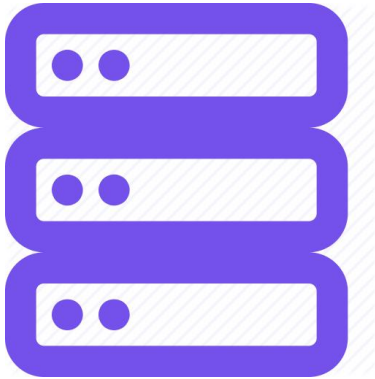
Formation DS-IML PROJET 3 : SEATTLE ENERGY BENCHMARKING

City of Seattle



Cadastre

Conso. énergétique

Rejets CO2

MISSION

- Modélisation conso. énergétique
- Modélisation rejets GHG
- Discussion *Energy Star Score*

SOMMAIRE

1 - RÉUNION DES DATASETS

DATA = 2015 + 2016

a) Renommer des colonnes
b) Ventiler certaines données (adresses…)

n = 6624

2 - GESTION DES DOUBLONS

seulement 3340 obs. uniques

PB de COHÉRENCE des données

1er choix compliqué permettant de garder une faible quantité de données.

2nd choix plus simple, de bon sens consistant à garder les data récentes

n = 3340

Très vite...

- Corrections
- Éliminations var. non pertinentes : méta, géo, etc.
  - sauf “Neighborhood”
- Création de “DecadeBuilt” à partir de “YearBuilt”

Groupe var. quantitatives

Points significatifs

A)

Distribution commune “étalée à droite”

Transformation LOG

Création var. binaire “Outlier”

B)

Transformations sur variables “GFA”

GFATotal = GFABuilding + GFAParking

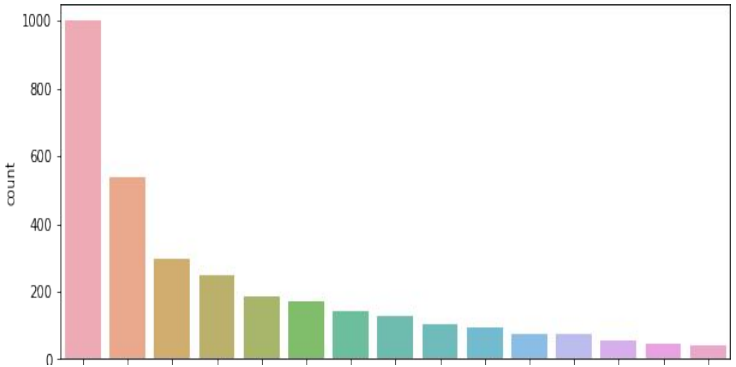
BGFARate = GFABuilding / GFA Total

L_PUTGFA = LargestPropertyUseTypeGFA / PropertyGFATotal **x 3**

Groupe var. qualitatives

Points significatifs

VARIABLE	NB de CATÉGORIES
BuildingType	7
PrimaryPropertyType	31
LargestPropertyUseType	57
SecondLargestPropertyUseType	51
ThirdLargestPropertyUseType	44



Travail nécessaire sur la “profondeur” de ces variables

Cas de ListOfAllPropertyUse et de ses 475 catégories **inutilisable !**

→ **Imputations sur val. manquantes**

→ **Création var. quantitative NumberOfPropertyUse**

Groupe var. “cibles”

Points significatifs

15 var. quantitatives relatives aux mesures de conso & d’émissions GHG

Conso. & émis. : brutes | norm. météo | par unité de surface

Var. EnergyStarScore ajoutée aux var.indépendantes

conso. particulières : Steam | Natural Gas | Other

→ Création 3 var. binaires Steam / NaturalGas / Other

Smart Data avant modélisation

```
Int64Index: 3303 entries, 0 to 3339
Data columns (total 20 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   NumberOfPropertyUse           3303 non-null   int64
 1   NumberofBuildings             3303 non-null   float64
 2   NumberofFloors                3303 non-null   float64
 3   PropertyGFATotal              3303 non-null   int64
 4   PropertyBuildingGFARate       3303 non-null   float64
 5   L_PUTGFA                      3303 non-null   float64
 6   S_PUTGFA                      3303 non-null   float64
 7   T_PUTGFA                      3303 non-null   float64
 8   ENERGYSTARScore               2498 non-null   float64
 9   Neighborhood                  3303 non-null   category
 10  BuildingType                  3303 non-null   category
 11  PrimaryPropertyType           3303 non-null   category
 12  LargestPropertyUseType        3303 non-null   category
 13  SecondLargestPropertyUseType  3303 non-null   category
 14  ThirdLargestPropertyUseType   3303 non-null   category
 15  DecadeBuilt                   3303 non-null   category
 16  Steam                         3303 non-null   category
 17  NaturalGas                    3303 non-null   category
 18  Other                         3303 non-null   category
 19  Outlier                       3303 non-null   category
```

3303 observations

20 variables

9 numériques

11 catégorielles

dont

4 binaires

City of Seattle

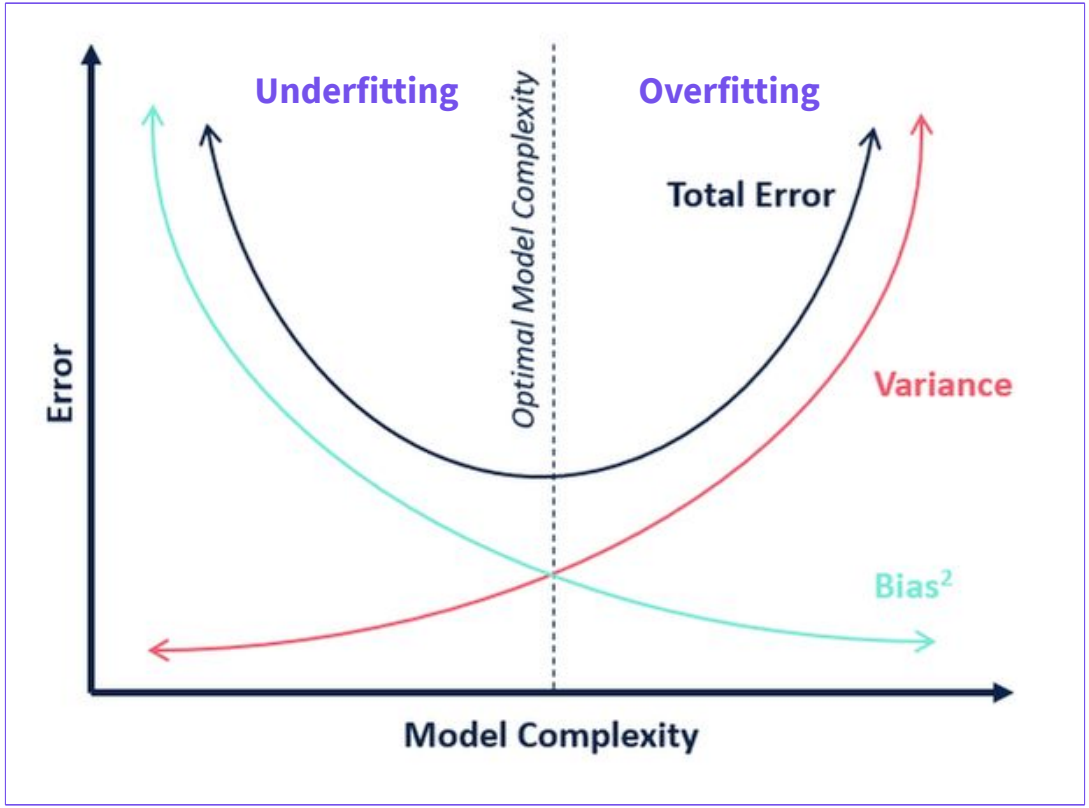
SOMMAIRE

Démarche suivie	machine learning supervisé
	régression
Pool d'estimateurs	lequel va matcher avec les données ?
Encodage	test & choix entre 3 type d'encodages

Sélection des meilleurs modèles

Var. Catégorielles	optimisation de leur profondeur
Feature Selection	élimination de variables pénalisantes
Hyper-Paramétrage	optimisation des paramètres des modèles

*Sélection **DU** meilleur modèle*


Underfitting
Optimal Model Complexity
Overfitting
Total Error
Error
Variance
Bias²
Model Complexity

Métriques utilisées :

3 indicateurs différents de performance

MAE

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|$$

métrique “basique”

RMSLE

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(\hat{y}_i + 1) - log(y_i + 1))^2}$$

métrique “relative”

Score R2

$$R^2 = 1 - \frac{\sum_{1=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{1=1}^{N}(y_i - \bar{y})^2}$$

métrique “empirique”

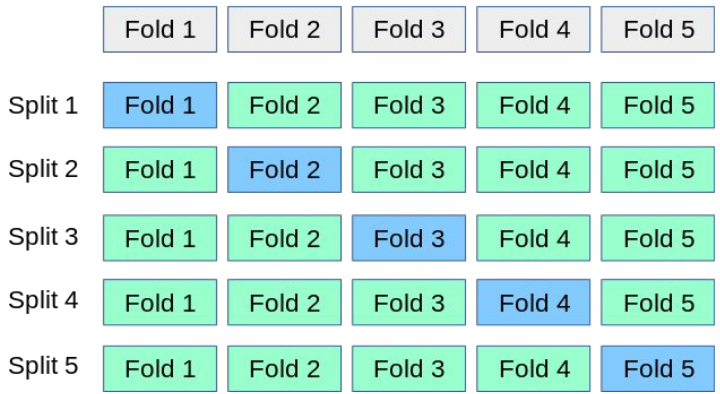
Pre processing des données numériques

Transformation logarithmique

Standardisation

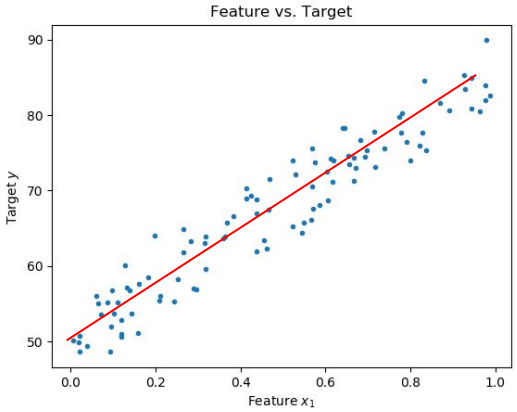
Échantillonnage Train/Test

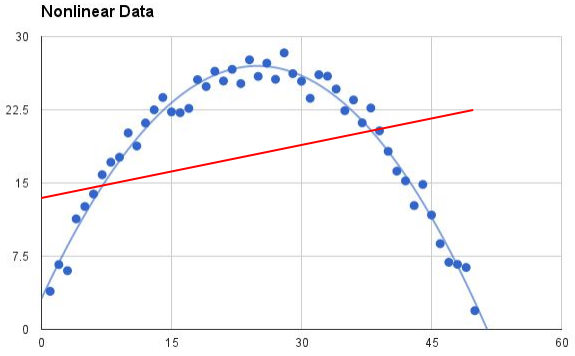
Pris en charge par la validation croisée



LES ALGORITHMES

Régression Linéaire



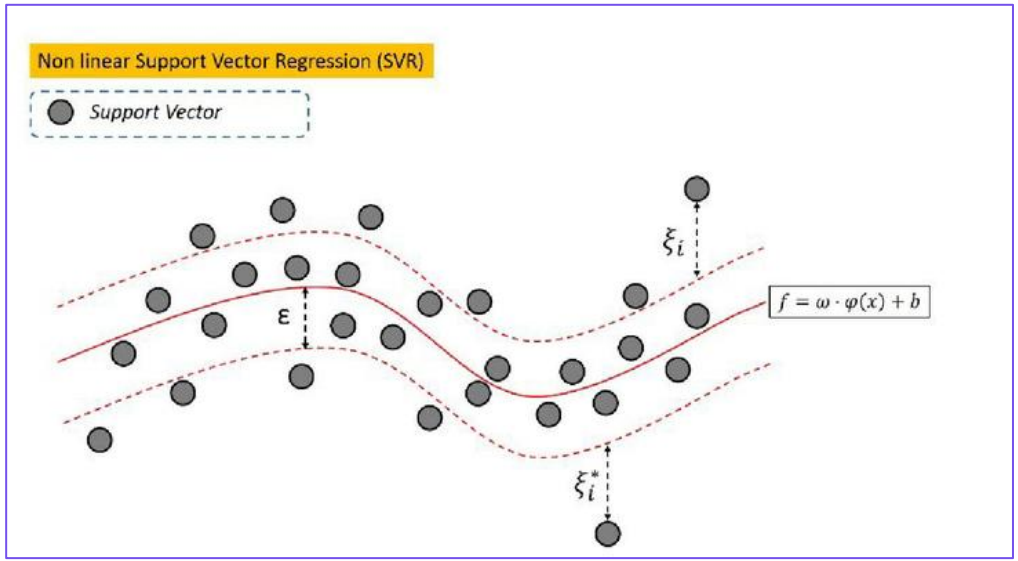


Variantes Ridge et Lasso

- Moyen de contrôler l'overfitting d'une RL
- Il s'agit de RL dont on "pénalise" les apprentissages

Augmentation du BIAIS pour diminuer la VARIANCE

Régression à vecteur de support : SVR

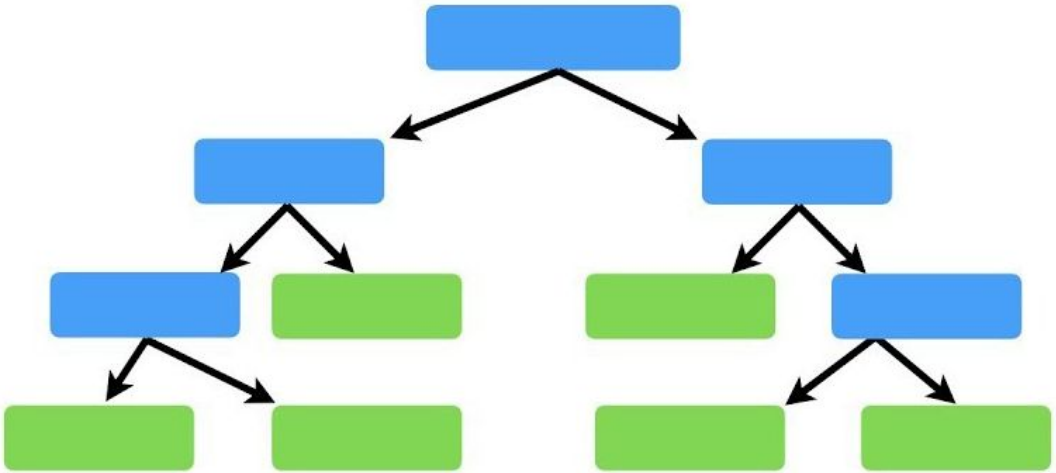


Algorithme “black box”

Hyperplan de régression au centre d’un “cylindre” dont on définit la largeur, et qui contiendrait un maximum des points.

Hyperplans non linéaires

Arbre de décision



Par le calcul, est déterminée pour chaque “node” la condition de partition des observations apportant le “meilleur score RSS” à une modélisation hypothétique ayant comme prédictions les moyennes du G1 et du G2.

On répète le processus, qui crée une arborescence de conditions, jusqu’à atteindre des limites fixées par des paramètres faisant en sorte que les “nodes” en bout de chaine sont des “feuilles” qui porte les valeurs de prédictions possibles.

Algo. ensemblistes parallèles : Bagging

Faire travailler N “apprenant faible” et mettre leurs résultats en commun

Bagging Regressor

N arbres de décisions “un peu” différents puisque entraînés sur des échantillons “bootstrapés” du dataset.

Random Forest Regressor

Même principe, mais en plus, chaque arbre ne prend en compte qu’une partie des variables définissant les observations.

Algo. ensemblistes séquentiels : Boosting

Adaboost

Enchaînement de “stumps” dont les résultats déterminent :

- un poids en vue du vote final
- les données sur lesquelles travailleront les arbres suivants

Gradient Boosting

Minimisation à “petit pas” d’une fonction d’erreur née d’une prédiction arbitraire

- Principe totalement différent
- Chaque apprenant contribue à faire “converger” le modèle vers les valeurs à prédire.

Pourquoi encoder ?

Label Encoding

color	color
red	0
green	1
blue	2

One Hot Encoding

Pet	Cat	Dog	Turtle	Fish
Cat	1	0	0	0
Dog	0	1	0	0
Turtle	0	0	1	0
Fish	0	0	0	1
Cat	1	0	0	0

Target Encoding

On remplace une catégorie par la moyenne des valeurs de la cible pour chaque observation en faisant partie.

Quarté gagnant

- **ridge** + OHE
- **SVR** + OHE
- **Random Forest** + TE
- **Gradient Boosting** + TE

VARIABLE	NB de CATÉGORIES
BuildingType	7
PrimaryPropertyType	31
LargestPropertyUseType	57
SecondLargestPropertyUseType	51
ThirdLargestPropertyUseType	44

```
X["BuildingType"].value_counts()
```

NonResidential	1449
Multifamily LR (1-4)	1020
Multifamily MR (5-9)	553
Multifamily HR (10+)	107
SPS-District K-12	84
Nonresidential COS	67
Campus	23

Utilisation d'une fonction "maison" qui...

- En prenant les variable dans le sens décroissant de leur nombre de catégories
- Et après avoir classé les catégories dans le sens décroissant de leur fréquence

Va vérifier si se passer de N catégories améliore la modélisation, et le cas échéant supprime ces catégories.

Diminution sensible de la complexité du modèle...

Recours à deux fonction de FEATURES SELECTION

RFECV

SelectPercentile

Diminution sensible de la complexité du modèle...

Recherche des meilleurs “hyper-paramètres”

```
# grille d'hyper-paramétrage GRADIENT BOOSTING
param_gb_rand = {"learning_rate" : [0.05, 0.075, 0.1, 0.125, 0.15],
                 "n_estimators" : [50, 100, 150, 200, 250],
                 "subsample" : [0.8, 0.85, 0.9, 0.95, 1],
                 "max_depth" : [2, 3, 4, 5, 10],
                 "min_samples_split" : [10, 20, 30, 40],
                 "min_samples_leaf" : [5, 10, 20, 30]
                }
```

10 000 modélisations à effectuer !

1 - Recherche randomisée

2 - Recherche précise

	Model	Fit_Time	Score_time	MAE	Std MAE	RMSLE	Std RMSLE	R2	Std R2
0	best_ridge	0.033002	0.010000	0.326486	±0.0175	0.031402	±0.0029	0.822575	±0.0247
1	best_SVR	3.911224	0.698040	0.310002	±0.0187	0.031113	±0.0033	0.822234	±0.0270
2	best_RF	18.097035	0.303018	0.319439	±0.0144	0.031271	±0.0030	0.821934	±0.0257
3	best_GB	4.475256	0.019001	0.299881	±0.0132	0.028999	±0.0024	0.848079	±0.0201

City of Seattle

SOMMAIRE

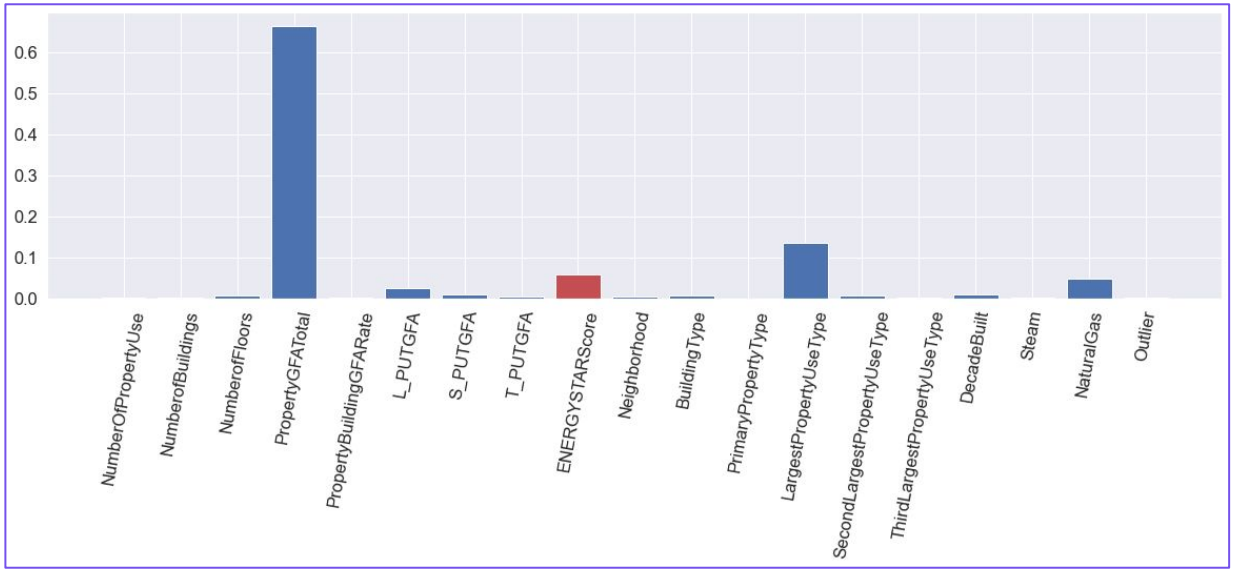
Importance d'une variable dans un modèle

1- Feature Importance

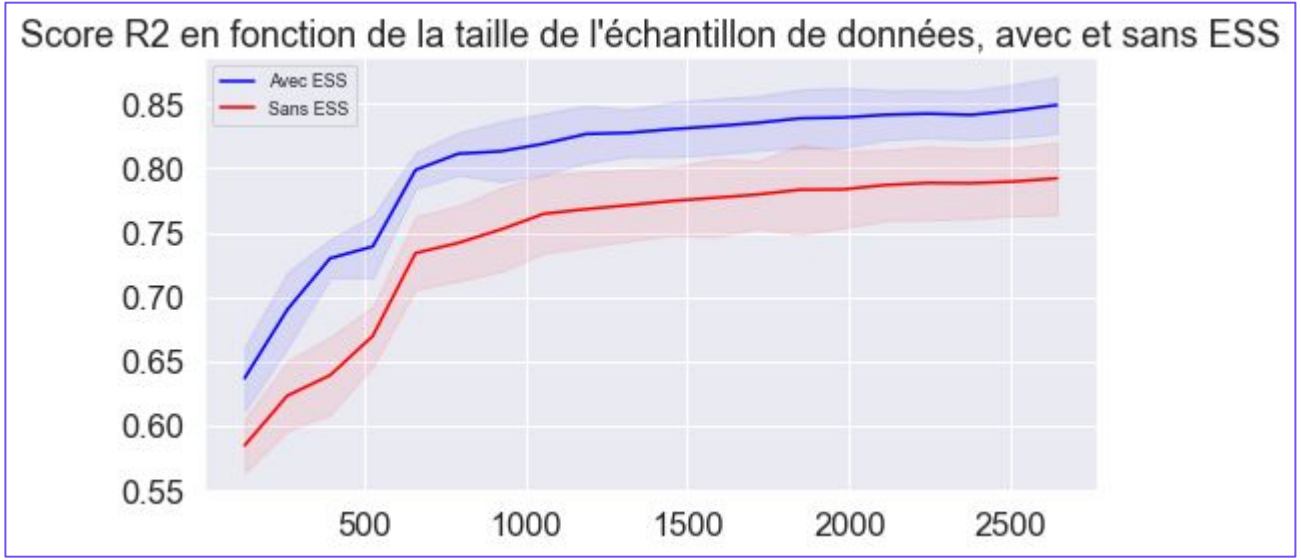
ESScore : 6%

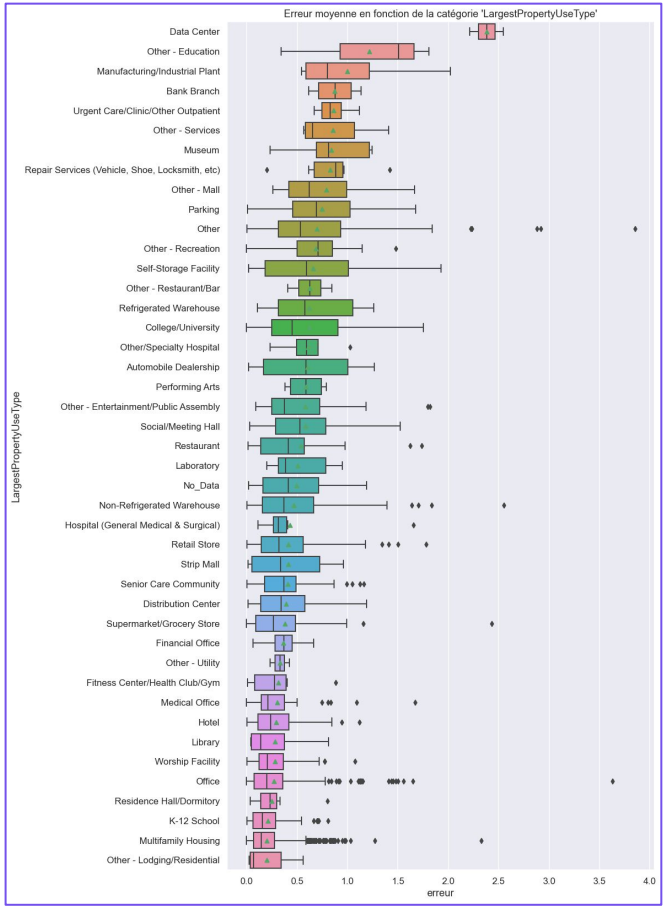
2- Modélisation sans...

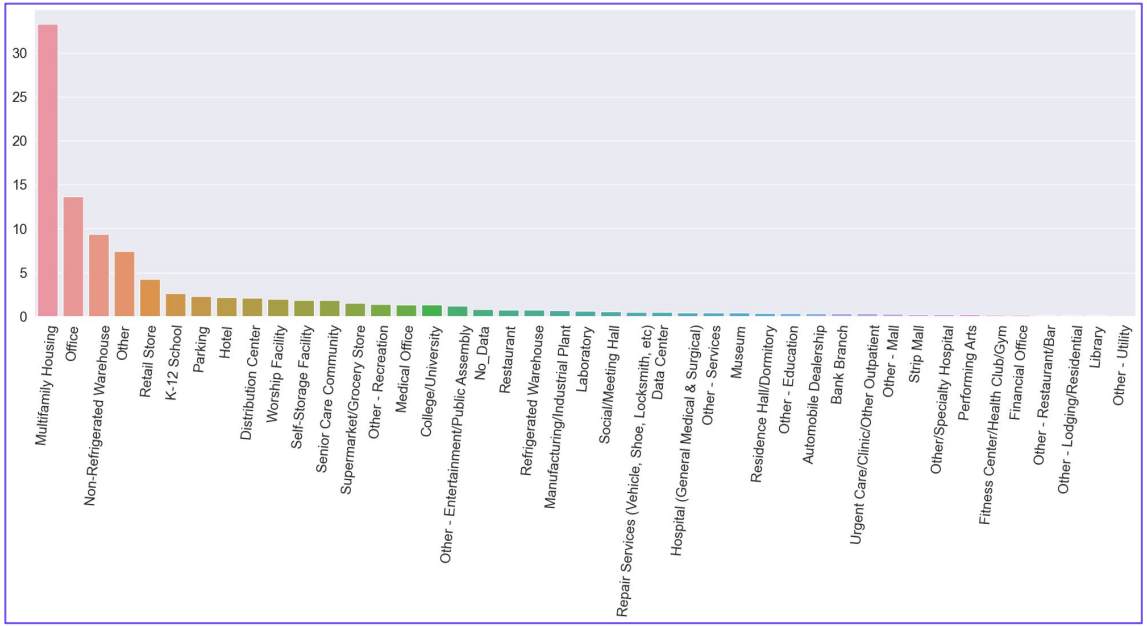
-0.05 R2 Score



Courbe d'apprentissage du modèle

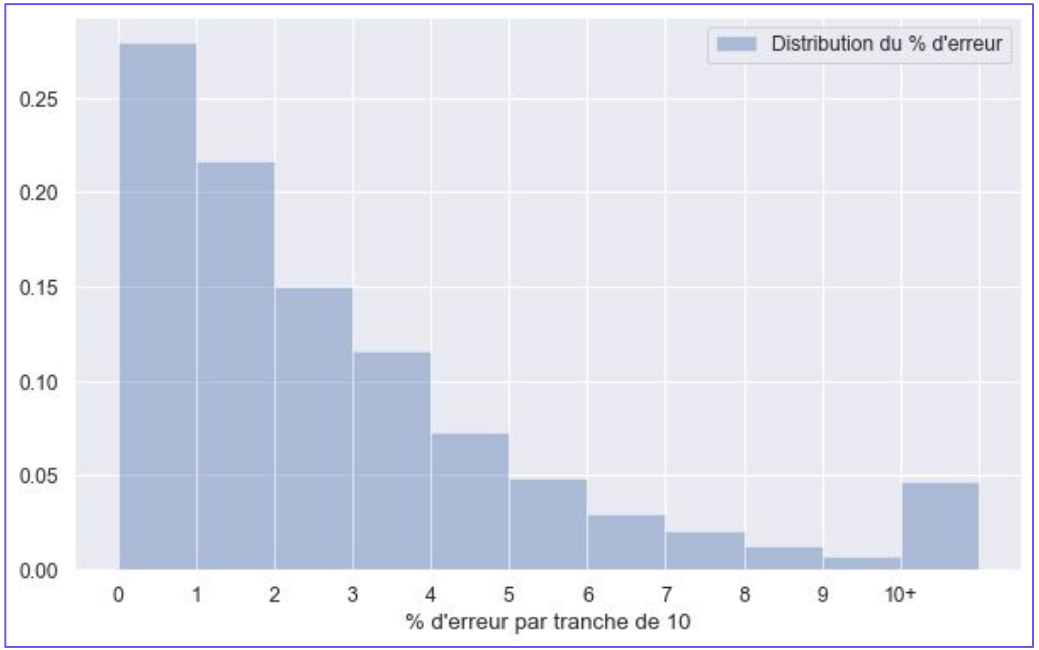






La rareté est facteur d'erreur

L'erreur globale est majoritairement constituée d'une accumulation de "petites erreurs".



MAIS...

Si on compare :

$\sum$ **Prédictions**

avec

$\sum$ **Valeurs Cibles**

10 % d'erreur !

Prédictions individuelles difficilement utilisables

Prédiction collective valable

En l'état, on dispose d'un modèle à utiliser avec **précaution.**

On peut le peaufiner à la marge en utilisant différents moyens **techniques.**

Pour vraiment l'améliorer, il faut avant tout plus de **données** !