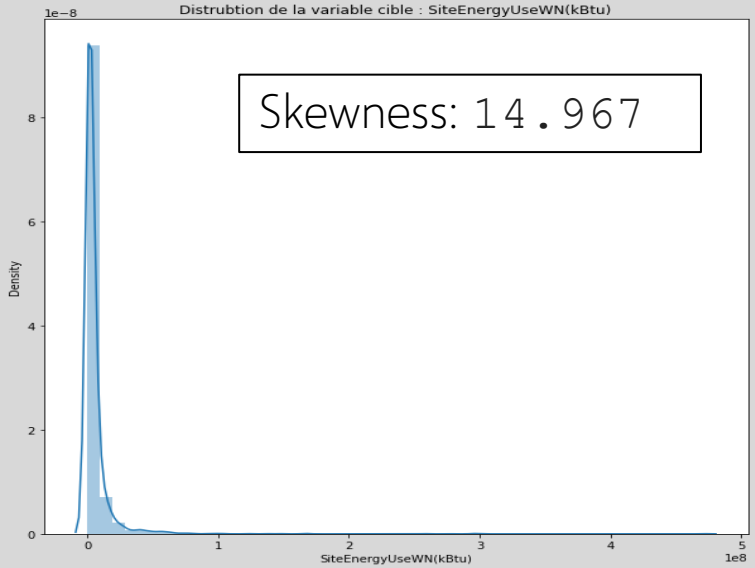
Année des données, ville, nom de la propriété, address, données de la police (SPD Beats) et d'aures..

Sélectionner les variables cibles

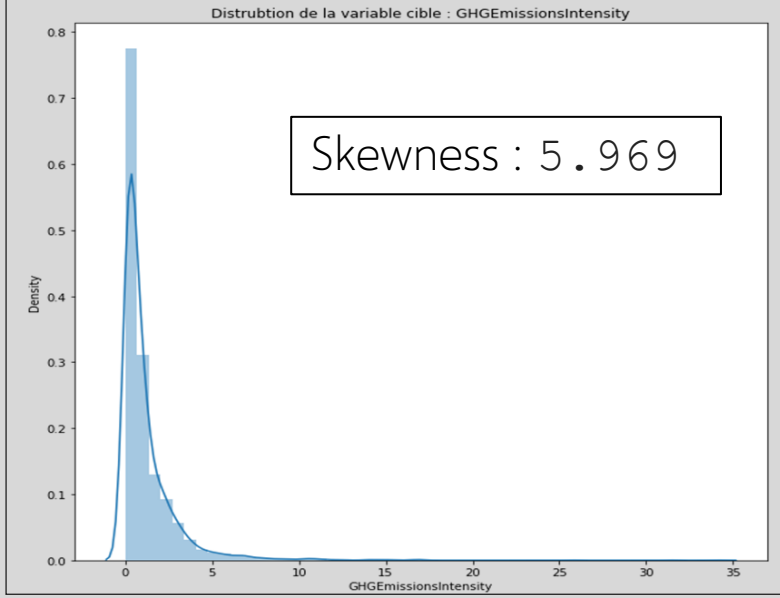
- Emissions des bâtiments
- Consommation totale des bâtiments

Suppression de features **de consommation et d'émissions** pour éviter la **fuite** d'**information**.

- Observation de la distribution pour les variables cibles :

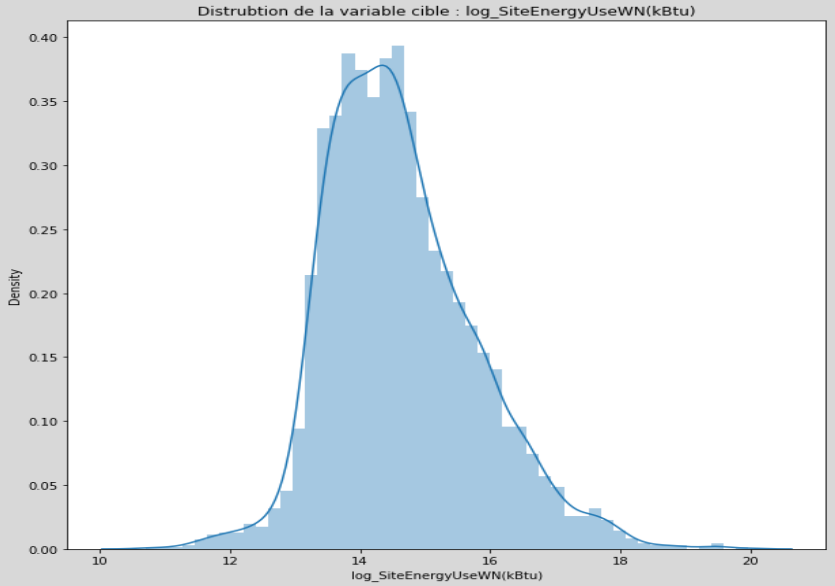


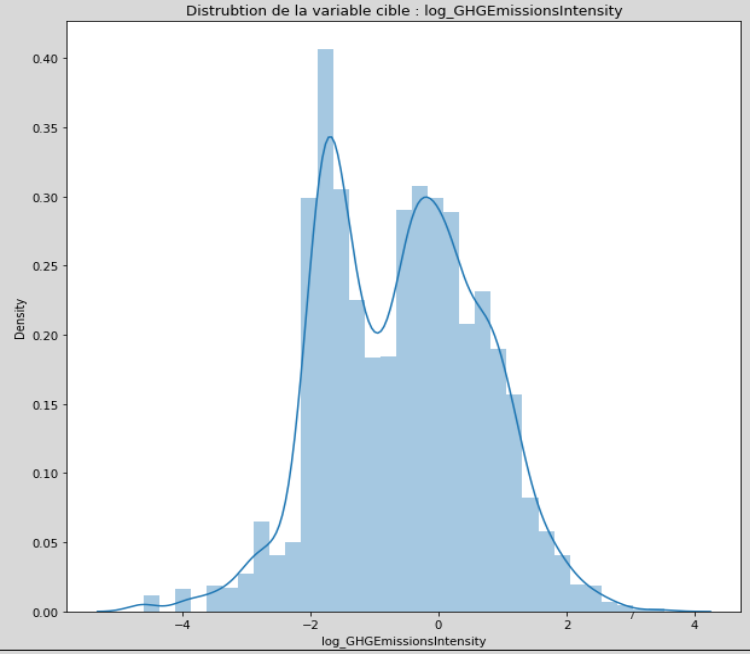
Skewness: 14.967



Skewness : 5.969

Passage au log





Imputation des valeurs manquantes

Colonnes numériques

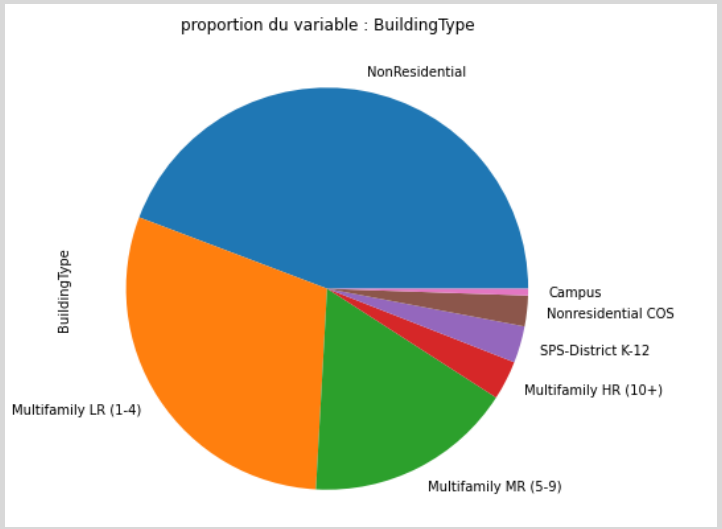
- **LargestPropertyUseTypeGFA**
- **SecondLargestPropertyUseTypeGFA**
- **ThirdLargestPropertyUseTypeGFA**

→ En remplacent les NaNs par la valeur moyennes de «**PrimaryPropertyType** »

Colonnes catégoriels → **Encodage** → Colonnes numériques

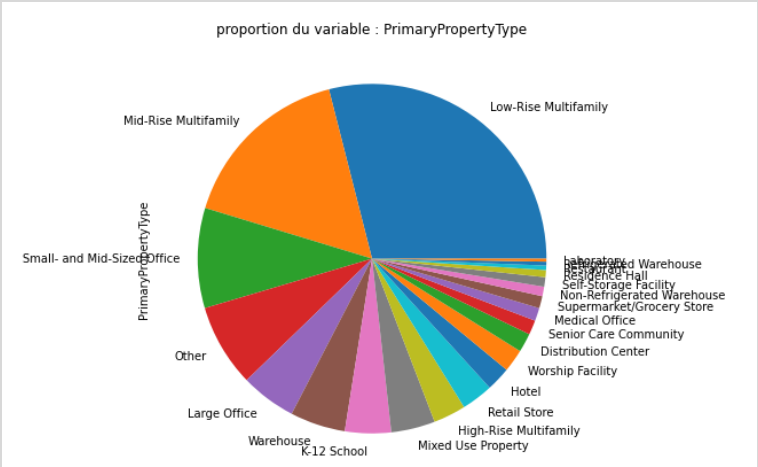
- One Hot Encoding (OHE)
  - Créer autant de colonnes qu'il y a de catégories
- Target Encoding
  - Remplacer chaque catégorie d'une variable par la valeur moyenne de la variable cible des bâtiments de la même catégorie

One Hot Encoding (OHE)



Pour le variable BuildingType :

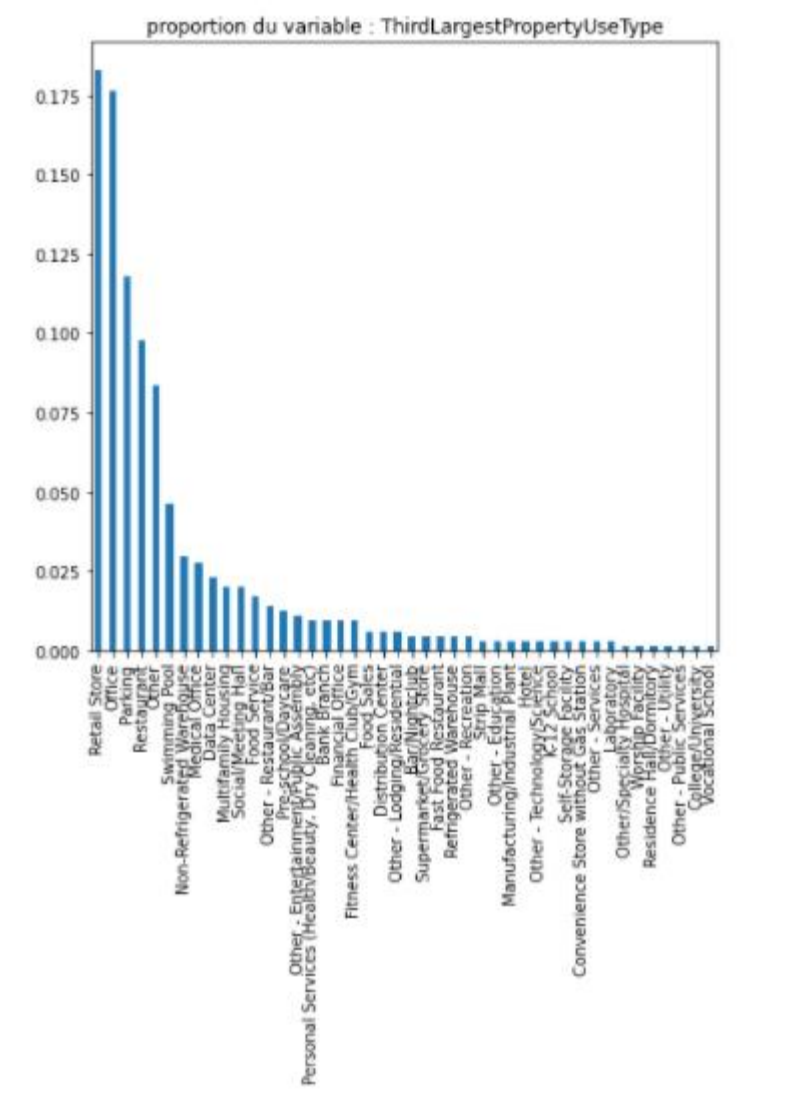
x0_Campus	x0_Multifamily HR (10+)	x0_Multifamily LR (1-4)	x0_Multifamily MR (5-9)	x0_NonResidential	x0_Nonresidential COS	x0_SPS-District K-12
0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0



Pour le variable PrimaryPropertyType :

x1_Distribution Center	x1_High-Rise Multifamily	x1_Hotel	x1_K-12 School	x1_Laboratory	x1_Large Office	x1_Low-Rise Multifamily	x1_Medical Office	x1_Mid-Rise Multifamily	x1_Mixed Use Property	x1_Non-Refrigerated Warehouse
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	1.0	0.0	0.0	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	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	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

Nombre de catégorie est plus élevés



Target Encoding

`GHGEmissionsIntensity`

`SiteEnergyUseWN(kBtu)`

df1

Création de deux df

df2

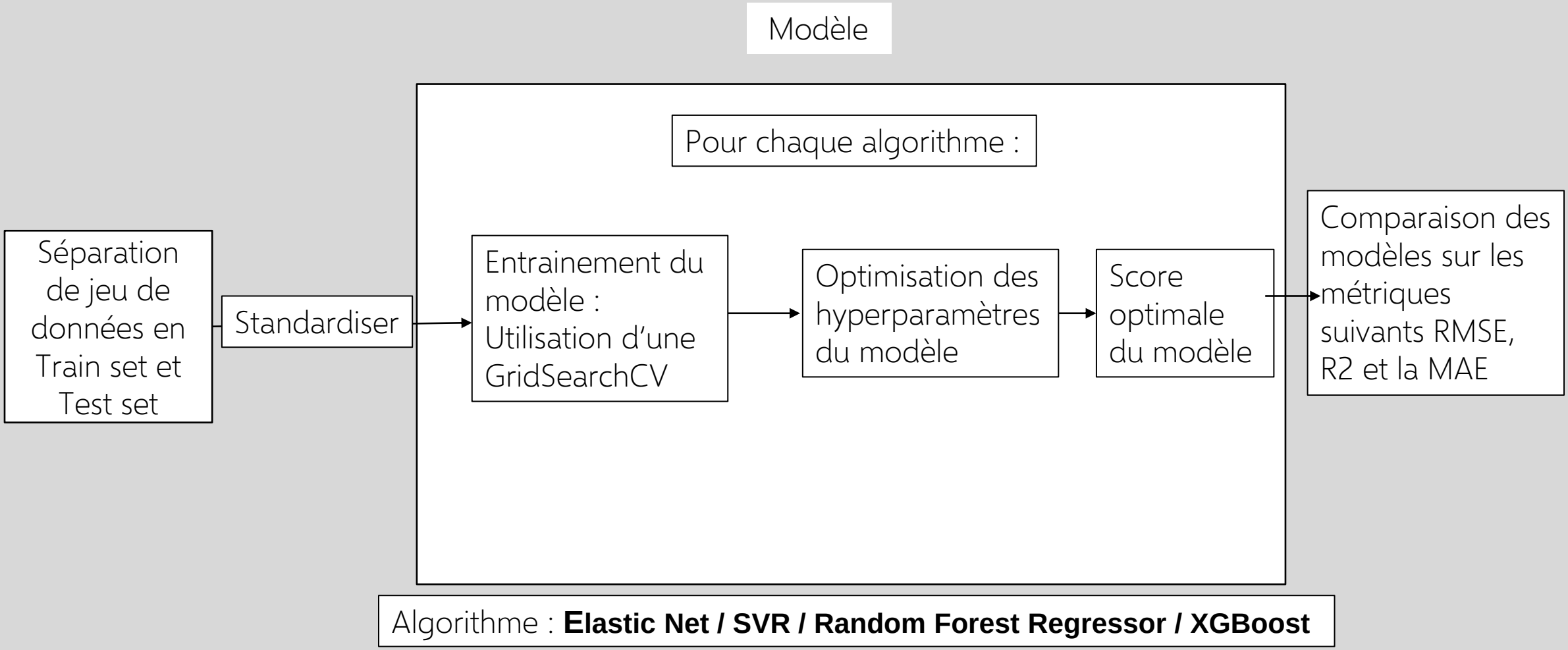
Colonne du df1

ThirdLargestPropertyUseType
15.824469
14.618421
14.618421
14.809576
14.618421
14.618421
14.618421
14.618421
14.618421
14.618421

Colonne du df2

ThirdLargestPropertyUseType
-0.558566
-0.558566
-0.558566
-0.558566
-0.558566
-0.558566
-0.558566
-0.558566
-0.247675
-0.558566

Modélisations



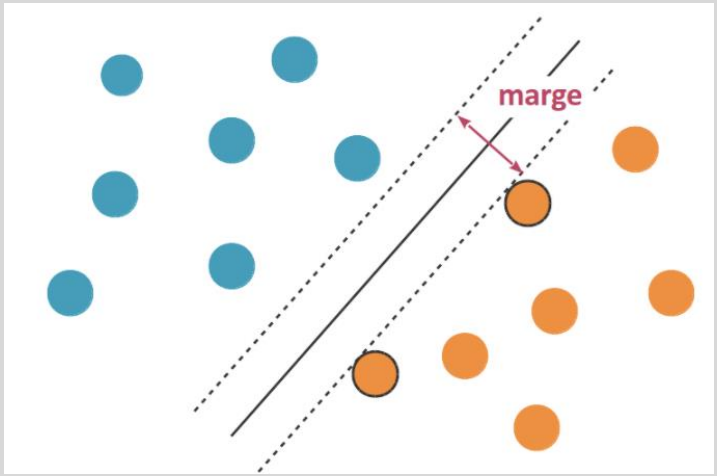
GridSearchCV → Cross validation → CV= 5

	Train set				
Split 1	Val	Train	Train	Train	Train
Split 2	Train	Val	Train	Train	Train
Split 3	Train	Train	Val	Train	Train
Split 4	Train	Train	Train	Val	Train
Split 5	Train	Train	Train	Train	Val

Modèle 1	Modèle 2
R^2=0.92	R^2=0.91
0.88	0.90
0.89	0.91
0.93	0.92
0.86	0.80
0.89	0.92

Modélisations

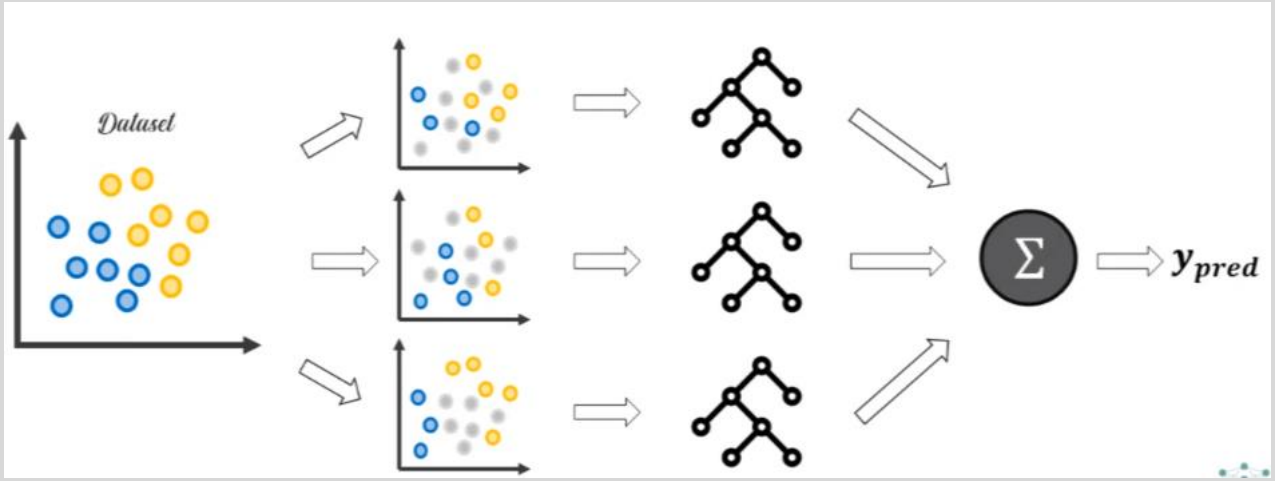
- Régression linéaire
  - Ridge
  - Lasso
  - Elastic Net Regressor → Combine les pénalités L1 et L2
- Support vector machine SVM
  - Support vector regressor (SVR)



- Ensembles Learning
  - Bagging → Random forest
  - Boosting → Gradient Boosting XGBoost

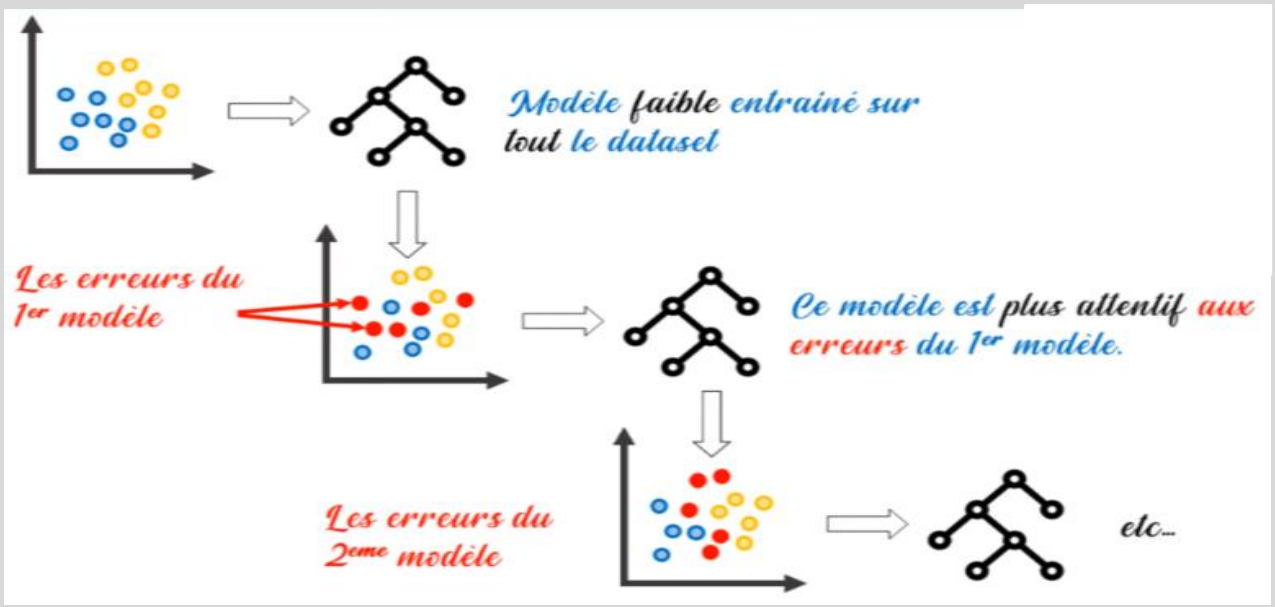
Bagging

Random forest



Boosting

Gradient Boosting XGBoost



Les hyperparamètres pour le modelé de consommation

		Best hyperparamètres
`ElasticNet()`	**Alpha** : [0.0001, 0.001, 0.01, 0.1, 1, 10] **L1_ratio** : np.arange(0.0, 1.0, 0.1)	{'alpha': 0.1, 'l1_ratio': 0.1}
RandomForestRegressor (n_estimators=500,random_state=123)	**max_depth** : [5,10,15,20] **max_features** : ['auto', 'sqrt','log2'] **min_samples_leaf** : [1,3,5,10]	`grid_Rforest.best_params_` {'max_depth': 15, 'max_features': 'sqrt', 'min_samples_leaf': 1}
SVR()	gamma : [1e-2, 1e-3, 1e-4, 1e-5] epsilon : [0.001, 0.01, 0.1, 1], C: [0.001, 0.10, 0.1, 10, 25, 50,100]	`svm_grid1.best_params_` {'C': 10, 'epsilon': 0.1, 'gamma': 0.01}
`XGBRegressor(n_jobs=-1)`	n_estimators : [10,50,100,500,1000,2000] learning_rate : [1,0.1,0.01,0.001,0.0001]	`xgb_grid.best_params_` {'learning_rate': 0.01, 'n_estimators': 2000}

Pour le modelé de consommation

Comparaison des modèles

	Modèle	Score_RMSE	Score_R2	Score_MAE
0	Elasticnet Regression	-0.471677	0.614404	-0.523044
0	Random Forest Regressor	-0.304071	0.751636	-0.392674
0	Support Vector Regressor	-0.341532	0.720614	-0.415541
0	XGBoost	-0.306144	0.750251	-0.395958

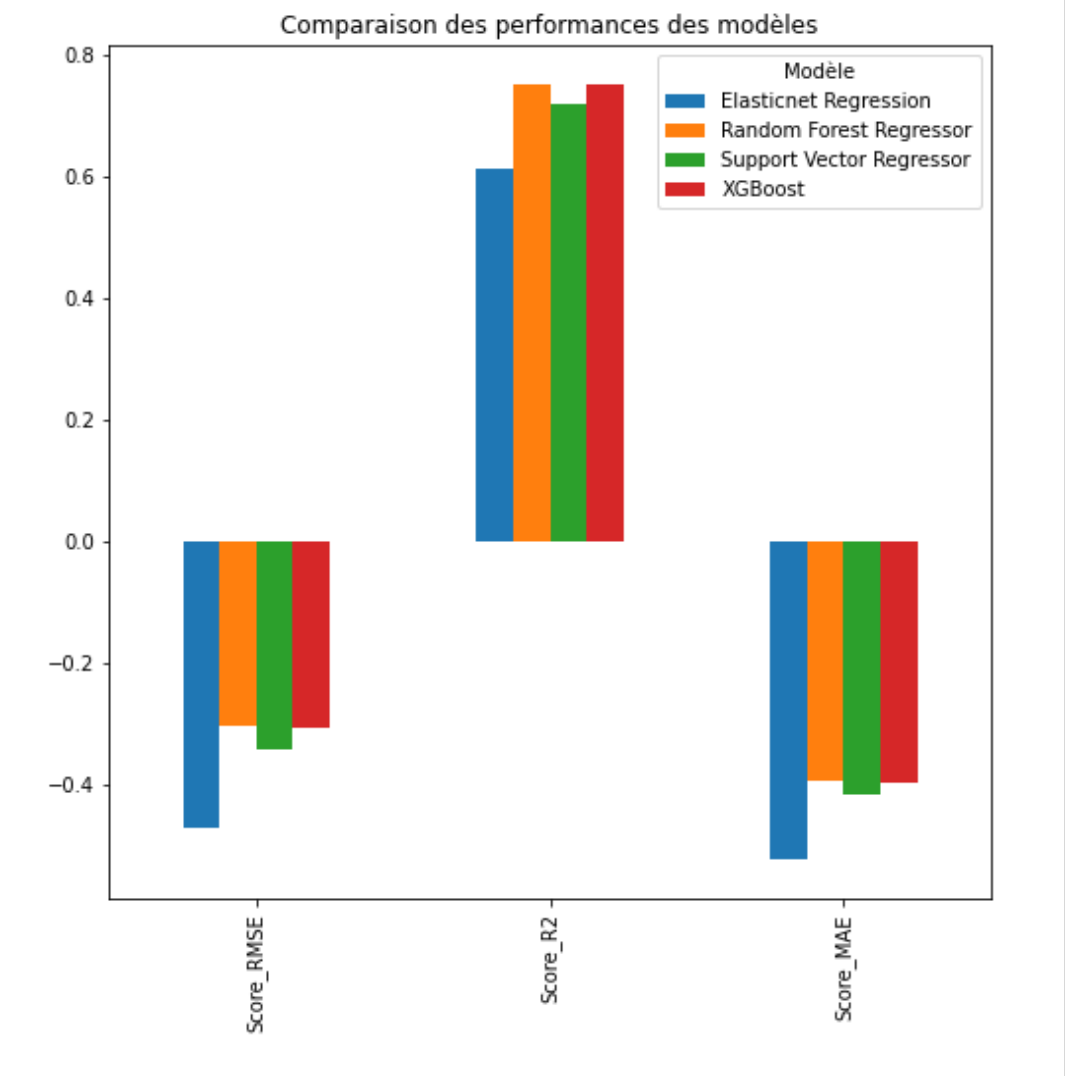
Temps de prédiction pour chaque modelé :

```
1 %timeit svm_grid3.predict(X_test)

10 loops, best of 3: 132 ms per loop
```

```
1 %timeit grid_Rforest3.predict(X_test)

10 loops, best of 3: 98.9 ms per loop
```

```
1 %timeit grid_elastic3.predict(X_test)

The slowest run took 6.76 times longer than the fastest.
10000 loops, best of 3: 159 µs per loop
```

```
1 %timeit xgb_grid3.predict(X_test)

10 loops, best of 3: 40.9 ms per loop
```



Pour le modelé de consommation

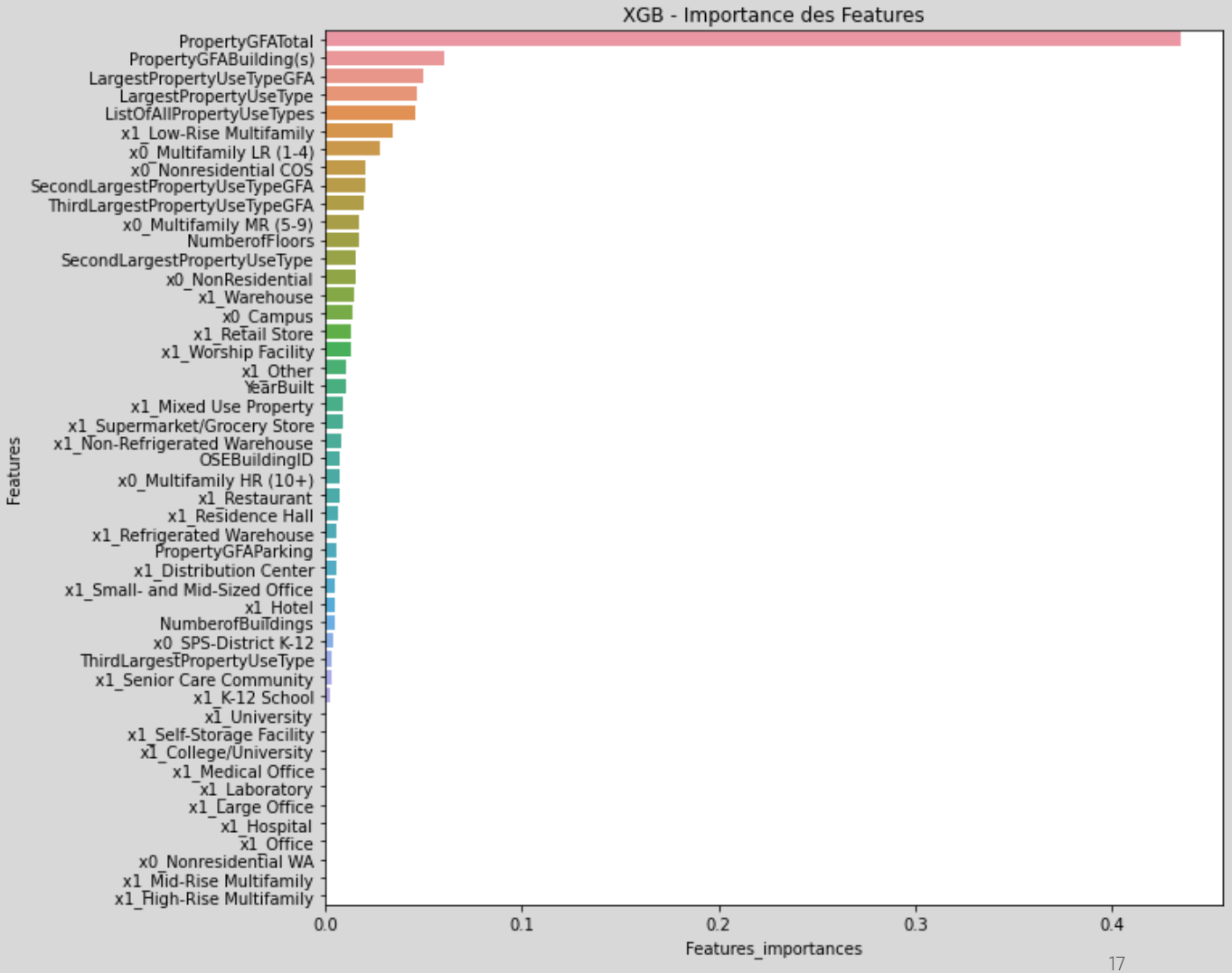
Le modelé finale sélectionnée est le :

XGBoost

```
1 xgb_grid.best_params_
```

```
{'learning_rate': 0.01, 'n_estimators': 2000}
```

Meilleur score de validation en consommation (RMSE= 0,96)



Pour le modelé de consommation

Avec Energy star score

```
1 xgb_grid_ES.best_params_
```

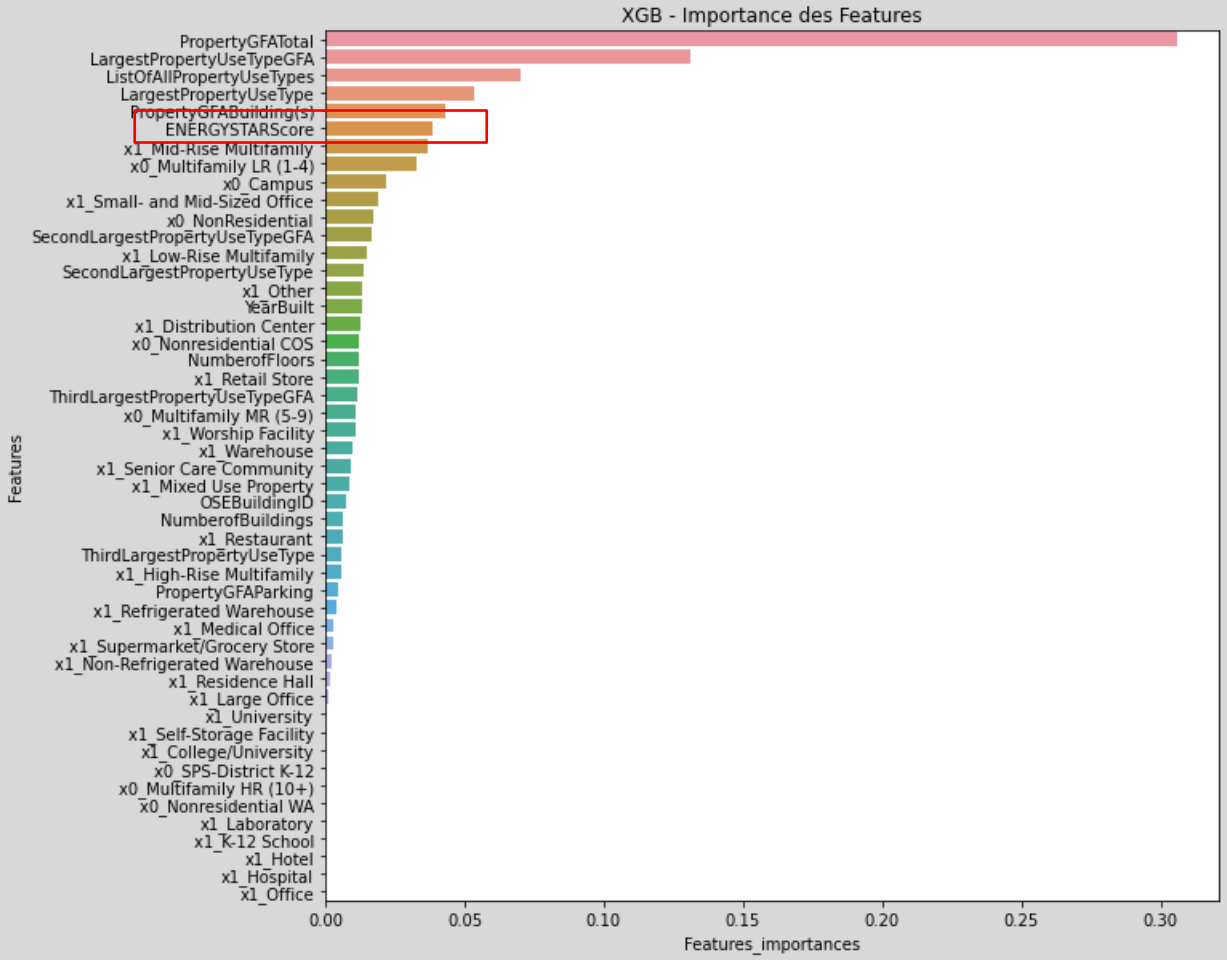
```
{'learning_rate': 0.01, 'n_estimators': 3000}
```

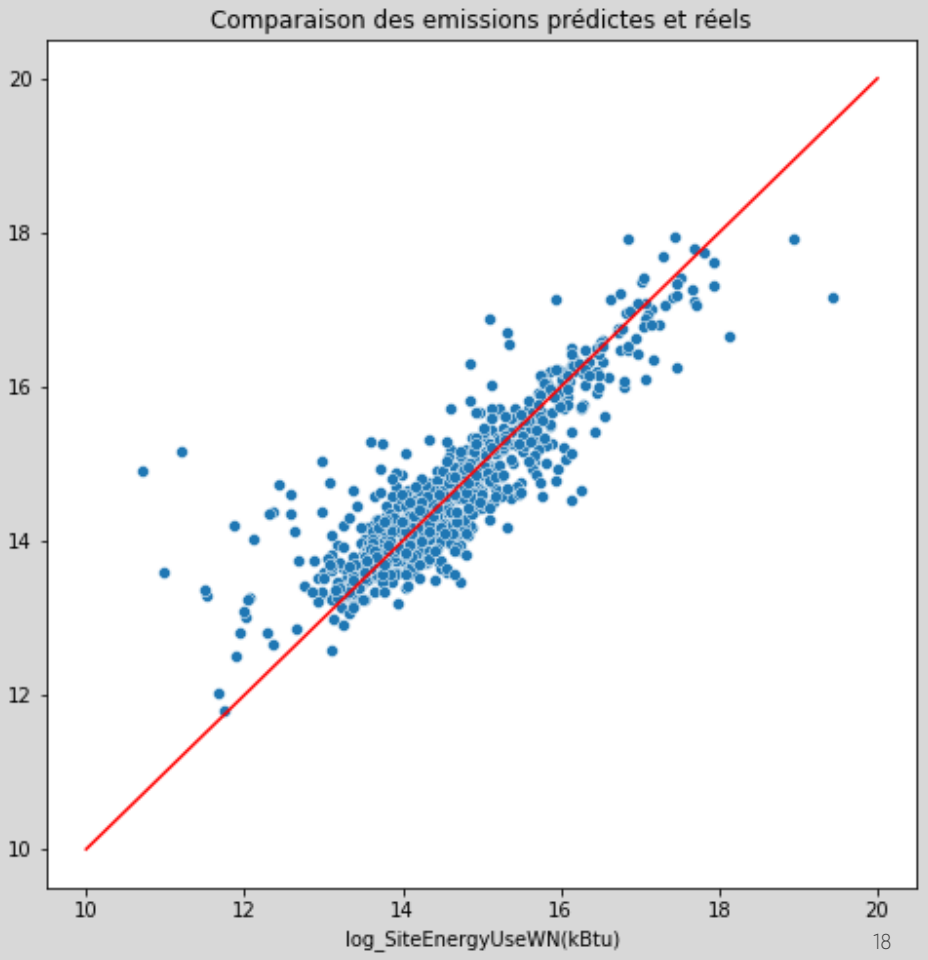
```
1 %timeit xgb_grid_ES.predict(x_Test)
```

```
10 loops, best of 3: 42.6 ms per loop
```

```
1 xgb_grid_ES.best_score_
```

```
-0.2430284243517209
```

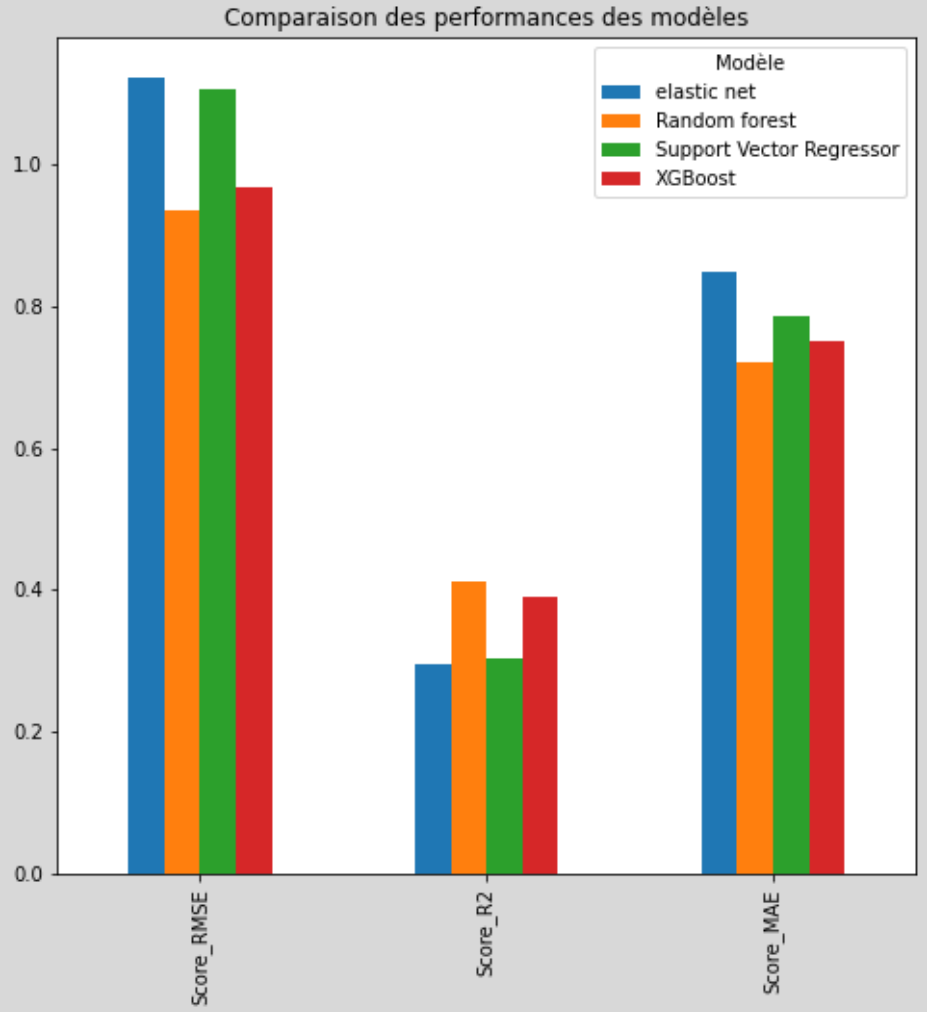




Pour le modelé d'émissions

Prédiction de GHGEmissions Intensity

Modèle	Score_RMSE	Score_R2	Score_MAE
elastic net	-1.121810	0.295098	-0.848687
Random forest	-0.933967	0.411795	-0.720923
Support Vector Regressor	-1.105281	0.304620	-0.785931
XGBoost	-0.967772	0.390858	-0.749866



Temps de prédiction pour chaque modelé :

```
1 %timeit svm_grid_1.predict(XTest)

10 loops, best of 3: 132 ms per loop
```

```
1 %timeit xgb_grid_1.predict(XTest)

10 loops, best of 3: 42.7 ms per loop
```

```
1 %timeit grid_Rforest_1.predict(XTest)

10 loops, best of 3: 79.8 ms per loop
```

```
1 %timeit grid_elastic_1.predict(XTest)

The slowest run took 19.97 times longer than the fastest.
10000 loops, best of 3: 150 µs per loop
```

Pour le modelé d'émissions

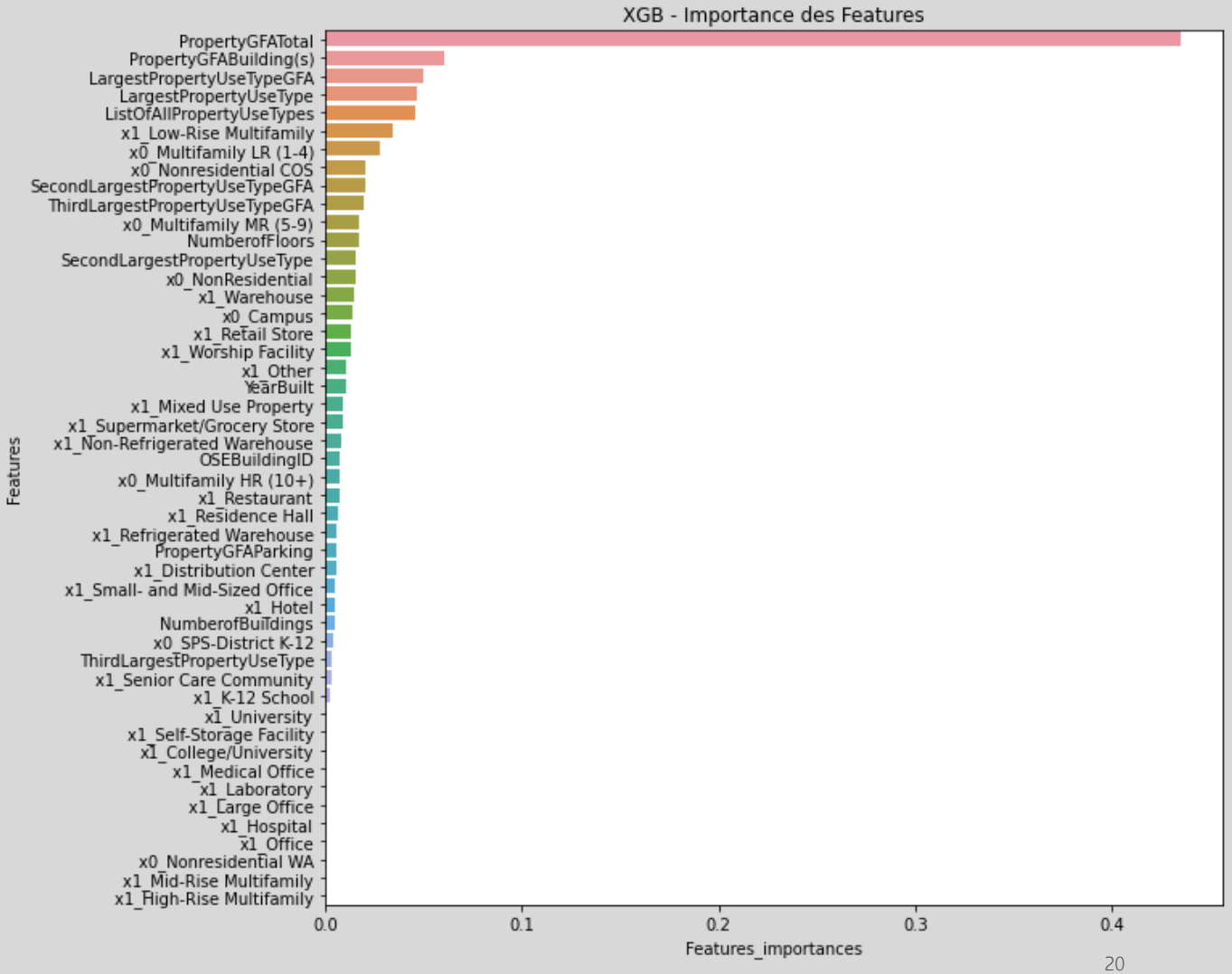
Le modelé finale sélectionnée est le :

XGBoost

```
1 xgb_grid.best_params_
```

```
{'learning_rate': 0.01, 'n_estimators': 2000}
```

Meilleur score de validation en émission (RMSE= 0,96)



Pour le modelé d'émissions

↓

Avec Energy star score

```
1 xgb_grid_E.best_params_
```

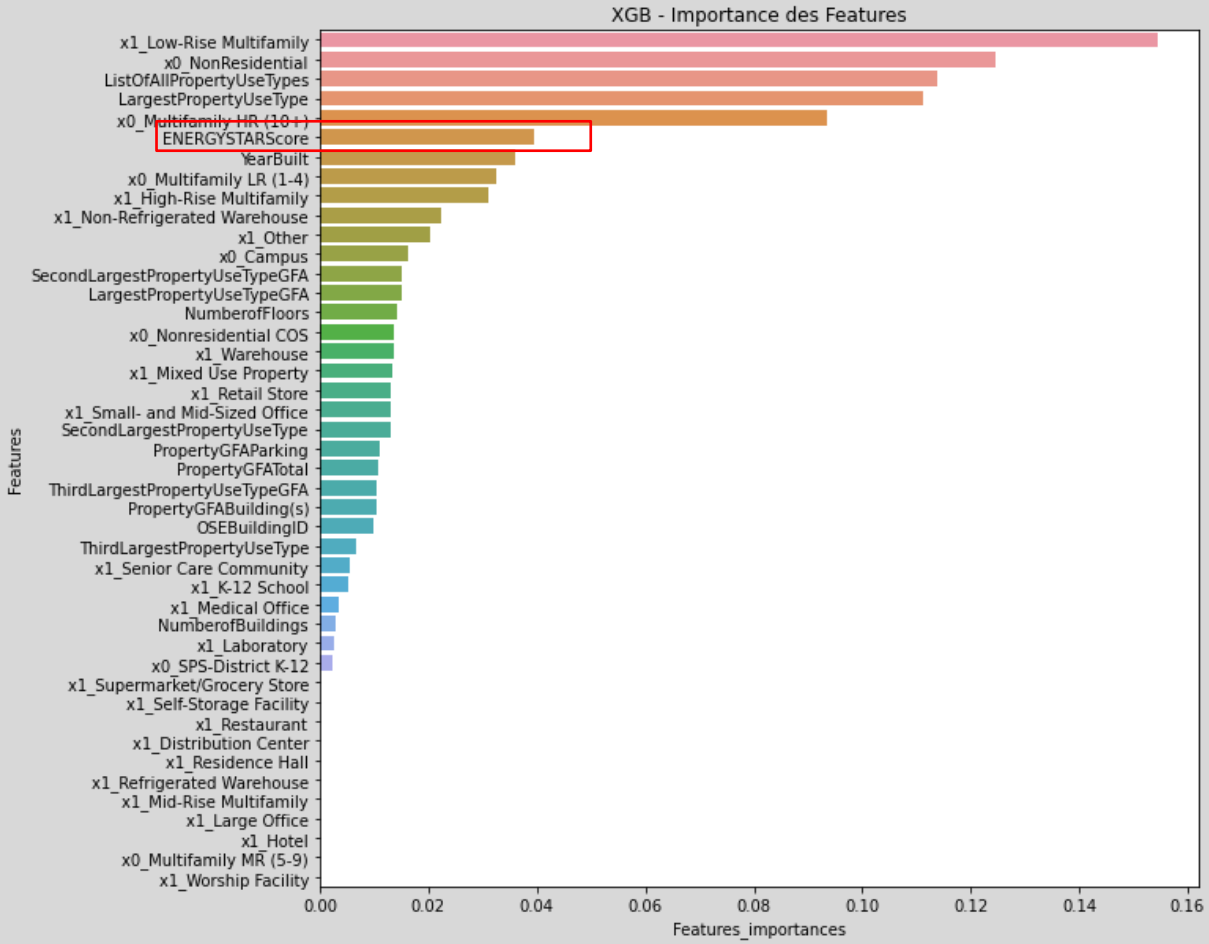
```
{'learning_rate': 0.01, 'n_estimators': 2000}
```

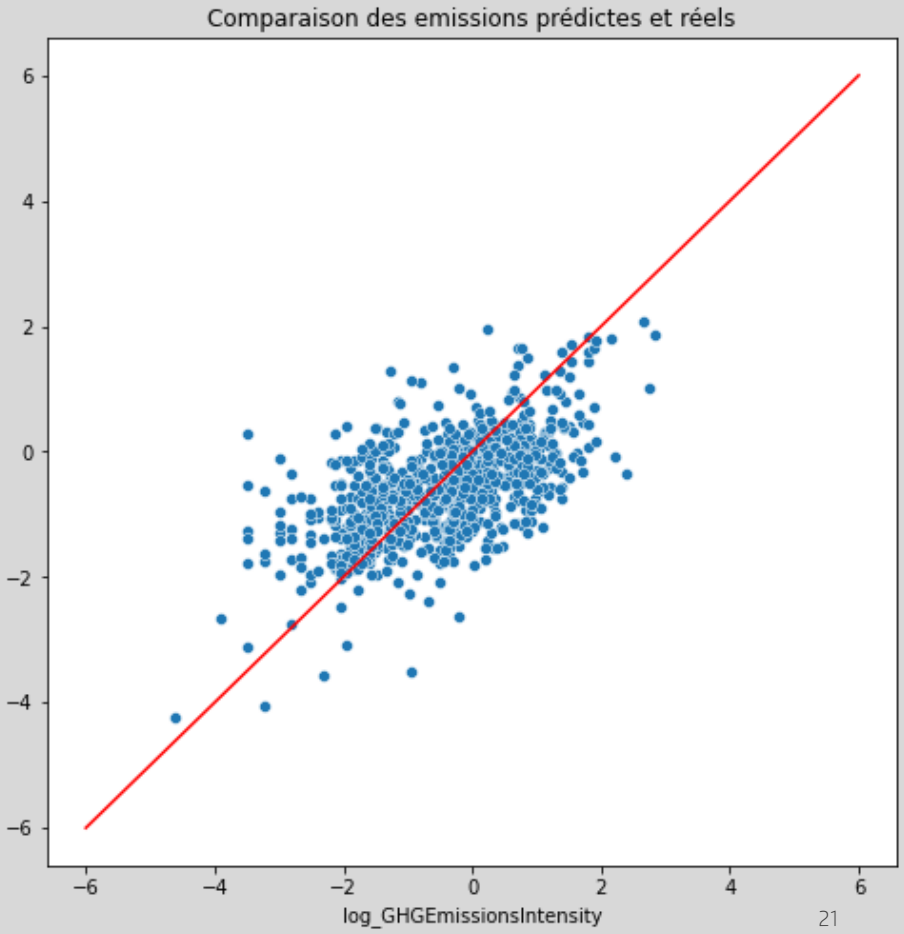
```
1 %timeit xgb_grid_E.predict(X_Test)
```

```
10 loops, best of 3: 41.4 ms per loop
```

```
1 xgb_grid_E.best_score_
```

```
-0.8660695968525746
```





Conclusion

- Le modèle final retenu pour la prédiction de consommation est le XGBoost entraîné sur toutes les features (avec energy star score) et avec les paramètres optimal suivants:
  {'learning_rate': 0.01, 'n_estimators': 2000}

- Le modèle finale retenu pour la prédiction d'émissions est XGBoost entraîné sur toutes les features (avec energy star score) , et avec les paramètres optimal suivants:
{‘learning_rate': 0.01, 'n_estimators': 3000

- Intérêt du ENERGY STAR Score

Modèle	XGBoost(sans ESS)	XGBoost(avec ESS)
Consommation	-0.3	-0.24
Emission	-0.96	-0.86

→ Améliore la performance du modelé

Merci pour votre attention