ANTICIPEZ LES BESOINS EN CONSOMMATION DE BÂTIMENTS

MARWA EL HOURI

APPEL A PROJET

- Objectif: Prédiction de la consommation d'énergie et émissions de CO2 des bâtiments non destinés a l'habitations de la ville de Seattle

PLAN

1. Exploration des données
2. Préparation des variables pour la régression
3. Méthodes de régressions et résultats
4. Conclusion

EXPLORATION DES DONNÉES

- Description du jeu de donnée
- Traitement des valeurs NAN
- Sélection des variables « feature engine »
- Analyse exploratoire

1- EXPLORATION DU JEU DE DONNÉE

- Taille : (3376,46)
- Types

```
dtypes: bool(1), float64(22), int64(8), object(15)
```

	OSEBuildingID	DataYear	BuildingType	PrimaryPropertyType	PropertyName	Address	City	State	ZipCode	TaxParcelIdentificationNumber	...
0	1	2016	NonResidential	Hotel	Mayflower park hotel	405 Olive way	Seattle	WA	98101.0	0659000030	...
1	2	2016	NonResidential	Hotel	Paramount Hotel	724 Pine street	Seattle	WA	98101.0	0659000220	...
2	3	2016	NonResidential	Hotel	5673-The Westin Seattle	1900 5th Avenue	Seattle	WA	98101.0	0659000475	...
3	5	2016	NonResidential	Hotel	HOTEL MAX	620 STEWART ST	Seattle	WA	98101.0	0659000640	...
4	8	2016	NonResidential	Hotel	WARWICK SEATTLE HOTEL (ID8)	401 LENORA ST	Seattle	WA	98121.0	0659000970	...

En gardant les bâtiments non destines a l'habitation
On passe a (2046, 46)

1- EXPLORATION DU JEU DE DONNÉE

- Informations relatives aux bâtiments :
  - OSEBuildingID, PropertyName, Address, TaxParcelIdentificationNumber…
- Informations sur la structure du bâtiment :
  - BuildingType, PrimaryPropertyType, NumberofBuildings, NumberofFloors
- Informations géographiques :
  - Neighborhood, Latitude, Longitude
- Informations techniques :
  - 'SiteEUI(kBtu/sf)', 'SiteEUIWN(kBtu/sf)', 'SourceEUI(kBtu/sf)', 'SourceEUIWN(kBtu/sf)..
- Informations sur les énergies :
  - 'SteamUse(kBtu)', 'Electricity(kWh)', 'Electricity(kBtu)', 'NaturalGas(therms)', 'NaturalGas(kBtu)'
- Les variables étiquettes :
  - 'SiteEnergyUse(kBtu)' et TotalGHGEmissions

2- TRAITEMENT DES VALEURS NAN

- Remplacement des valeurs manquantes suivants leurs natures
  - Remplacement par 0
  - Remplacement par la moyenne
  - Remplacement par la valeur d'une autre colonne

3- TRANSFORMATION DES VARIABLES ENERGIE

- Transformation des variables energies (Electricity(kBtu), NaturalGas(kBtu) et SteamUse(kBtu) en variables booléenne
  - I s'il y a une consommation,
  - 0 sinon

```
df.loc[df['SteamUse(kBtu)']!=0, 'SteamUse(kBtu)']=1
df.loc[df['Electricity(kBtu)']!=0,'Electricity(kBtu)']=1
df.loc[df['NaturalGas(kBtu)']!=0,'NaturalGas(kBtu)']=1
```

4- SÉLECTION DES VARIABLES- « FEATURE ENGINE »

- DropDuplicatedFeatures
  - 0 variables

```
1  sel_drop.features_to_drop_
```

```
['DataYear',
 'City',
 'State',
 'NumberofBuildings',
 'YearsENERGYSTARCertified',
 'SteamUse(kBtu)',
 'DefaultData',
 'ComplianceStatus',
 'Outlier']
```

DropConstantFeatures

4- SÉLECTION DES VARIABLES

- Enlever les variables sur les informations particulières des bâtiments
  - `['OSEBuildingID','PropertyName','Address', 'ZipCode', 'TaxParcelIdentificationNumber']`
- Enlever les variables d'energie
  - `['Electricity(kWh),'NaturalGas(therms)']` (les autres ont été transformer en variables booléennes)
- Enlever les variables relatives aux mesure d'energies :
  - `['GHGEmissionsIntensity','SiteEUI(kBtu/sf)', 'SiteEUIWN(kBtu/sf)', 'SourceEUI(kBtu/sf)', 'SourceEUIWN(kBtu/sf)', 'SiteEnergyUseWN(kBtu)']`
- Enlever les variables corrélés avec les variables cibles (dépend du modèle)

5- ANALYSE EXPLORATOIRE – TRAIN TEST SPLIT

- Division en Train set et Test set (70%/30%)
- Analyse exploratoire a été faite sur le train set
- Taille (1420, 19)

```
1 categ=X_train.dtypes[df.dtypes=='object'].index
2 categ
```

```
Index(['BuildingType', 'PrimaryPropertyType', 'Neighborhood',
       'LargestPropertyUseType', 'SecondLargestPropertyUseType',
       'ThirdLargestPropertyUseType'],
      dtype='object')
```

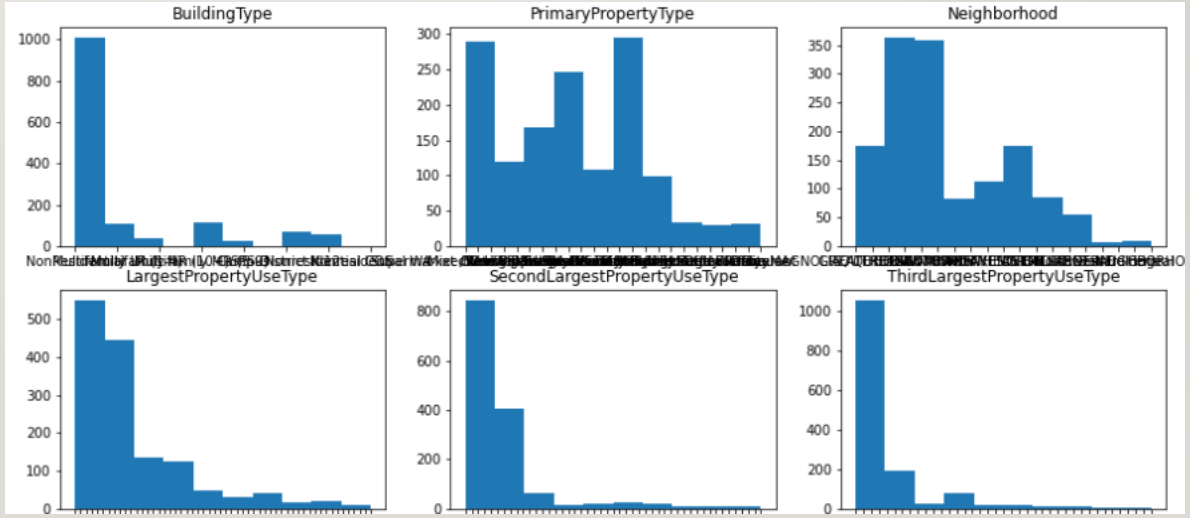
```
1 numeric=X_train.dtypes[X_train.dtypes!='object'].index
2 numeric
```

```
Index(['CouncilDistrictCode', 'Latitude', 'Longitude', 'YearBuilt',
       'NumberofFloors', 'PropertyGFATotal', 'PropertyGFAParking',
       'PropertyGFABuilding(s)', 'LargestPropertyUseTypeGFA',
       'SecondLargestPropertyUseTypeGFA', 'ThirdLargestPropertyUseTypeGFA',
       'SiteEUI(kBtu/sf)', 'SiteEUIWN(kBtu/sf)', 'SourceEUI(kBtu/sf)',
       'SourceEUIWN(kBtu/sf)', 'SiteEnergyUseWN(kBtu)', 'Electricity(kWh)',
       'Electricity(kBtu)', 'NaturalGas(therms)', 'NaturalGas(kBtu)',
       'GHGEmissionsIntensity'],
      dtype='object')
```

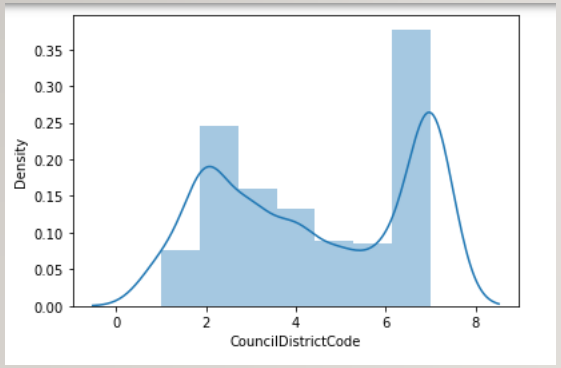
5- ANALYSE EXPLORATOIRE - VARIABLES CATÉGORIELLES

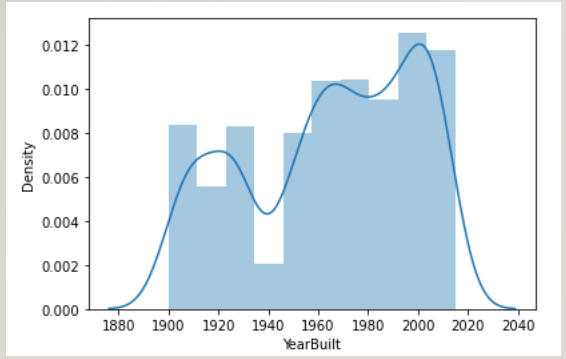
```
1 categ=X_train.dtypes[df.dtypes=='object'].index
2 categ
```

```
Index(['BuildingType', 'PrimaryPropertyType', 'Neighborhood',
       'LargestPropertyUseType', 'SecondLargestPropertyUseType',
       'ThirdLargestPropertyUseType'],
      dtype='object')
```

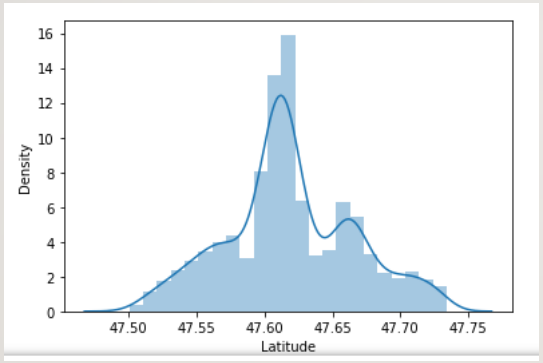


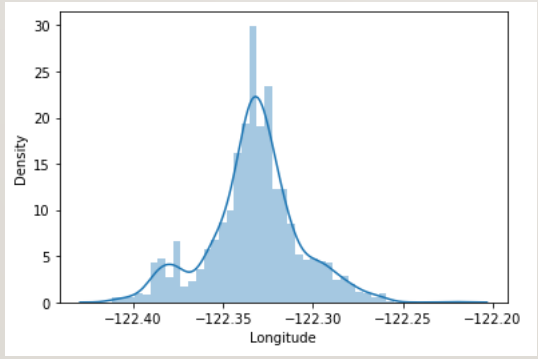
5- ANALYSE EXPLORATOIRE - VARIABLES NUMÉRIQUES



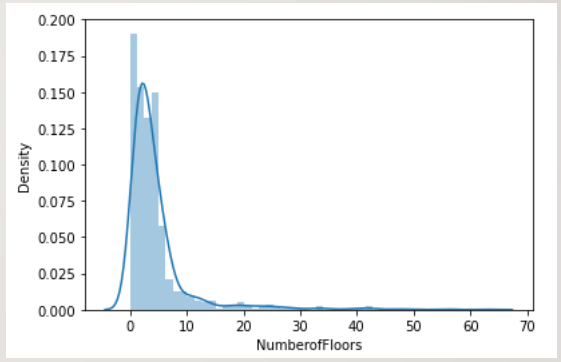


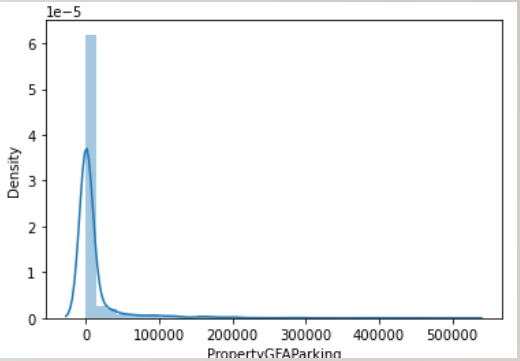
Variables Discrètes

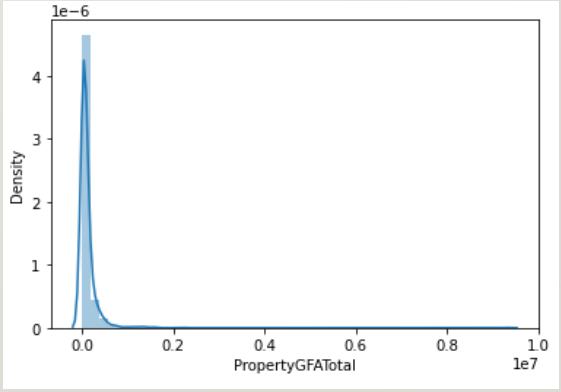


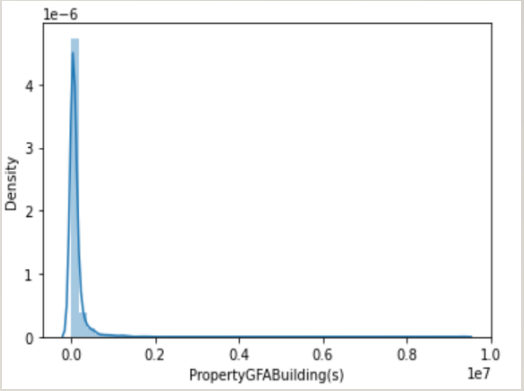


Variables Normales



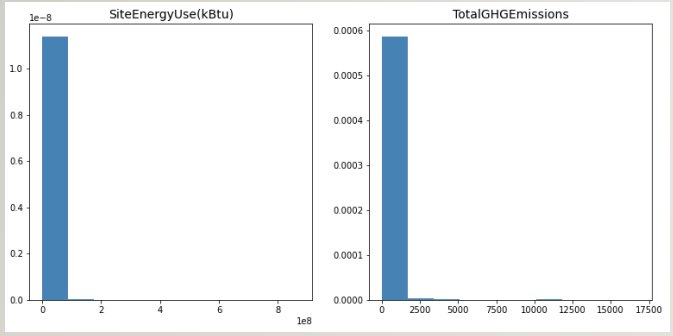






Variables asymétriques (skewed)

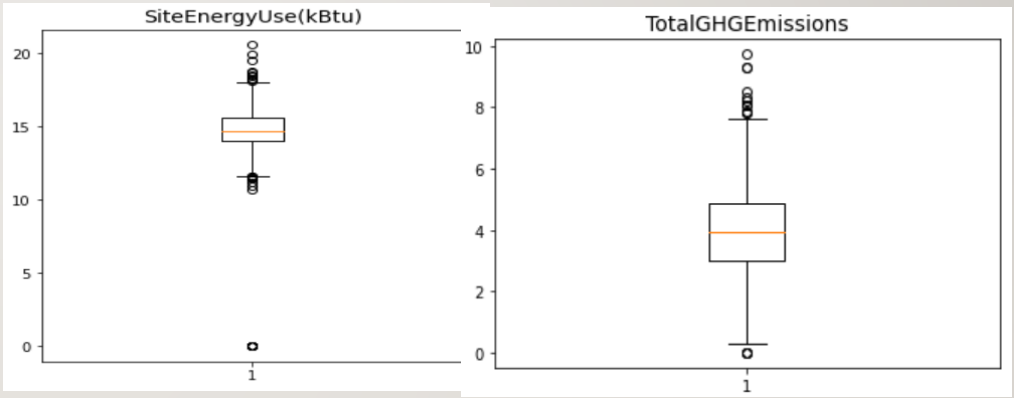
5- ANALYSE EXPLORATOIRE - VARIABLES CIBLES



Distribution asymétriques (skewed)

	SiteEnergyUse(kBtu)	TotalGHGEmissions
SiteEnergyUse(kBtu)	1.000000	0.860669
TotalGHGEmissions	0.860669	1.000000

Fortement corrélées



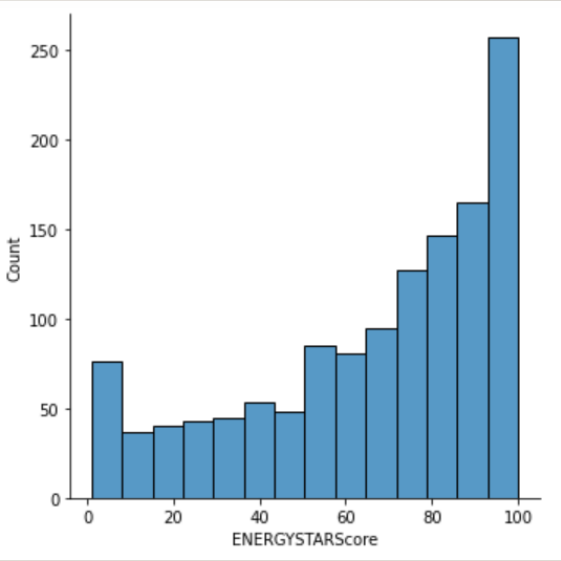
SiteEnergyUse(Kbtu) présente des valeurs aberrantes dans la partie inferieure (SiteEnergieUse(kBtu)==0)

5- ANALYSE EXPLORATOIRE - VARIABLE ENERGYSTARSCORE

- Variable « ENERGYSTARScore »
  - 36% de valeurs manquantes
  - La rectification de ces valeurs vont présenter biais sur l'importance de cette variable dans la prédiction de l'émission et de la consommation.
- Décision:
  - Enlever cette variable du dataset d'étude
  - Prendre un autre dataset contenant cette variable (sans les NaN) pour l'étude de la pertinence de cette variable pour la prédiction de l'émission de CO2

```
df.isna().mean().sort_values(ascending=False).head(2)
```

```
ENERGYSTARScore     0.361262
OSEBuildingID       0.000000
```



4- ANALYSE EXPLORATOIRE - CORRÉLATION AVEC VARIABLES CIBLES

```
PropertyGFATotal                0.798337
PropertyGFABuilding(s)          0.813925
LargestPropertyUseTypeGFA       0.838447
SiteEnergyUse(kBtu)             1.000000
TotalGHGEmissions               0.860669
Name: SiteEnergyUse(kBtu), dtype: float64
```

Variables corrélées avec la consommation d'énergie

```
SiteEnergyUse(kBtu)         0.860669
SiteEnergyUseWN(kBtu)       0.857128
TotalGHGEmissions           1.000000
Name: TotalGHGEmissions, dtype: float64
```

Variables corrélées avec l'émission de CO2

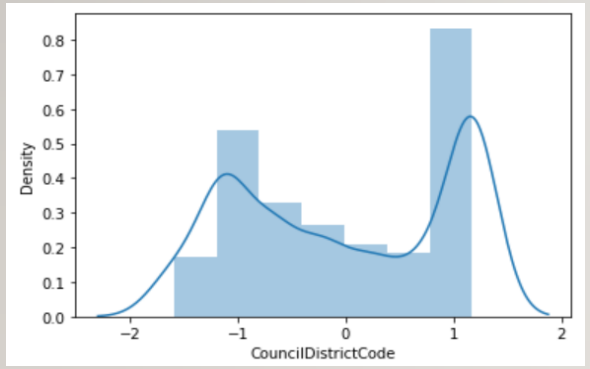
PRÉPARATION DES VARIABLES POUR LA RÉGRESSION

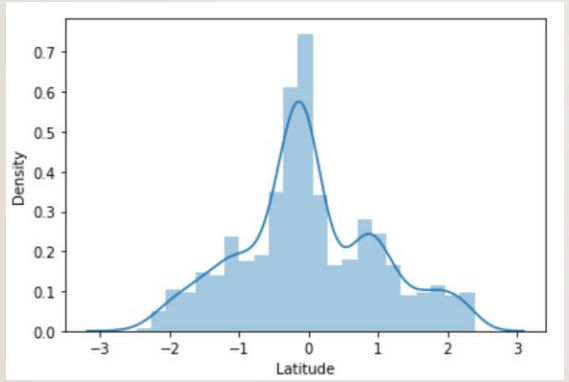
- Transformation des variables
- Méthodes de régressions
- Recherches d'hyperparametres

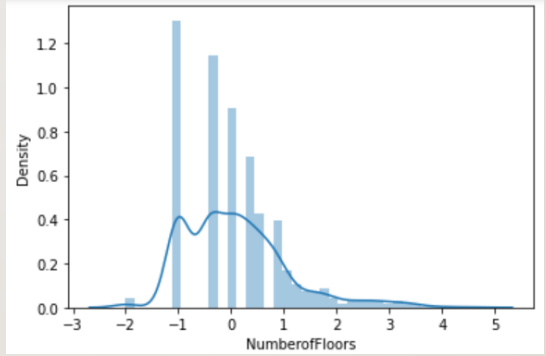
1- TRANSFORMATION DES VARIABLES

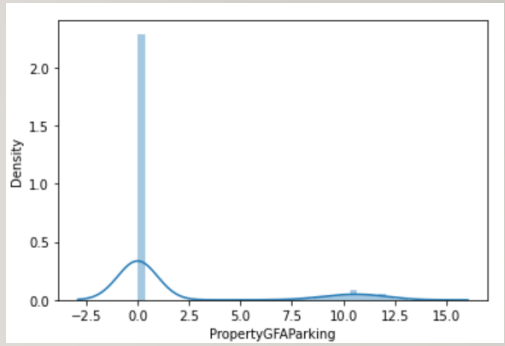
- Pour une meilleur performance des méthodes de régression, les variables doivent se rapprocher d'une distribution normale.
- Pour appliquer les transformations : **ColumnTransformer**
- Transformations sur les variables selon leur type et leur distribution
  - **Normale**, **Discrete** **StandardScaler**
  - **Skewed** : Pipeline : **FunctionTransformer**(np.log1p) puis **RobustScaler**
  - **Catégorielle** : **BackwardDifferenceEncoder**
  - **Booleenne** Pas de transformation (**remainder='passthrough'** dans ColumnTransformer)

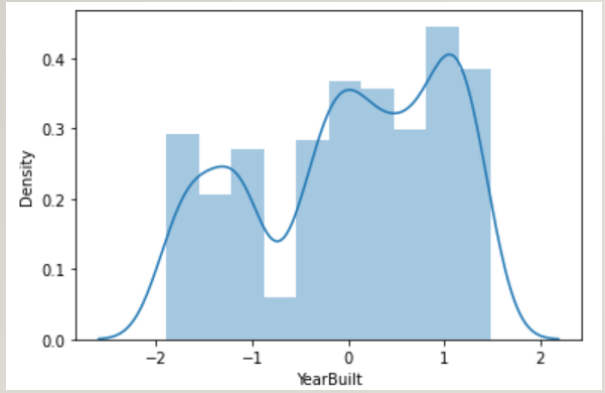
1- TRANSFORMATION DES VARIABLES – VARIABLES NUMERIQUES

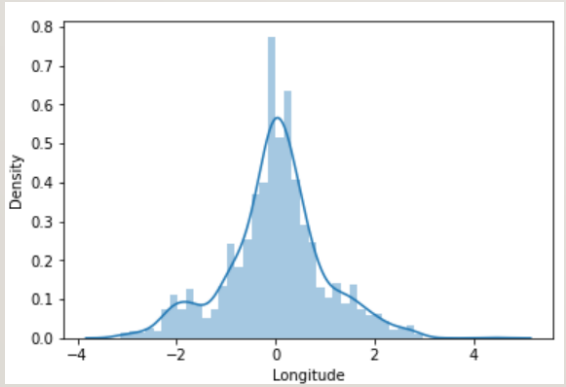


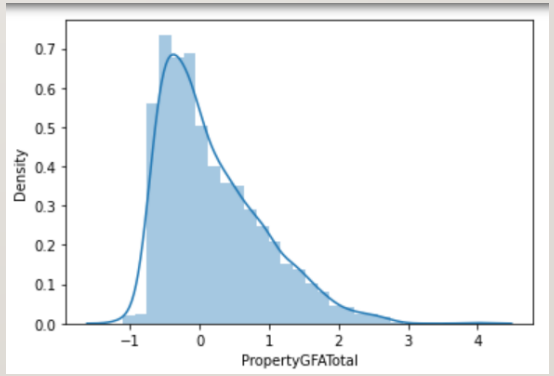


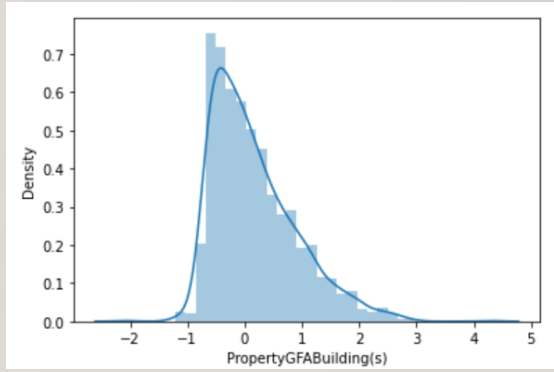












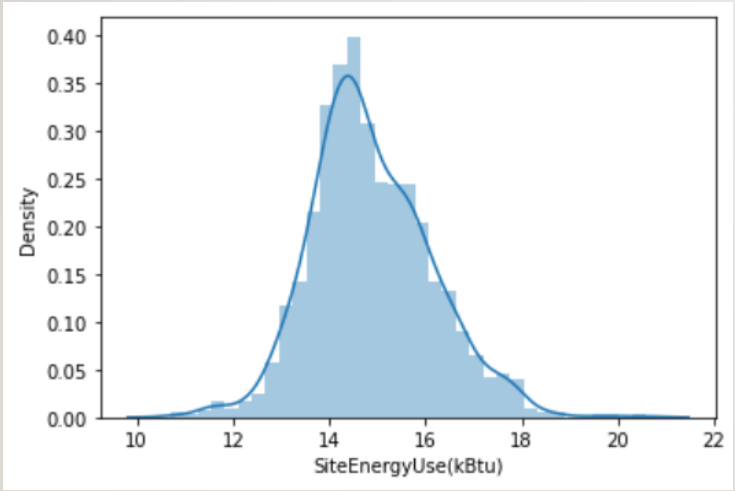
Variables Discrètes

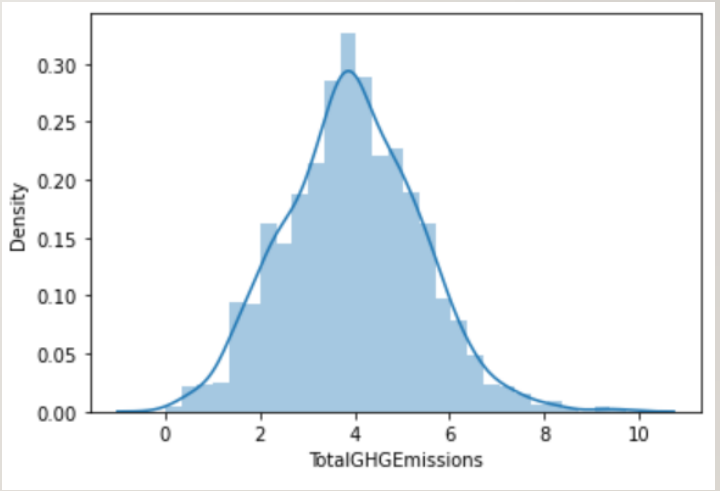
Variables Normales

Variables asymétriques (skewed)

1- TRANSFORMATION DES VARIABLES – VARIABLES CIBLES

- **Variables cibles** Pipeline : **FunctionTransformer**(np.log1p) puis **StandardScaler**





Les variables ont une allure plus normale après transformation et normalisation

2- MÉTHODES DE RÉGRESSION

- **Dummy regression**
- **Méthodes linéaires**
  - Linear regression
  - Regression Ridge
  - Regression Lasso
- **Méthodes ensemblistes**
  - Decision Tree
  - Random Forest
- **Descente de gradient**:
  - XGBoost

3- RECHERCHE DES HYPERPARAMÈTRES

- Pour les méthodes Ridge, Lasso et Decision Tree
  - **GridSearchCV** : Recherche exhaustive de toutes les combinaisons possibles des paramètres donnes et effectuant une Cross-validation pour trouver le meilleur score
- Pour les méthodes RandomForest et XGBoost
  - **RandomizedSearchCV**: Recherche aléatoire de combinaisons de paramètres en se basant sur les scores précédants et en effectuant une Cross-validation pour trouver le meilleur score.
  - Choisit pour le gain de temps de recherche comme il y a un grand nombre de paramètres a essayer

3- RECHERCHE DES HYPERPARAMÈTRES

Méthode	Hyperparamètres	Méthode de recherche
Ridge, Lasso	**alpha** : np.logspace(-5, 5, 300) **fit_intercept**:[True,False]	GridSearchCV
Decision Tree	**splitter**:["best","random"], **max_depth** :[1,3,5,7,9,11,12], **min_samples_leaf**:[1,2,3,4,5,6,7,8,9,10], **min_weight_fraction_leaf**:[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9], **max_features**:["auto","log2","sqrt",None], **max_leaf_nodes**:[None,10,20,30,40,50,60,70,80,90] }	GridSearchCV
Random Forest	**n_estimators** = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)] **max_features** = ['auto', 'sqrt'] **max_depth** = [int(x) for x in np.linspace(10, 110, num = 11)] max_depth.append(None) **min_samples_split** = [2, 5, 10] **min_samples_leaf** = [1, 2, 4] **bootstrap** = [True, False]	RandomizedSearchCV
XGboost	**"learning_rate": [0.05,0.10,0.15,0.20,0.25,0.30],** **"max_depth" : [3, 4, 5, 6, 8, 10, 12, 15],** **"colsample_bytree" : [0.3, 0.4, 0.5 , 0.7],** **"n_estimators": [200, 500, 1000, 1500, 2000]**	RandomizedSearchCV

4- CALCUL D'ERREUR ET SCORE

- Le test set a été utiliser pour la validation des résultats
  - Calcul des prédictions pour chaque méthode
  - Calcul des erreurs
    - RMSE racine de l'erreur quadratique moyenne
    - R2 le coefficient de corrélation

RÉSULTATS

- Prédiction de l'émission de CO2
  - Sans ENERGYSTARScore
  - Avec ENERGYSTARScore
  - Pertinence de ENERGYSTARScore
- Prédiction de la consommation de l'énergie

1- EMISSION CO2 - HYPERPARAMÈTRES

Modele	Ridge	Lasso	Decision Tree	Random Forest	XGBoost
Hyper-parameters	**alpha**: 4.1565125582259, **fit_intercept**: True	**alpha**: 0.00101552112763, **fit_intercept**: True	**max_depth**: 5, **max_features**: 'auto', **max_leaf_nodes**: None, **min_samples_leaf**: 1, **min_weight_fraction_leaf**: 0.1, **splitter**: 'best'	**n_estimators**: 800, **min_samples_split**: 5, **min_samples_leaf**: 2, **max_features**: 'auto', **max_depth**: None, **bootstrap**: True	**learning_rate**: 0.1, **max_depth**: 3, **colsample_bytree**: 0.3 **n_estimators**: 200

1- EMISSION CO2 - RÉSULTATS DE PERFORMANCE

- Temps de recherche des paramètres et Score R2 de la validation croisée sur le **Train set**

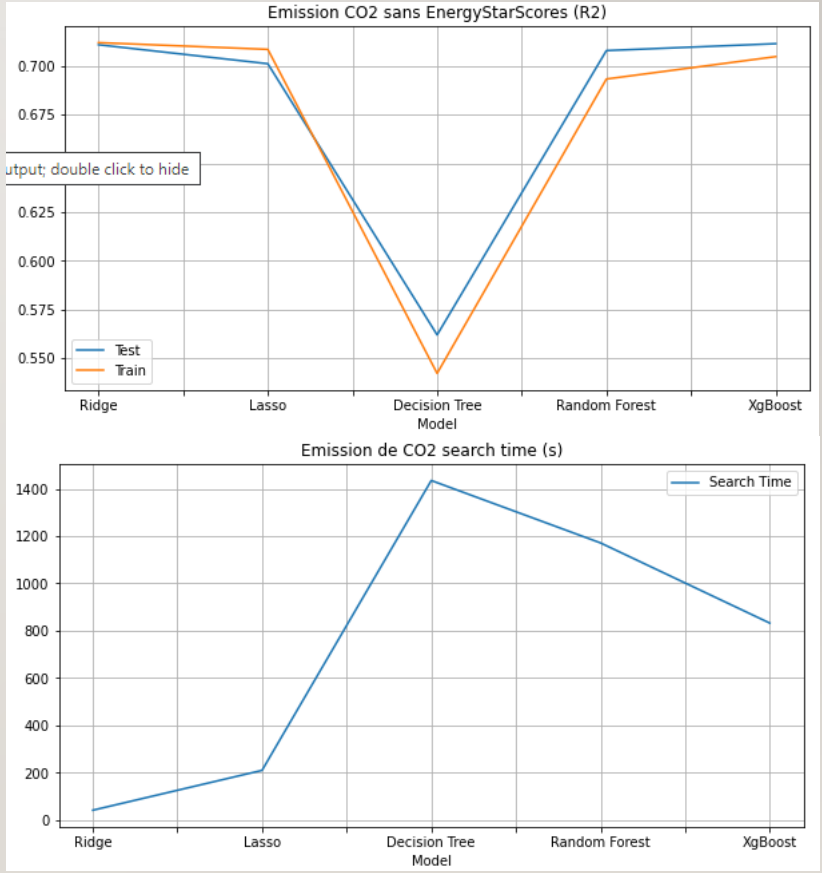
Model	Ridge	Lasso	Decision Tree	Random Forest	XgBoost
Search Time	41.837421	210.189438	1434.754854	1171.13446	832.490669
Best Score	0.71193	0.708474	0.54229	0.693262	0.704748

- Temps d'exécution de la prédiction et Scores R2 et RMSE sur le **Test set**

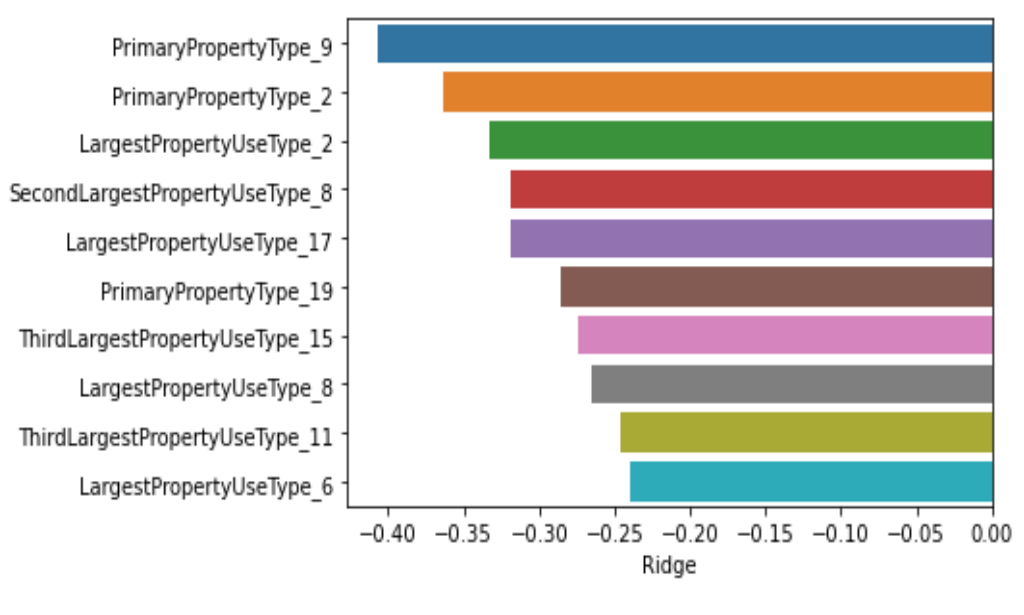
Model	Dummy	Linear Regression	Ridge	Lasso	Decision Tree	Random Forest	XgBoost
Time	0.000575	0.039536	0.037938	0.467231	0.0179	18.647228	0.865007
RMSE	1.013578	2885008112.90101	0.544909	0.554016	0.670638	0.547728	0.5444
R2	-0.000435	-8105297949664456704.0	0.71085	0.701105	0.562023	0.707851	0.71139

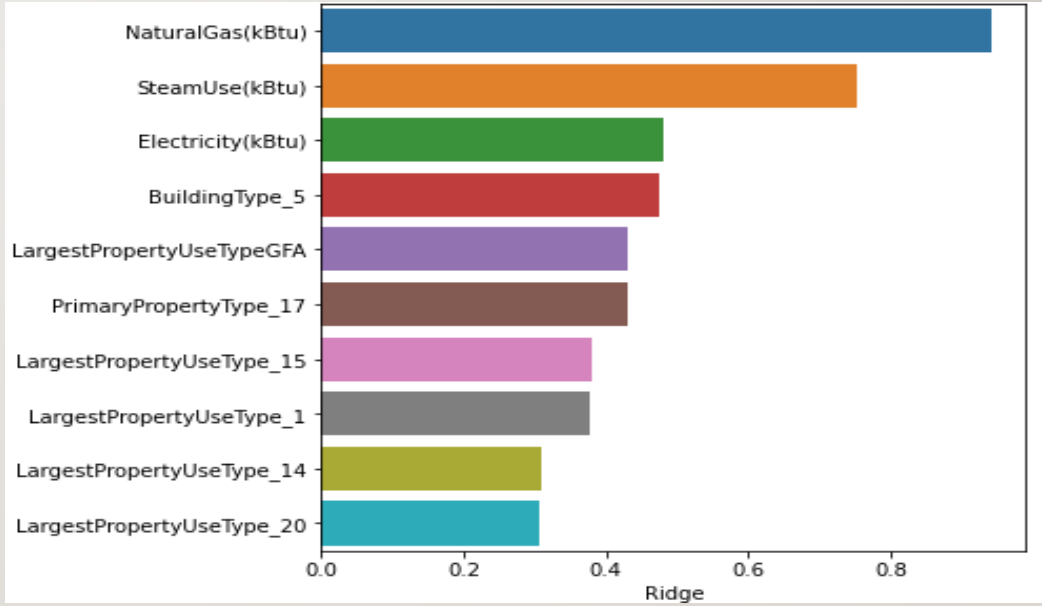
1- EMISSION CO2 - RÉSULTATS DE PERFORMANCE

- Les méthodes de régression linéaire avec régularisation (Ridge et Lasso) sont bien efficaces en temps de recherche pour les hyperparamètres et en temps de calcul de prédiction. Leur score est très acceptable
- L'arbre de décision réduit énormément le nombre de variables pour la prédiction mais n'arrive pas a faire de bonnes prédictions
- Le Random Forest et XgBoost prennent beaucoup plus de temps de recherche des hyperparamètres et donne un score très proche du score des régressions linéaires régularisées



1- EMISSION CO2 - FEATURE IMPORTANCES - RIDGE

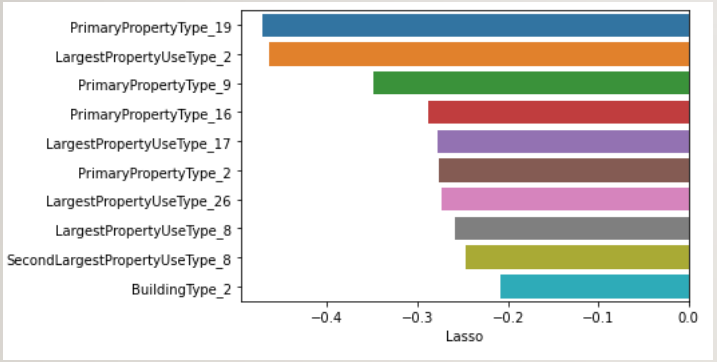


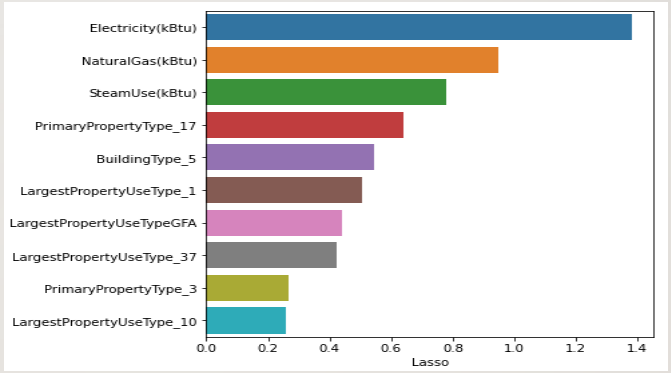


PrimaryPropertyType
LargestPropertyUseType
SecondLargestPropertyUseType
ThirdLargestPropertyUseType

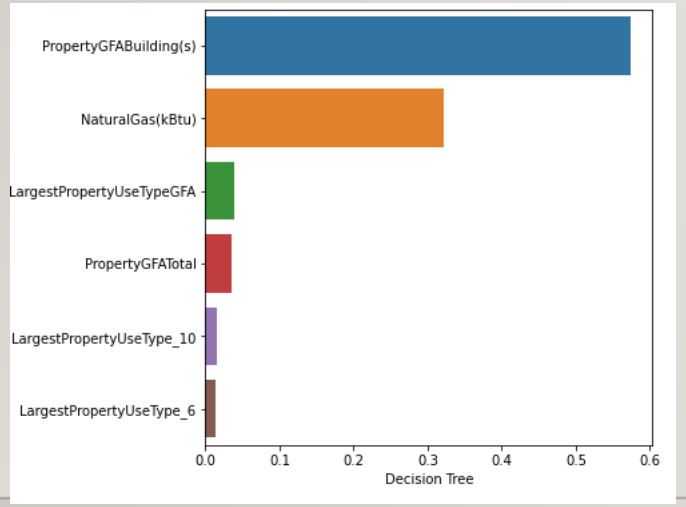
NaturalGas
SteamUse
Electricity
BuildingType
LargestPropertyUseTypeGFA

1- EMISSION CO2 - FEATURE IMPORTANCES

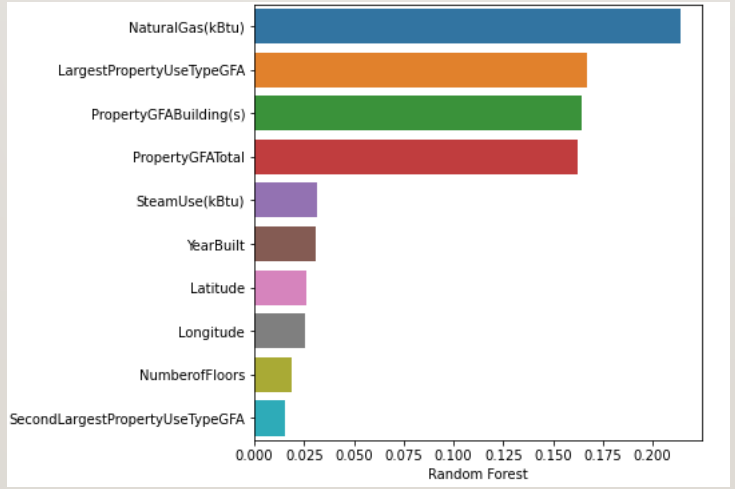




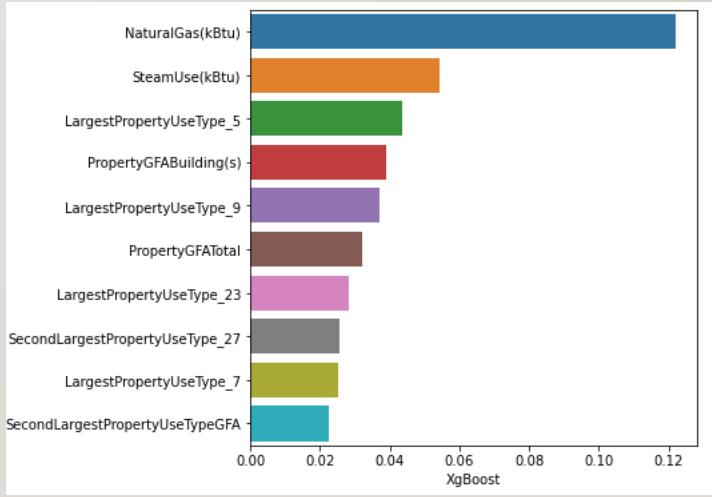
Régression Lasso



Decision Tree Regression



Random Forest Regression



XGBoost Regression

2 - EMISSION CO2 AVEC ENERGYSTARSCORE - HYPERPARAMETERS

Modele	Ridge	Lasso	Decision Tree	Random Forest	XGBoost
Hyper-parameters	**alpha**: 4.768611697714 47 **fit_intercept**: True	**alpha**: 0.001851648417 91, **fit_intercept**: True	**max_depth**: 7, **max_features**: None, **max_leaf_nodes**: 50, **min_samples_leaf**: 3, **min_weight_fraction_leaf**: 0.1, **splitter**: 'random'	**n_estimators**: 1400, **min_samples_split**: 2, **min_samples_leaf**: 1, **max_features**: 'auto', **max_depth**: 100, **bootstrap**: True	**learning_rate**: 0.1 **max_depth**: 3, **colsample_bytree**: 0.5 **n_estimators**: 500

2- EMISSION CO2 AVEC ENERGYSTARSCORE – RÉSULTATS DE PERFORMANCE

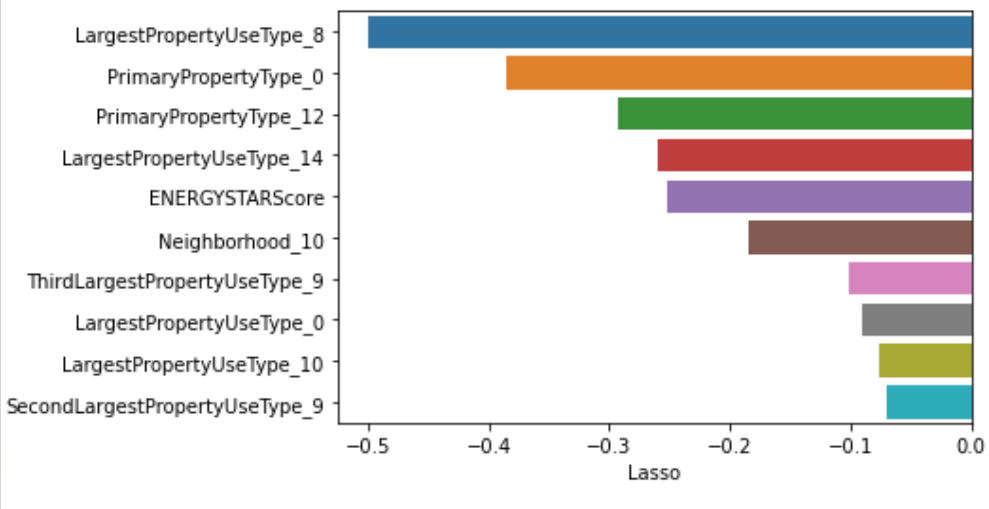
- Temps de recherche des paramètres et Score R2 de la validation croisée sur le **Train set**

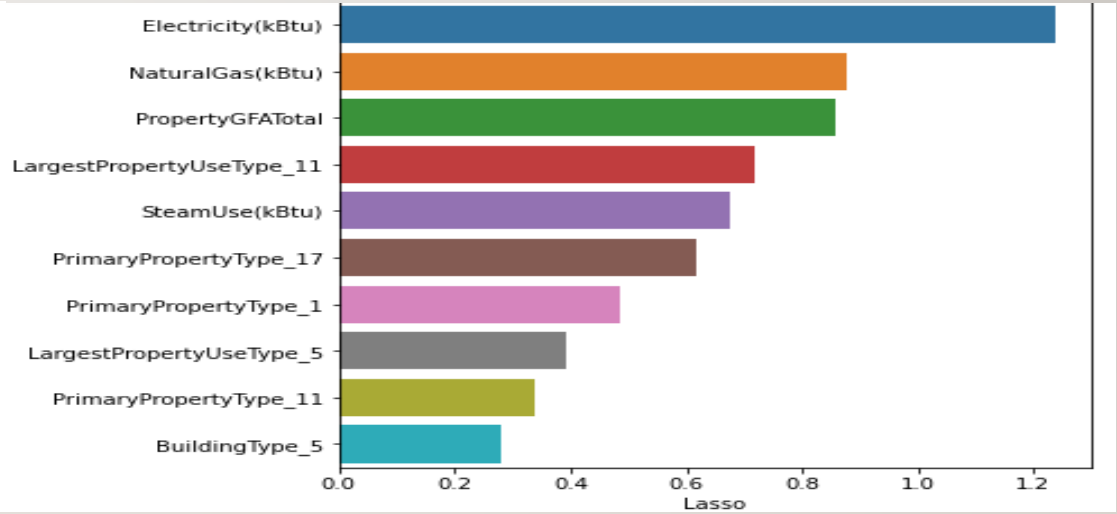
Model	Ridge	Lasso	Decision Tree	Random Forest	XgBoost
Search Time	39.584624	119.970718	941.570203	627.21644	343.217671
Best Score	0.768639	0.769493	0.556161	0.726778	0.764352

- Temps d'exécution de la prédiction et Scores R2 et RMSE sur le **Test set**

Model	Dummy	Linear Regression	Ridge	Lasso	Decision Tree	Random Forest	XgBoost
Time	0.004153	0.148298	0.049332	0.16051	0.020169	16.924327	1.491865
RMSE	1.005927	0.466576	0.467917	0.464954	0.690251	0.465441	0.458912
R2	-0.010729	0.782556	0.781305	0.784065	0.5241	0.783613	0.789641

2- EMISSION CO2 AVEC ENERGYSTARSCORE - FEATURE IMPORTANCES - LASSO

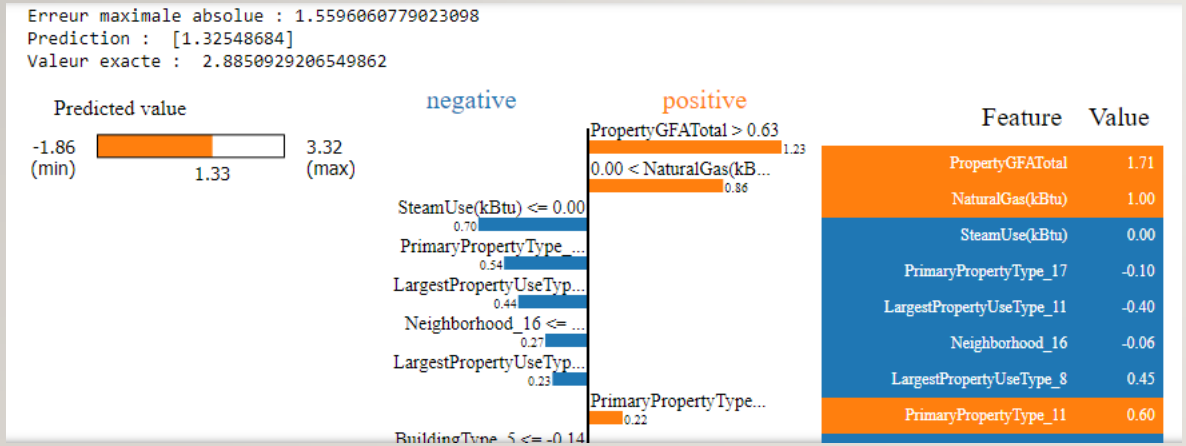




LargestPropertyUseType
PrimaryPropertyType
ENERGYSTARScore
Neighborhood
ThirdLargestPropertyUseType
SecondLargestPropertyUseType

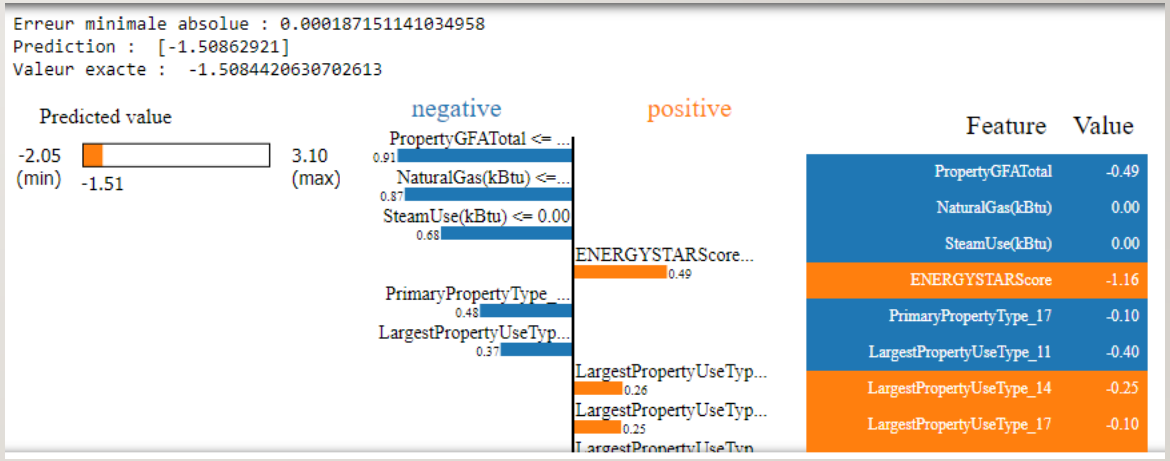
Electricity
NaturalGas
PropertyGFATotal
SteamUse
BuildingType

2- EMISSION CO2 AVEC ENERGYSTARSCORE – ETUDE LOCALE DES FEATURE IMPORTANCES

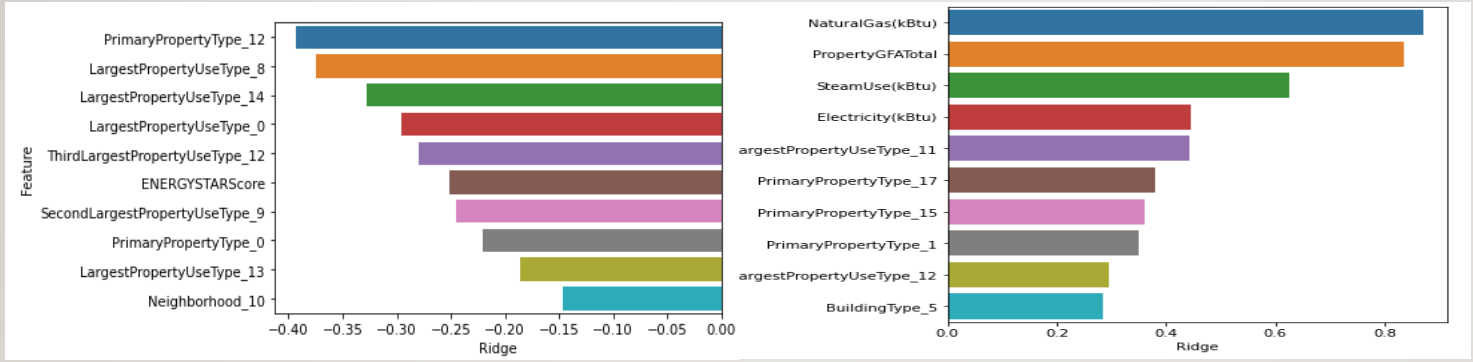


Etude locale sur les feature importances de l'instance avec erreur maximale de prédiction par Lasso

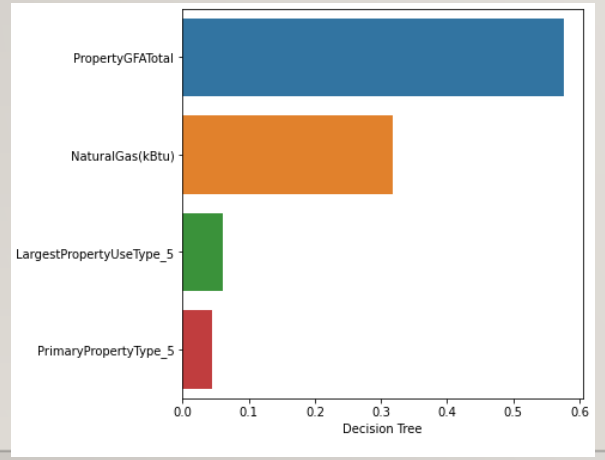
Etude locale sur les feature importances de l'instance avec erreur minimale de prédiction par régression Lasso



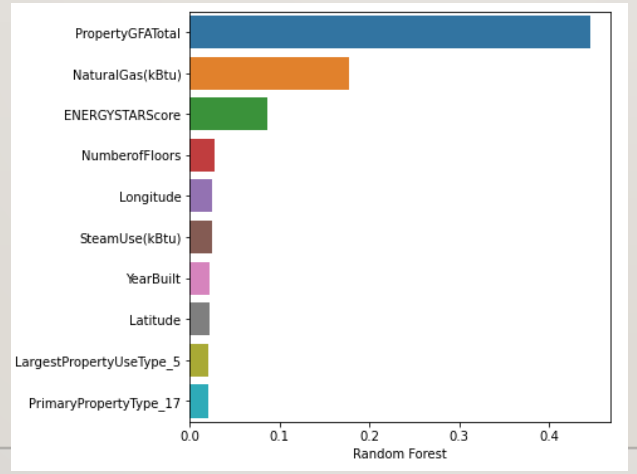
2- EMISSION CO2 AVEC ENERGYSTARSCORE - FEATURE IMPORTANCES



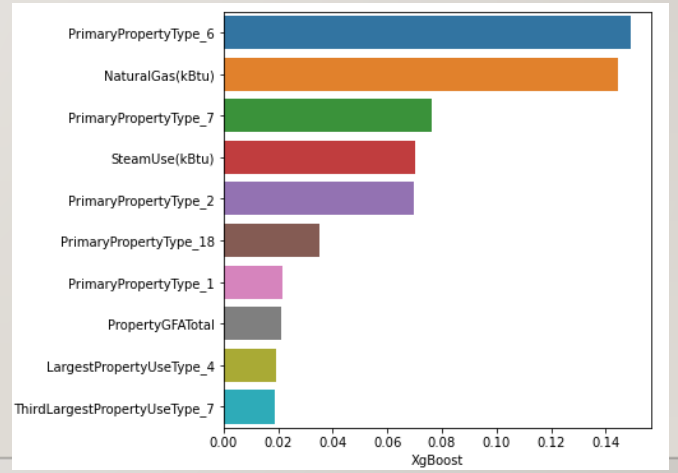
Régression Ridge



Decision Tree Regression



Random Forest Regression

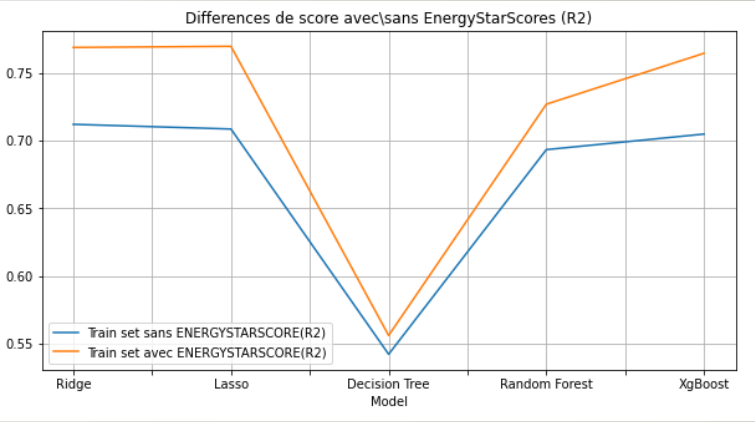


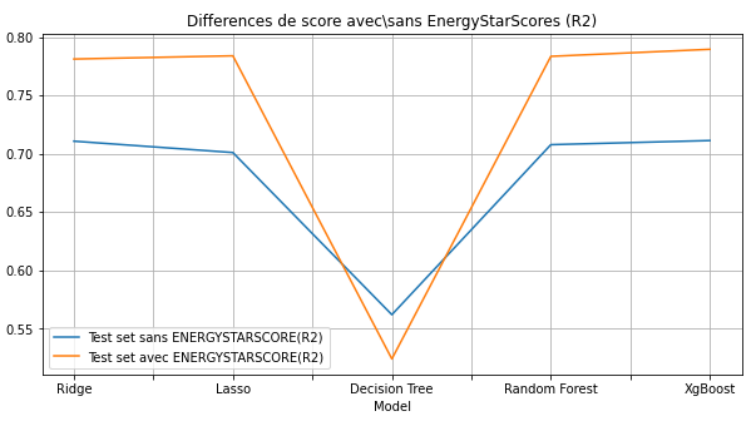
XGBoost Regression

2- EMISSION CO2 AVEC ENERGYSTARSCORE – PERTINENCE DE LA VARIABLE

- La variable ENERGYSTARScore Contribue a une amélioration des performances sur toutes les méthodes de prédiction (Sauf Decision Tree)

	Model	Train difference percentage	Test difference percentage
	Ridge	7.965625	9.911271
	Lasso	8.612765	11.83284
	Decision Tree	2.557991	-6.747607
	Random Forest	4.834648	10.703187
	XgBoost	8.45749	10.999821





3- CONSOMMATION D'ÉNERGIE - HYPERPARAMETERS

Modele	Ridge	Lasso	Decision Tree	Random Forest	XGBoost
Hyper-parameters	**alpha**: 3.37006432927193 **fit_intercept**: False	**alpha**: 0.00054844165761, **fit_intercept**: False	**max_depth**: 5, **max_features**: auto, **max_leaf_nodes**: None, **min_samples_leaf**: 1, **min_weight_fraction_leaf**: 0.1, **splitter**: 'best'	**n_estimators**: 1400, **min_samples_split**: 2, **min_samples_leaf**: 2, **max_features**: auto', **max_depth**: 70, **bootstrap**: True	**learning_rate**: 0.05, **max_depth**: 3, colsample_bytree': 0.7 **n_estimators**: 200

3- CONSOMMATION D'ÉNERGIE – RÉSULTATS DE PERFORMANCE

- Temps de recherche des paramètres et Score R2 de la validation croisée sur le **Train set**

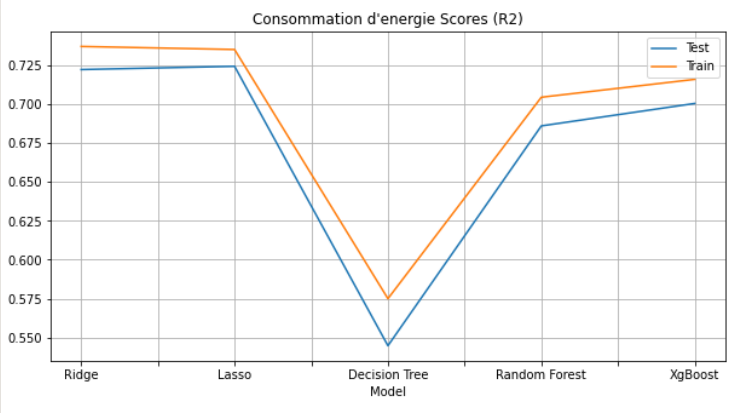
Model	Ridge	Lasso	Decision Tree	Random Forest	XgBoost
Search Time	25.160186	119.05885	2717.212877	1404.059603	665.040201
Best Score	0.73719	0.735225	0.574946	0.704509	0.716022

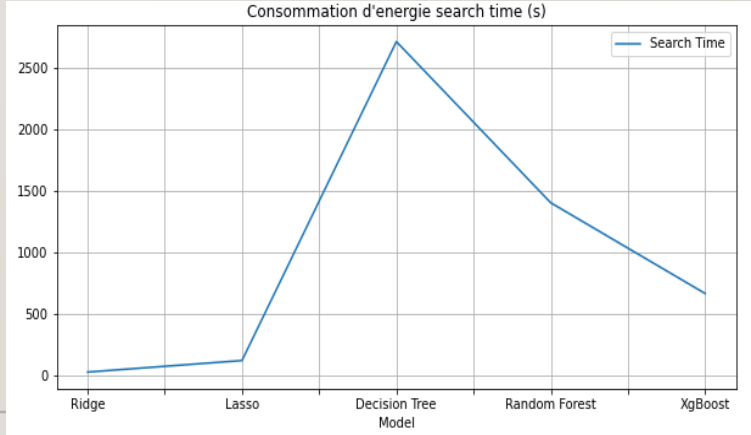
- Temps d'exécution de la prédiction et Scores R2 et RMSE sur le **Test set**

Model	Dummy	Linear Regression	Ridge	Lasso	Decision Tree	Random Forest	XgBoost
Time	0.000781	0.027789	0.04807	0.314054	0.032121	46.877624	1.409988
RMSE	1.04711	18206953921.124996	0.551431	0.54935	0.706141	0.586327	0.572685
R2	-0.001218	-3027042162170892 12416.0	0.722331	0.724424	0.54467	0.686077	0.700514

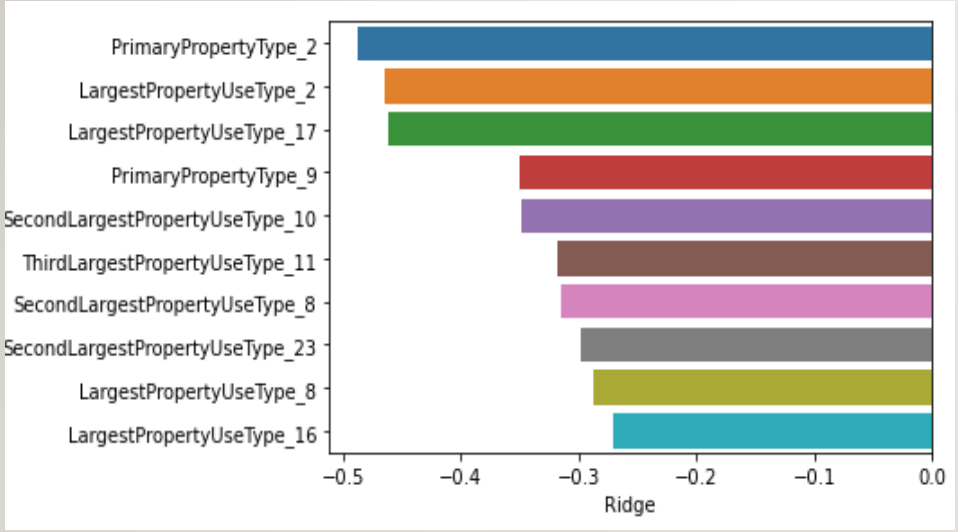
3- CONSOMMATION D'ÉNERGIE – RÉSULTATS DE PERFORMANCE

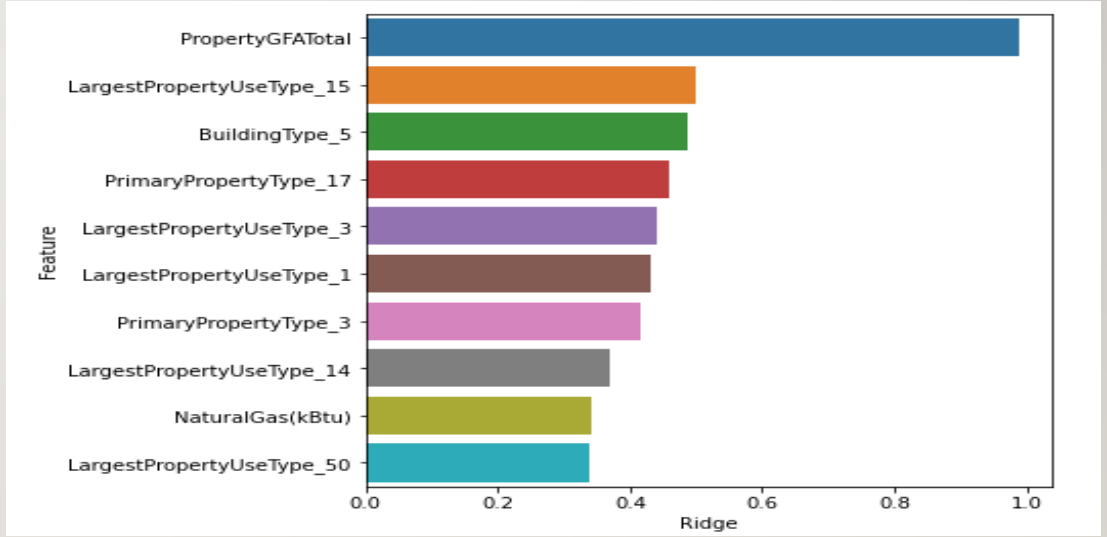
- Les méthodes linéaires régularisées sont les meilleures en fonction de temps de calcul et de performance
- La performance de prédiction de la consommation d'énergie est similaire a celle de l'emission de CO2.
- Cela est du au fait qu'on a omis les informations sur la consommation d'énergie (électricité, gaz et vapeur) des bâtiments





3- CONSOMMATION D'ÉNERGIE –FEATURE IMPORTANCES - RIDGE

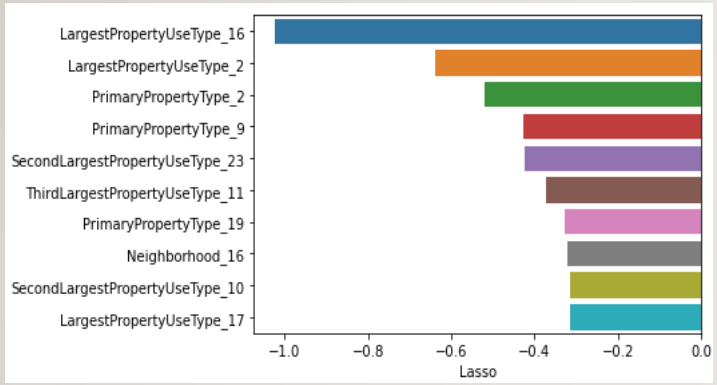


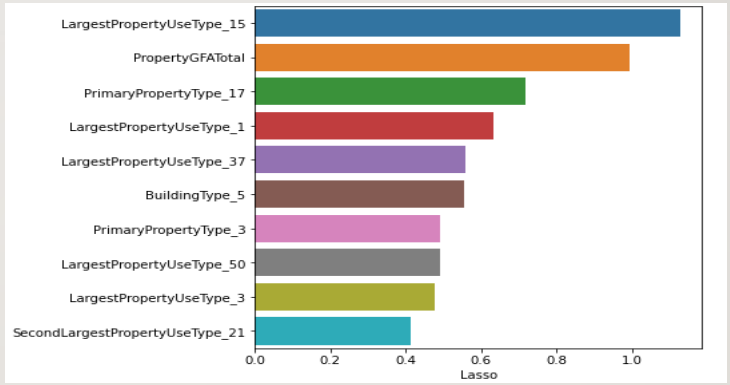


PrimaryPropertyType
LargestPropertyUseType
SecondLargestPropertyUseType
ThirdLargestPropertyUseType

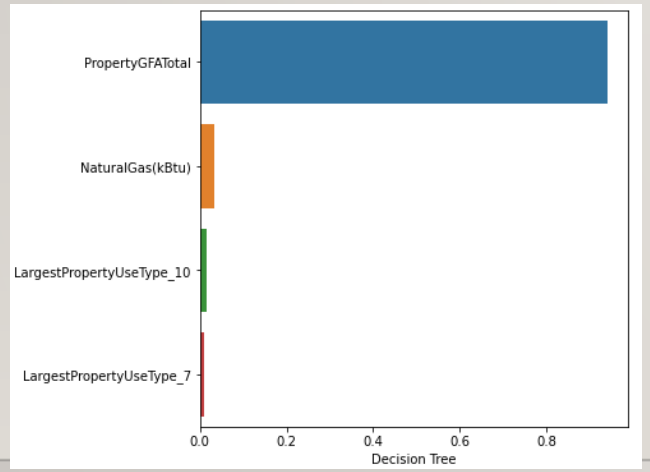
PropertyGFATotal
BuildingType
NaturalGas

3- CONSOMMATION D'ÉNERGIE –FEATURE IMPORTANCES

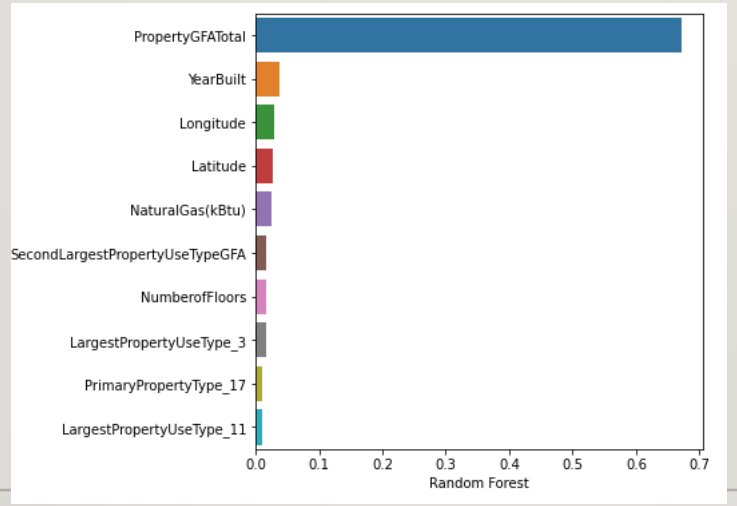




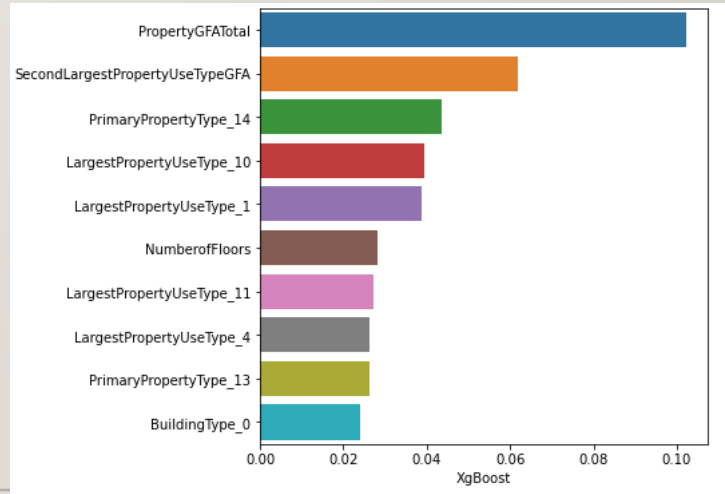
Régression Lasso



Decision Tree Regression



Random Forest Regression



XGBoost Regression

Merci!