P4 – ANTICIPEZ LES BESOINS EN CONSOMMATION ÉLECTRIQUE DE BÂTIMENTS

07/05/2021

Etudiant : Luc Rogers
Mentor : Etienne Sanchez

Sommaire

Problématique

Problématique

Seattle

- Prédire les besoins énergétiques des bâtiments de la ville de Seattle

- Objectifs :
  - Prédire consommation d'énergie
  - Prédire émissions de CO2
  - Evaluer l'influence de l'ENERGY Star Score sur la prédiction des émissions

Base de données open source :

https://www.kaggle.com/city-of-seattle/sea-building-energy-benchmarking#2015-building-energy-benchmarking.csv

Nettoyage

Nettoyage

- Vérification des doublons ➔ 0 doublons
- Sélection des variables avec < 30% de valeurs manquantes
- Première sélection à la main ➔ voir slide suivante
- Suppression des lignes avec targets vides
- Concaténation des deux années 2015 et 2016
  - Groupby sur la variable OSEBuildingID
  - Si doublon on ne garde que la ligne la mieux renseignée
- Suppression des valeurs aberrantes:
  - Total consommation ≥ Somme conso par type d'énergie
  - Valeurs négatives
  - Outliers: suppression à la main (voir slide correspondante)

➔ ***25 colonnes supprimées***

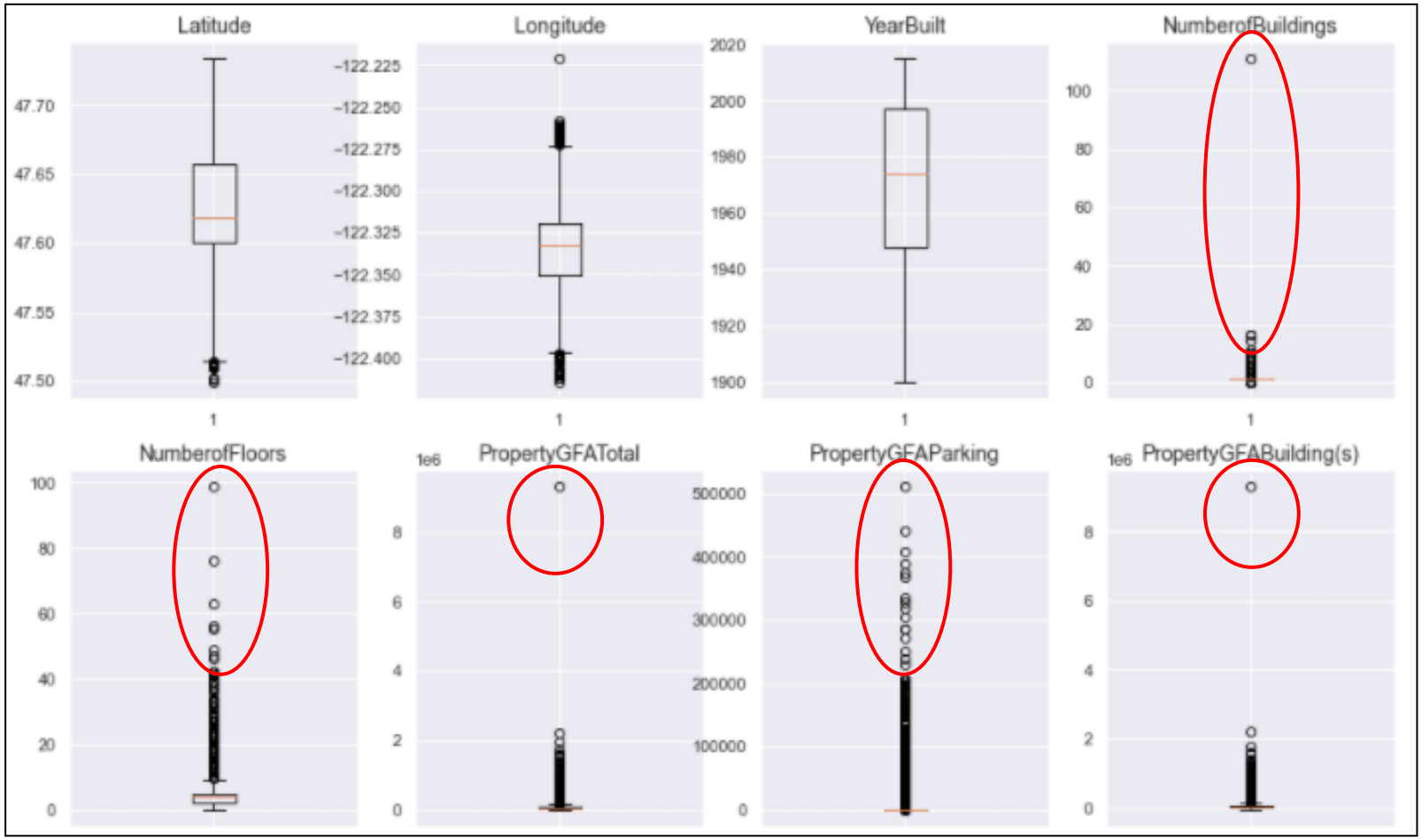
➔ ***146 lignes supprimées (~4% du nombre total de bâtiments uniques)***

Nettoyage - Variables sélectionnées:

- Données permettant la jointure des deux dataframes:
  - OSEBuildingID: ID du bâtiment
  - DataYear: année des relevés de consommation
- Données relatives à la problématique métier:
  - BuildingType : type du bâtiment (hôtel, caserne de pompier...)
  - PrimaryPropertyType : activité principale du bâtiment
  - Neighborhood: quartier
  - Latitude et Longitude
  - YearBuilt: année de construction
  - NumberofBuildings: nombre de bâtiments
  - NumberofFloors: nombre d'étages
  - PropertyGFATotal: surface totale
  - PropertyGFAParking: surface allouée au parking (consommation quasi nulle)
  - PropertyGFABuilding(s): surface allouée au bâtiment (information a priori redondante)
  - ListOfAllPropertyUseTypes: liste de toutes les activités du bâtiment
  - LargestPropertyUseType: activité dont la surface est la plus élevée
  - LargestPropertyUseTypeGFA: surface allouée à cette activité
  - ENERGYSTARScore: indice censé représenter la bonne utilisation des ressources énergétiques

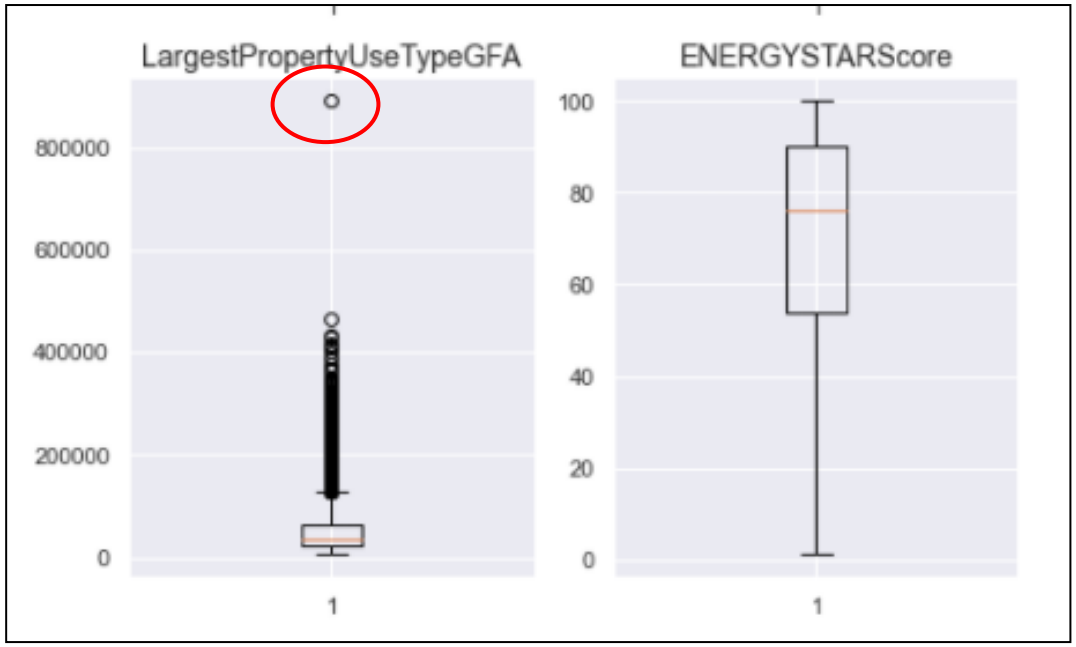
Nettoyage

- Suppression des outliers à la main:



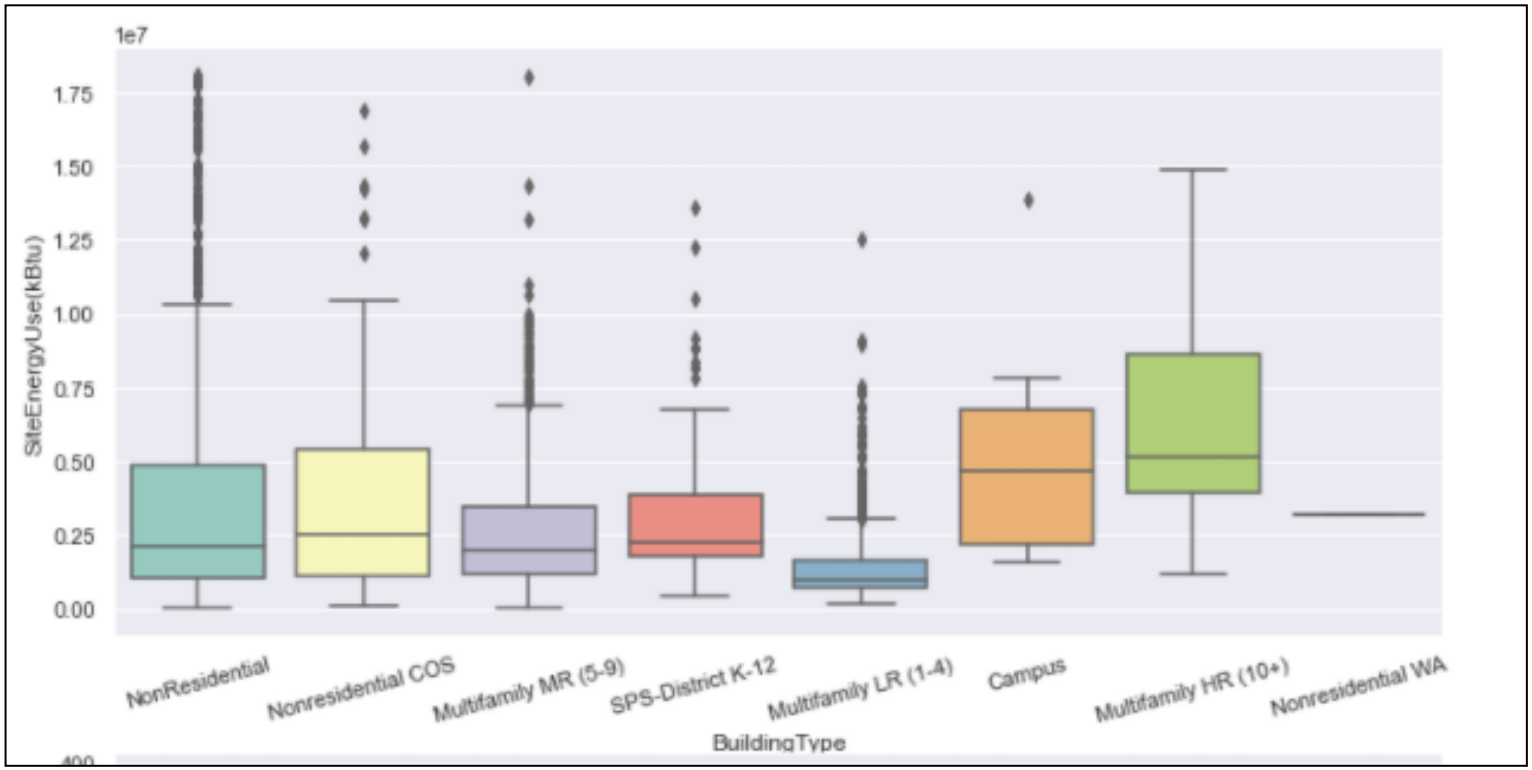
Nettoyage

- Suppression des outliers à la main:



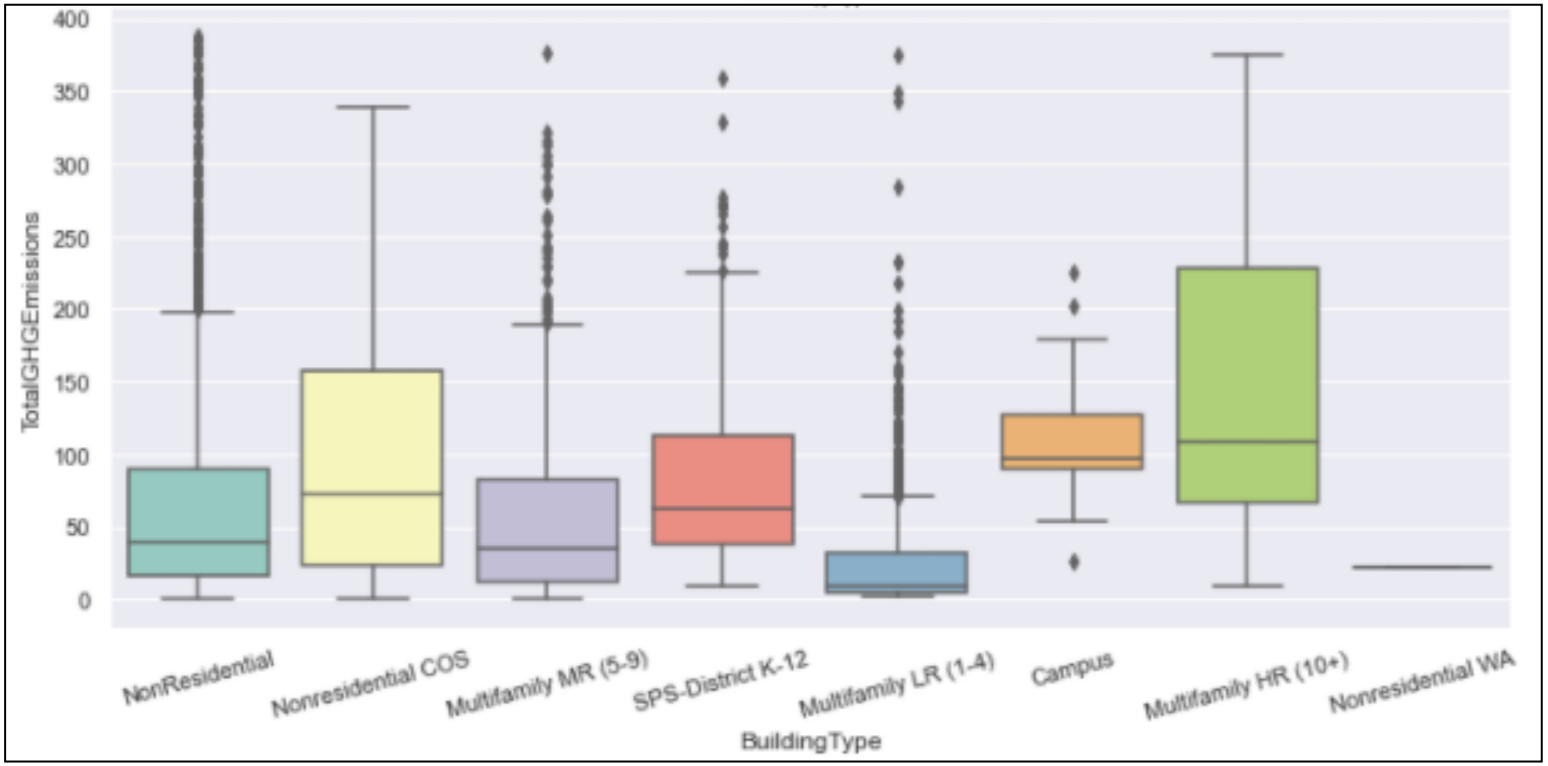
Exploration

Exploration



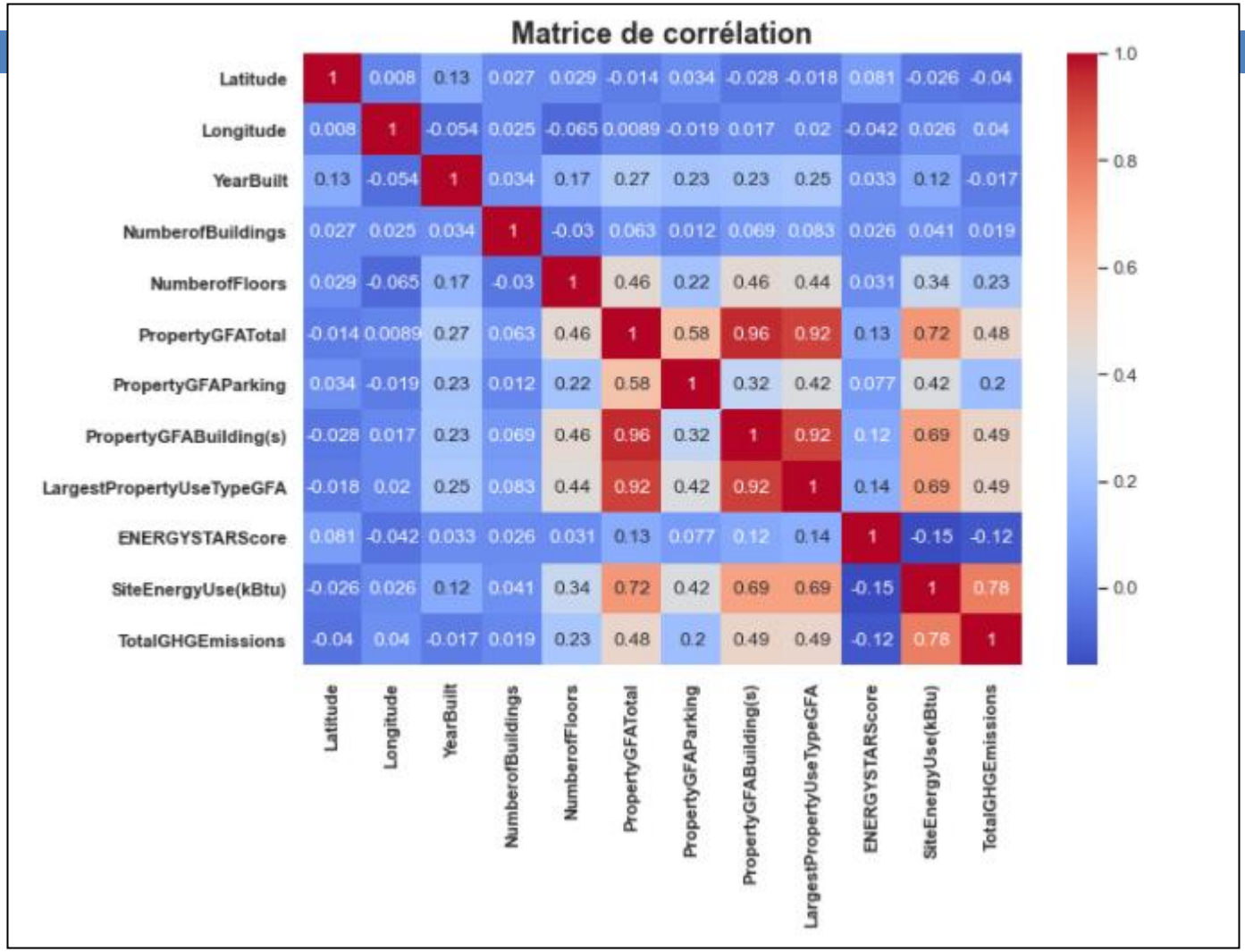
➔ Des distributions différentes selon le type de bâtiment

Exploration



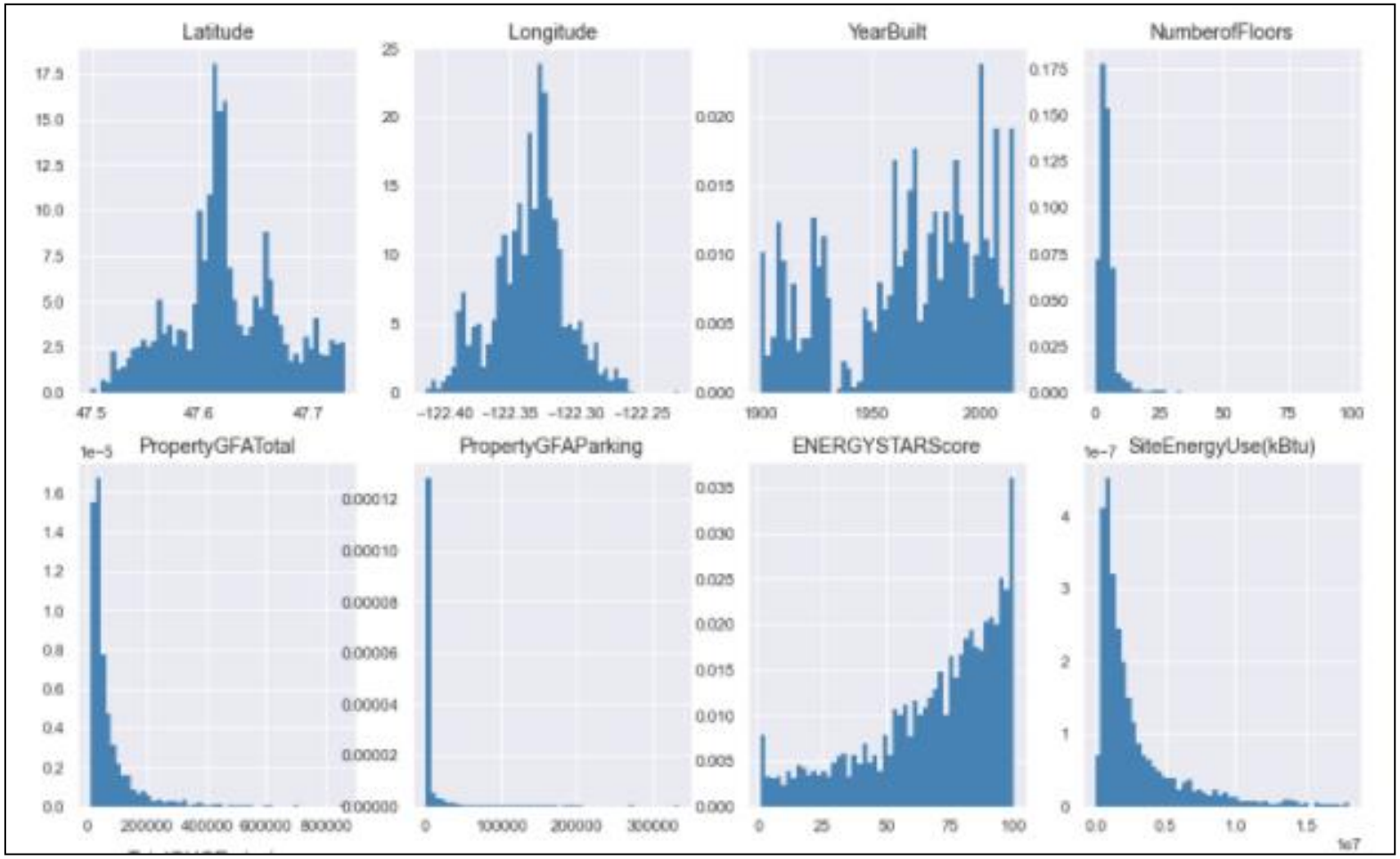
➔ Des distributions différentes selon le type de bâtiment

Exploration



➔ On supprime les variables explicatives corrélées

Exploration

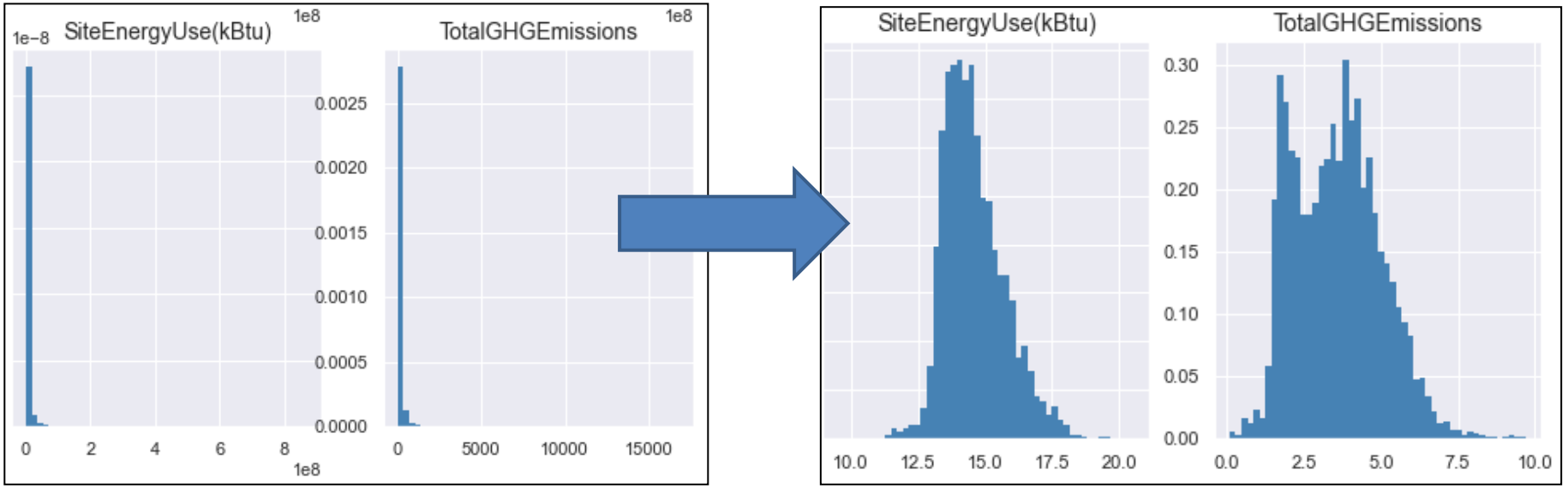


➔ Certaines features présentent une forte asymétrie positive

Feature Engineering

Feature engineering

- Traitement valeurs manquantes
  - Variable numérique ➔ moyenne
  - Variable catégorielle ➔ mode
- Transformation log(1+x) des features à forte asymétrie positive:



- Encodage des variables catégorielles
- Utilisation d'un scaler seulement pour les algorithmes le nécessitant

Feature engineering

Récapitulatif des transformations appliquées à chaque modèle:

Modèle	split random state=0	Transform Target	Cross validation	Scaling	ACP	Nombre hyper params
kNN	X	X	X	X	X	1
Lasso	X	X	X	X	X	1
Random Forest	X	X	X			2
Réseau neurones	X	X	X			4

Feature engineering

- Différents jeux de données selon la cible à étudier

  - Consommation énergétique:
    - X0: jeu de données sans ENERGY Star Score

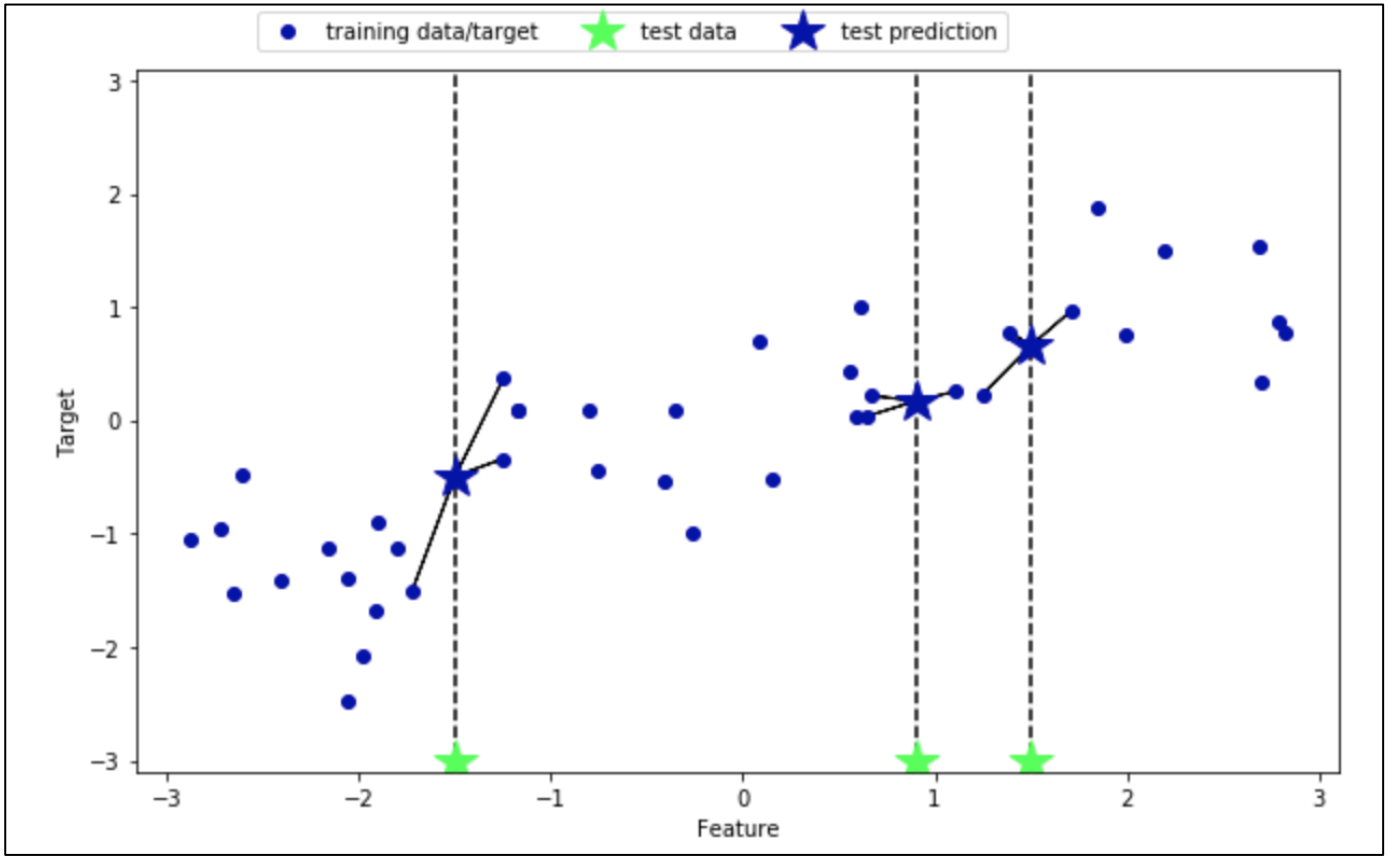
  - Emissions de CO2:
    - X1: jeu de données sans ENERGY Star Score
    - X2: jeu de données avec ENERGY Star Score

Modélisations

- kNN
- Lasso
- Random Forest
- Neural Network

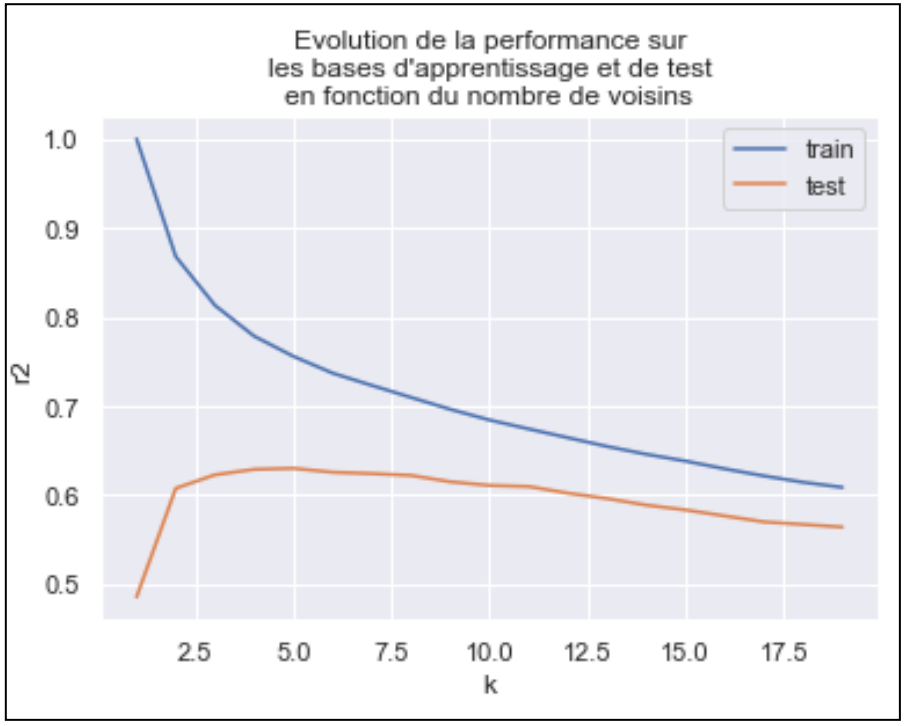
Modélisations - kNN

- k-NN – Méthode des k plus proches voisins
- Illustration du concept:



Modélisations - kNN

- k-NN – Méthode des k plus proches voisins
  - On trouve l'hyperparamètre k par validation croisée

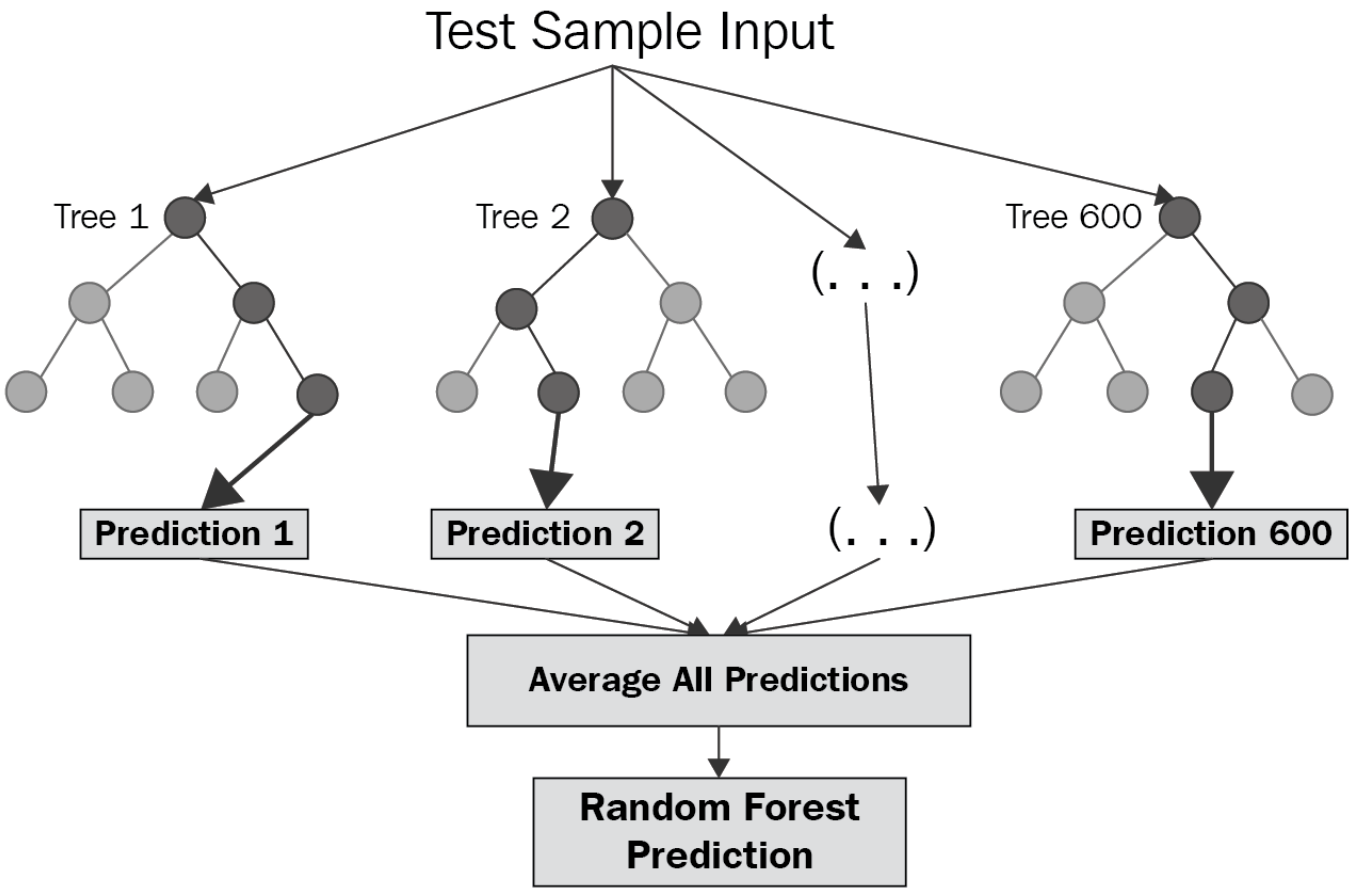


➔ *Globalement peu efficace sur ce jeu de données*

➔ *Temps de calcul assez faibles, facilité de paramétrage (un seul hyperparamètre)*

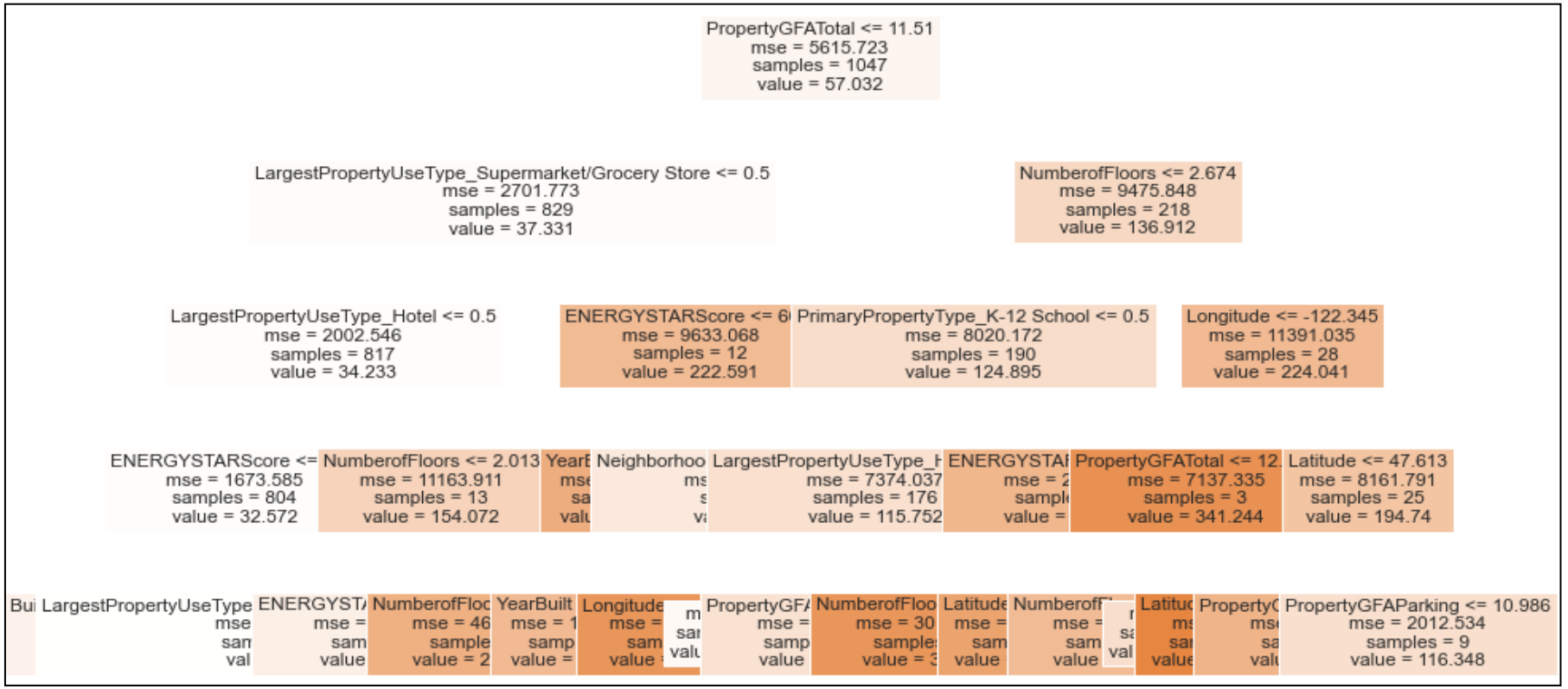
Modélisations – Random Forest

- Random Forest :



Modélisations – Random Forest

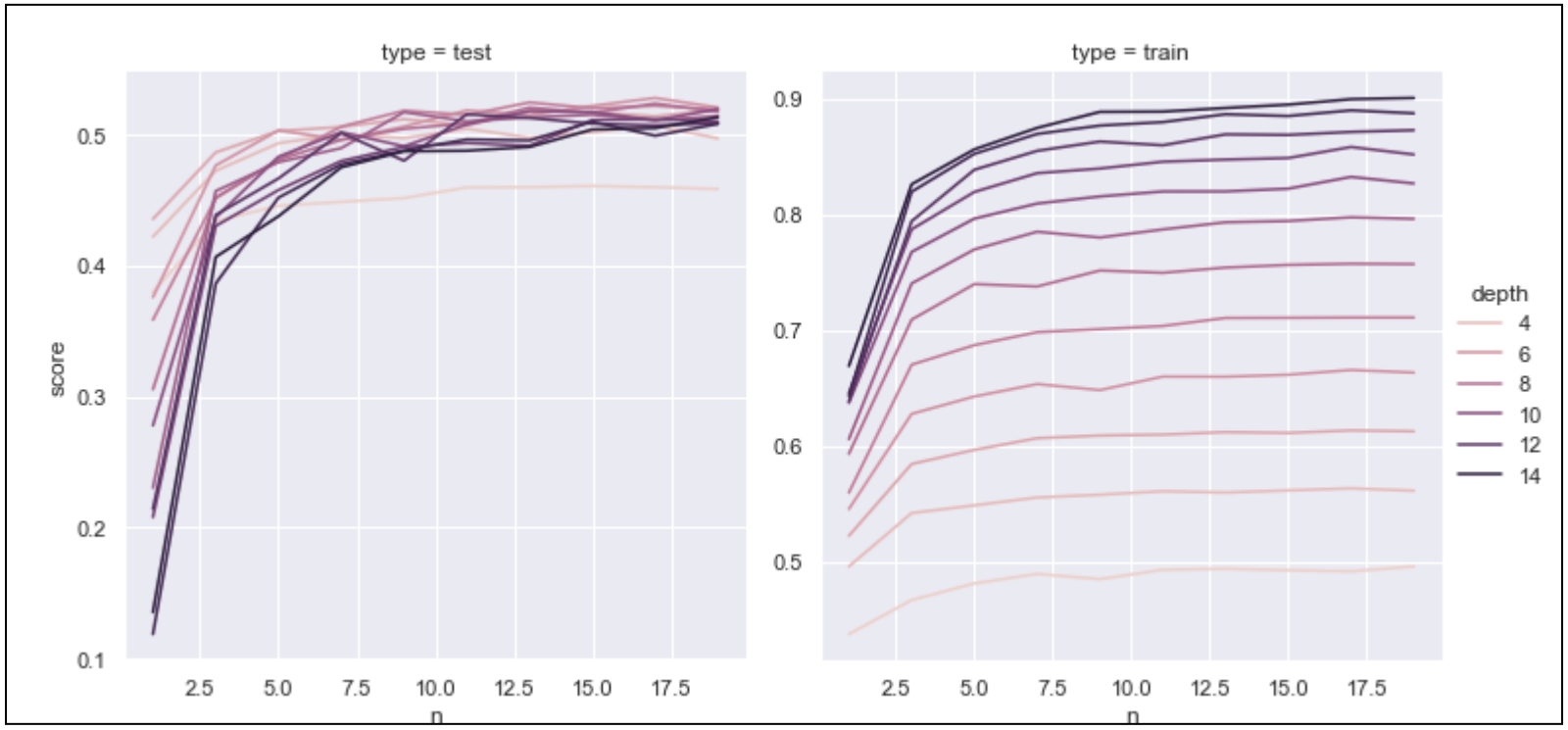
- Random Forest – exemple d'arbre de décision



- *Exemple d'arbre de décision*
- *Permet de faire ressortir les variables les plus utiles*

Modélisations - Random Forest

- Random Forest

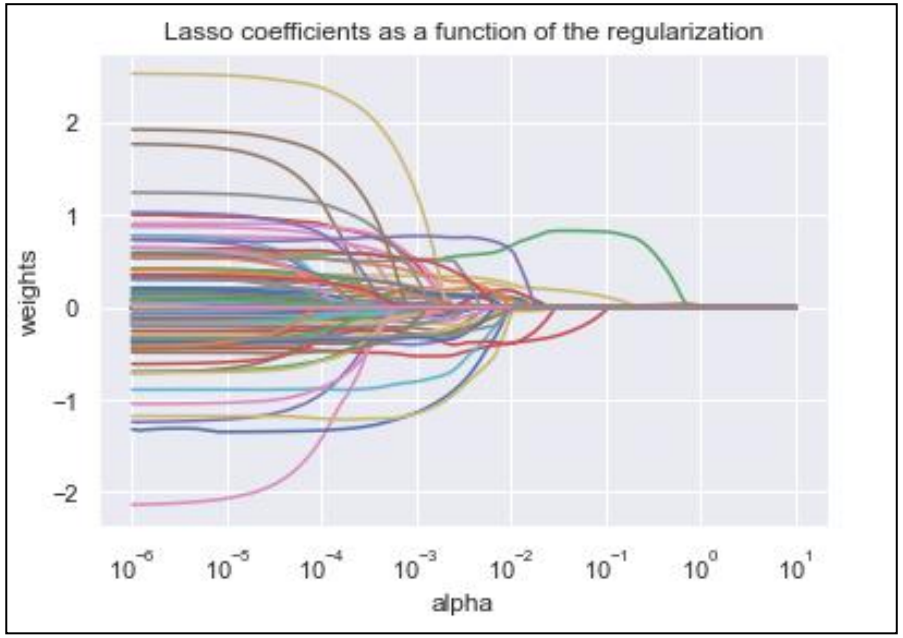


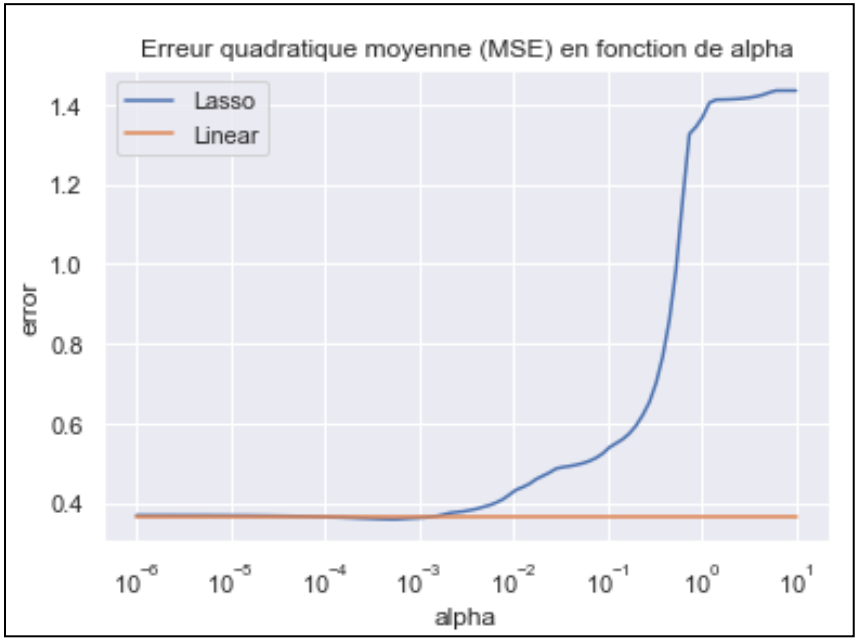
➔ *Avantages: converge très facilement*

➔ *Optimum sur la profondeur des arbres utilisés pour éviter sur-apprentissage*

Modélisations - Lasso

- Régression Lasso – feature selection





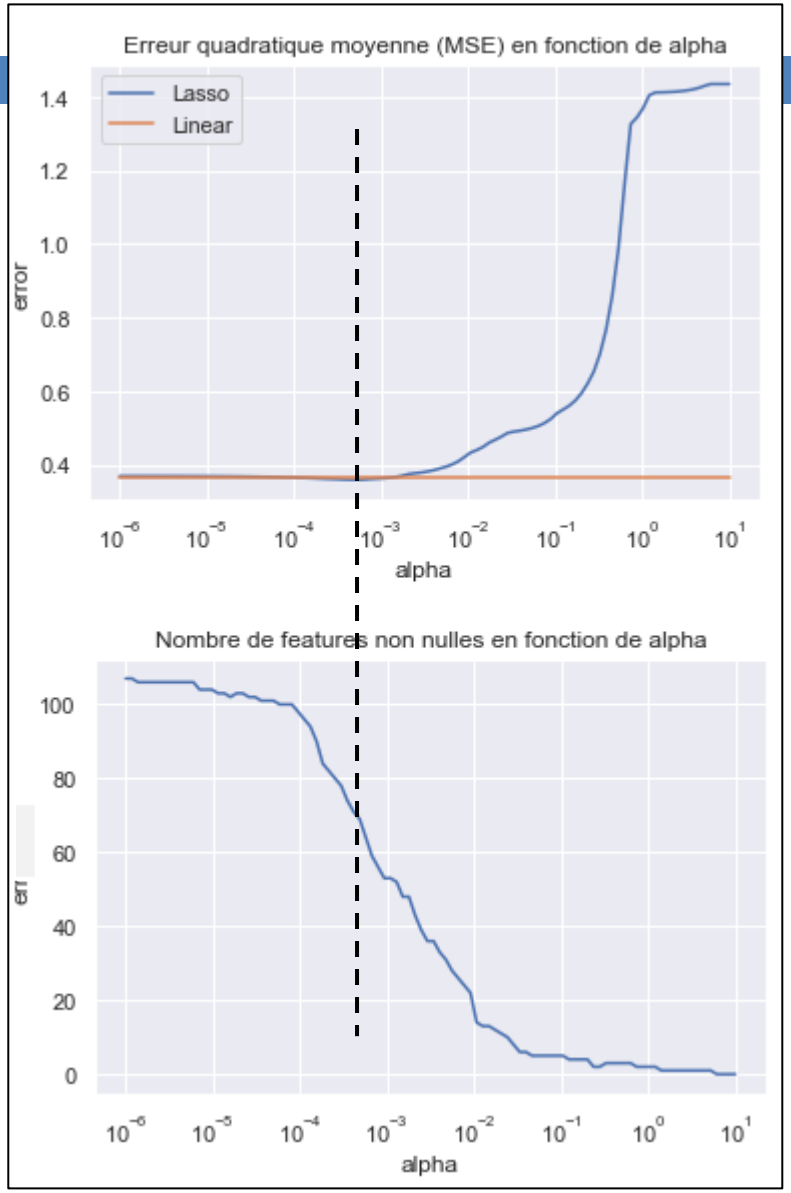
➔ *Gain faible sur l'erreur*

➔ *Environ la moitié des features peuvent être supprimées*

Modélisations - Lasso

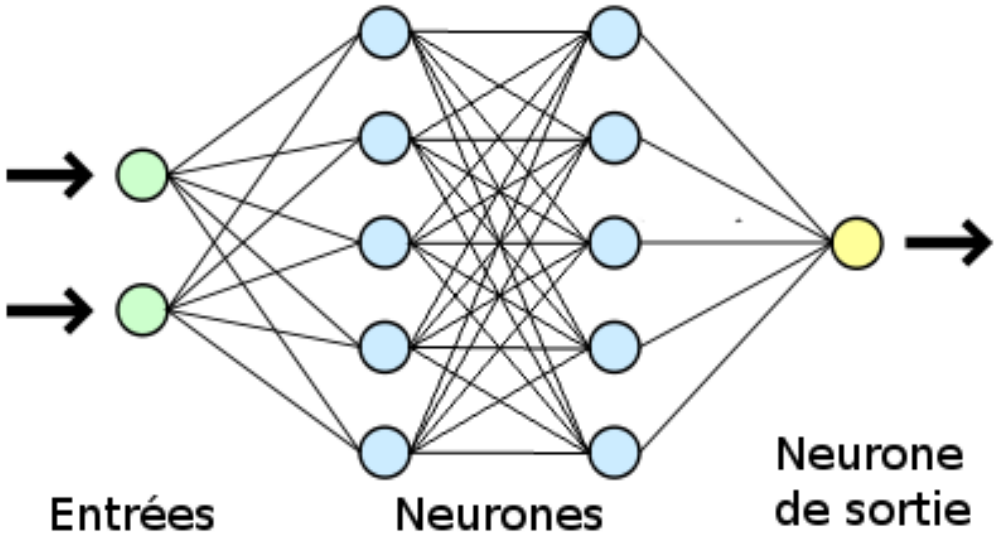
- Régression Lasso

➔ ***Permet la sélection des features pertinentes***



Modélisations – Réseau de neurones

- Réseau de neurones séquentiel:
  - Feed forward
  - Réputé pour traiter efficacement systèmes non linéaires (traitement de l'image, …)
  - Descente du gradient (risque d'optimum local différent en fonction de l'initialisation)



Modélisations – Réseau de neurones

- Réseau de neurones séquentiel
  - Résultats grid search:

	X0 (Conso énergie)			
grid search	epochs	100	300	600
	batch_size	32	48	64
	n_layer	5	10	15
	n_unit	5	10	20

choix optimal	epochs	300
	batch_size	64
	n_layer	10
	n_unit	5

X1 (CO2 sans ESS)		
epochs	200	300
batch_size	48	64
n_layer	10	12
n_unit	5	7

epochs	200
batch_size	64
n_layer	10
n_unit	7

X2 (CO2 avec ESS)		
epochs	200	300
batch_size	48	64
n_layer	10	12
n_unit	5	7

epochs	300
batch_size	64
n_layer	10
n_unit	5

➔ *Paramétrage très coûteux en calculs/temps de réglage*

➔ *Mais: les meilleures performances*

Comparatif

Comparatif

	Target	Features	Modèle	Hyperparamètre	Hyp. Opt.	R2	Temps calcul (s)
0	SiteEnergyUse(kBtu)	df_X0	k-NN	n_neighbors	10	0.497	8.30
1	TotalGHGEmissions	df_X1	k-NN	n_neighbors	13	0.333	6.15
2	TotalGHGEmissions	df_X2	k-NN	n_neighbors	10	0.339	6.36
3	SiteEnergyUse(kBtu)	df_X0	Lasso	alpha	5.59E-03	0.606	16.78
4	TotalGHGEmissions	df_X1	Lasso	alpha	2.92E-03	0.402	11.87
5	TotalGHGEmissions	df_X2	Lasso	alpha	2.92E-03	0.481	12.02
6	SiteEnergyUse(kBtu)	df_X0	Random Forest	(n_estimators, max_depth)	(36, 11)	0.672	134.37
7	TotalGHGEmissions	df_X1	Random Forest	(n_estimators, max_depth)	(41, 10)	0.436	98.69
8	TotalGHGEmissions	df_X2	Random Forest	(n_estimators, max_depth)	(41, 11)	0.425	103.60
9	SiteEnergyUse(kBtu)	df_X0	Neural Network	(epochs, batch_size, n_layers, n_units)	{'kr__batch_size': 64, 'kr__epochs': 300, 'kr_...	0.674	7012.48
10	TotalGHGEmissions	df_X1	Neural Network	(epochs, batch_size, n_layers, n_units)	{'kr__batch_size': 64, 'kr__epochs': 200, 'kr_...	0.512	512.05
11	TotalGHGEmissions	df_X2	Neural Network	(epochs, batch_size, n_layers, n_units)	{'kr__batch_size': 64, 'kr__epochs': 300, 'kr_...	0.580	539.78

Comparatif

Target = Consommation énergétique

	Target	Features	Modèle	Hyperparamètre	Hyp. Opt.	R2	Temps calcul (s)
0	SiteEnergyUse(kBtu)	df_X0	k-NN	n_neighbors	10	0.497	8.30
1	TotalGHGEmissions	df_X1	k-NN	n_neighbors	13	0.333	6.15
2	TotalGHGEmissions	df_X2	k-NN	n_neighbors	10	0.339	6.36
3	SiteEnergyUse(kBtu)	df_X0	Lasso	alpha	5.59E-03	0.606	16.78
4	TotalGHGEmissions	df_X1	Lasso	alpha	2.92E-03	0.402	11.87
5	TotalGHGEmissions	df_X2	Lasso	alpha	2.92E-03	0.481	12.02
6	SiteEnergyUse(kBtu)	df_X0	Random Forest	(n_estimators, max_depth)	(36, 11)	0.672	134.37
7	TotalGHGEmissions	df_X1	Random Forest	(n_estimators, max_depth)	(41, 10)	0.436	98.69
8	TotalGHGEmissions	df_X2	Random Forest	(n_estimators, max_depth)	(41, 11)	0.425	103.60
9	SiteEnergyUse(kBtu)	df_X0	Neural Network	(epochs, batch_size, n_layers, n_units)	{'kr__batch_size': 64, 'kr__epochs': 300, 'kr_...	0.674	7012.48
10	TotalGHGEmissions	df_X1	Neural Network	(epochs, batch_size, n_layers, n_units)	{'kr__batch_size': 64, 'kr__epochs': 200, 'kr_...	0.512	512.05
11	TotalGHGEmissions	df_X2	Neural Network	(epochs, batch_size, n_layers, n_units)	{'kr__batch_size': 64, 'kr__epochs': 300, 'kr_...	0.580	539.78

➔ *Random Forest aussi précis que Neural Network, R^2=0,672*

➔ *Mais temps de calcul et paramétrages beaucoup plus simples !*

Comparatif

Target = Emissions CO2

	Target	Features	Modèle	Hyperparamètre	Hyp. Opt.	R2	Temps calcul (s)
0	SiteEnergyUse(kBtu)	df_X0	k-NN	n_neighbors	10	0.497	8.30
1	TotalGHGEmissions	df_X1	k-NN	n_neighbors	13	0.333	6.15
2	TotalGHGEmissions	df_X2	k-NN	n_neighbors	10	0.339	6.36
3	SiteEnergyUse(kBtu)	df_X0	Lasso	alpha	5.59E-03	0.606	16.78
4	TotalGHGEmissions	df_X1	Lasso	alpha	2.92E-03	0.402	11.87
5	TotalGHGEmissions	df_X2	Lasso	alpha	2.92E-03	0.481	12.02
6	SiteEnergyUse(kBtu)	df_X0	Random Forest	(n_estimators, max_depth)	(36, 11)	0.672	134.37
7	TotalGHGEmissions	df_X1	Random Forest	(n_estimators, max_depth)	(41, 10)	0.436	98.69
8	TotalGHGEmissions	df_X2	Random Forest	(n_estimators, max_depth)	(41, 11)	0.425	103.60
9	SiteEnergyUse(kBtu)	df_X0	Neural Network	(epochs, batch_size, n_layers, n_units)	{'kr__batch_size': 64, 'kr__epochs': 300, 'kr_...	0.674	7012.48
10	TotalGHGEmissions	df_X1	Neural Network	(epochs, batch_size, n_layers, n_units)	{'kr__batch_size': 64, 'kr__epochs': 200, 'kr_...	0.512	512.05
11	TotalGHGEmissions	df_X2	Neural Network	(epochs, batch_size, n_layers, n_units)	{'kr__batch_size': 64, 'kr__epochs': 300, 'kr_...	0.580	539.78

➔ *Le Réseau de neurones est le plus efficace ici, $R^2 = 0{,}580$*

Conclusion

- Modèles retenus selon le score:
  - Consommation électrique: Random Forest
  - Emissions de CO2: Réseau de neurones séquentiel
- Amélioration des scores de prédiction en fonction de la complexité du modèle
- Mais parfois gain assez faible et non pertinent (RF ⇔ NN)
- Sans les données sur la consommation la prédiction des émissions devient bien plus difficile
- Utilité de l'ENERGY STAR Score démontrée

Merci de votre attention

Annexe

R^2 score pour kNN en fonction de N composantes ACP