Projet pour anticiper les besoins en consommation électrique de bâtiments

Cyril REGAN

Base de données : kaggle.com/city-of-seattle/

16 avril 2020

Table of Contents

Problématique

Prédire pour la ville de :

Seattle

- les émissions
- et les sources d'énergie

Jeu de donnée :

- en 2015 : 3340 rows $\times$ 47 columns
- en 2016 : 3376 rows $\times$ 46 columns

qui caractérisent

- des individus : batiments de Seattle
- des variables : ID, BuildingType, YearBuilt, ENERGYSTARScore, Zip Codes, etc...

Table of Contents

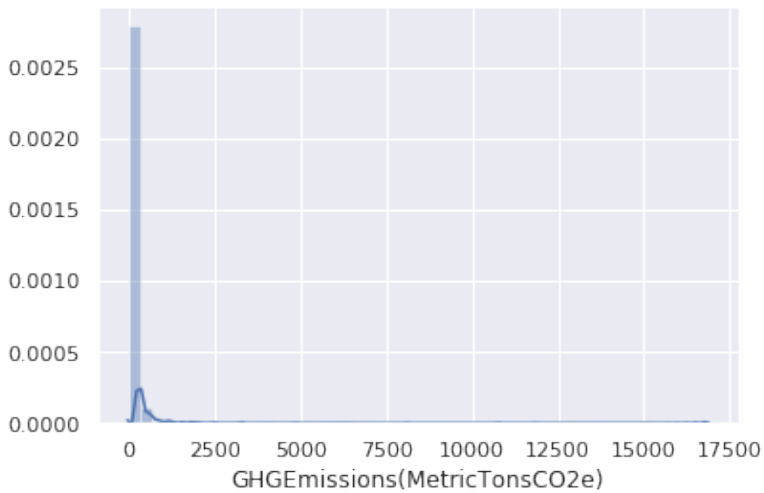
Nettoyage et selection des données

- Concaténation 2015 2016 en un seul tableau
- Selection d'une unique variable d'ordre géographique : *CouncilDistrictCode*
- Vérification des doublons sur l'ID des batiments par années
- Selection de 2 variables cibles : *GHGEmissions(MetricTonsCO2e)* et *SourceEUI(kBtu/sf)*
- Selection de 7 variables quantitatives : *CouncilDistrictCode*, *YearBuilt*, *NumberofBuildings*, *NumberofFloors*, *PropertyGFAParking*, *PropertyGFABuilding(s)*, *ENERGYSTARScore*
- Selection de 4 variables qualitatives : *BuildingType*, *LargestPropertyUseType*, *SecondLargestPropertyUseType*, *ThirdLargestPropertyUseType*

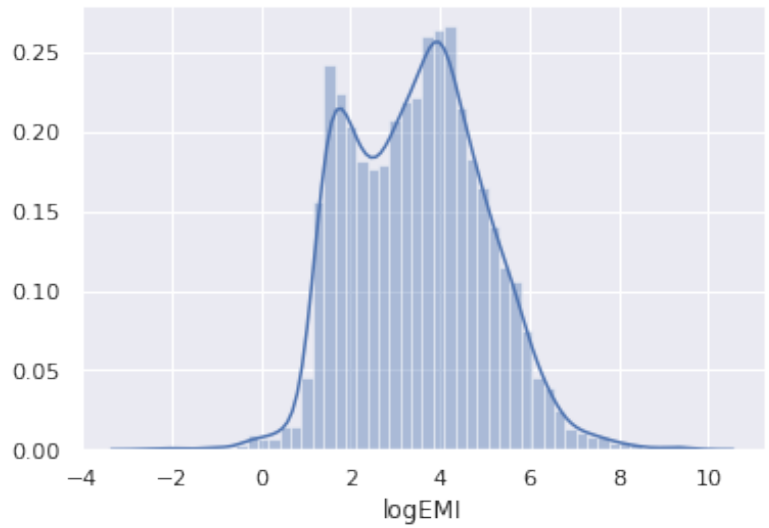
Transformation des variables quantitatives

- Variable cible de l'émission

distribution de l'émission



distribution de son log



=> nouvelle variable cible : logEMI

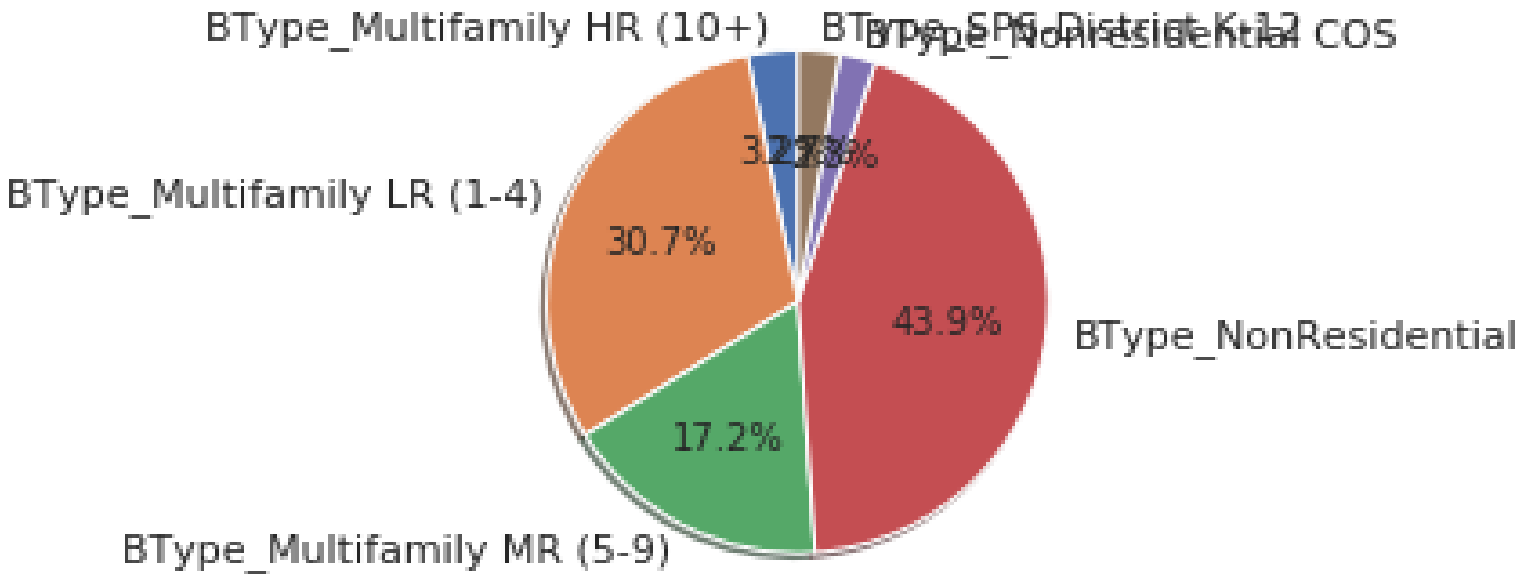
- Variable cible de la consommation

ConsoSourceBrute(kBtu) = $\underline{\text{SourceEUI(kBtu/sf)}} *$
$(PropertyGFAParking + PropertyGFABuilding(s))$
=> passage au log : ***logCONSO***

- *CouncilDistrictCode*
- *YearBuilt*
- *NumberofBuildings*=>***sqrtNbBuild***
- *NumberofFloors*=>***sqrtNbFloor***
- *PropertyGFAParking*=>***logGFAPark***
- *PropertyGFABuilding(s)*=> ***logGFABuild***
- *ENERGYSTARScore*

Transformation variables qualitatives

- BuildingType : OneHotEncoder à 5%



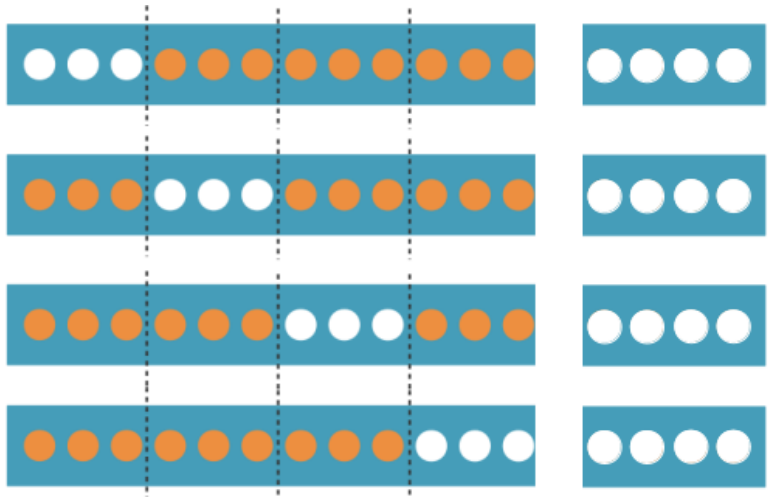
- PropertyUseType : OneHotEncoder à 4.6% **avec poids =**
  - **3** pour LargestPropertyUseType
  - **2** pour SecondLargestPropertyUseType
  - **1** pour ThirdLargestPropertyUseType

index	UType_Hotel	UType_K-12 School	UType_Multifamily Housing	UType_Non-Refrigerated Warehouse	UType_Office	UType_Other	UType_Parking	UType_Restaurant	UType_Retail Store
0	**3.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	**3.0**	0.0	0.0	0.0	0.0	0.0	**2.0**	**1.0**	0.0
2	**3.0**	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0
3	**3.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	**3.0**	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0

Table of Contents

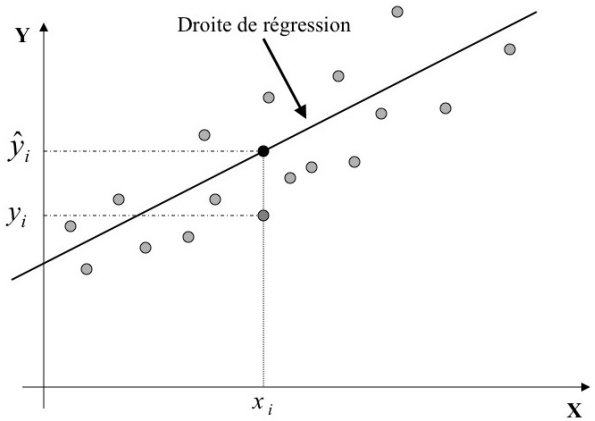
Présentation des modèles linéaires

- Regression linéaire
- Elastic net
- SVR
- KNN



Validation croisée avec *GridSearch* et *pipelines* (sklearn)

$X \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^n$ et $t > 0$



Régression Linéaire :

$$\underset{\beta \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \|y - X\beta\|_2^2$$

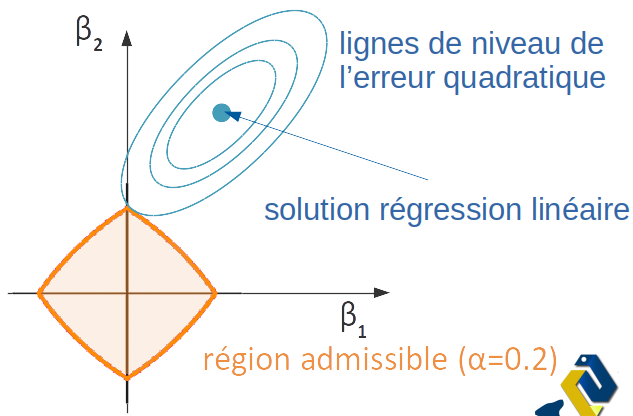
Elastic net :

$$\underset{\beta \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \|y - X\beta\|_2^2 + \lambda \left((1-\alpha)\|\beta\|_1 + \alpha\|\beta\|_2^2 \right)$$

ou

$$\underset{\beta \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \|y - X\beta\|_2^2$$

$$\text{t.q } (1-\alpha)\|\beta\|_1 + \alpha\|\beta\|_2^2 \leq t$$



=> **GridSearch :**

$\alpha = [0...1,5], \lambda = [-5... -1,5]_{log_{10}}$

Support Vector Regressor

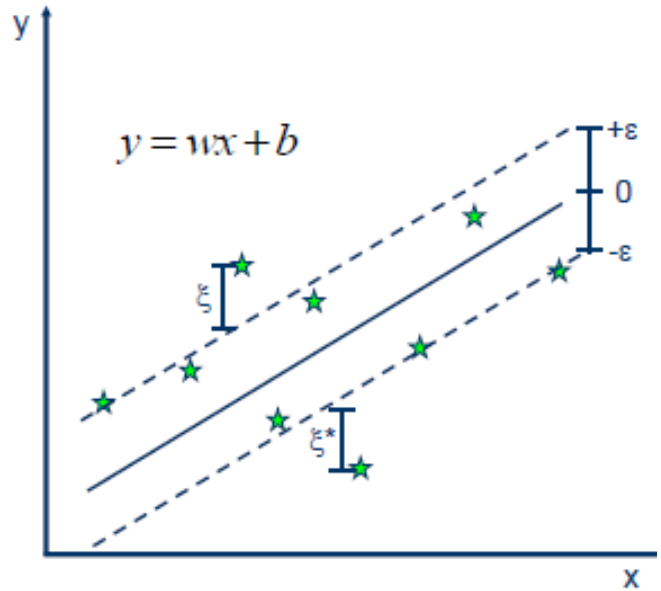
$$\underset{w\in\mathbb{R}^p, b\in\mathbb{R}, \xi\in\mathbb{R}^n_+, \xi^*\in\mathbb{R}^n_+}{\operatorname{argmin}} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n(\xi_i + \xi_i^*)$$

t.q

$$y_i - \langle w, x_i\rangle - b \leq \epsilon + \xi_i$$

et

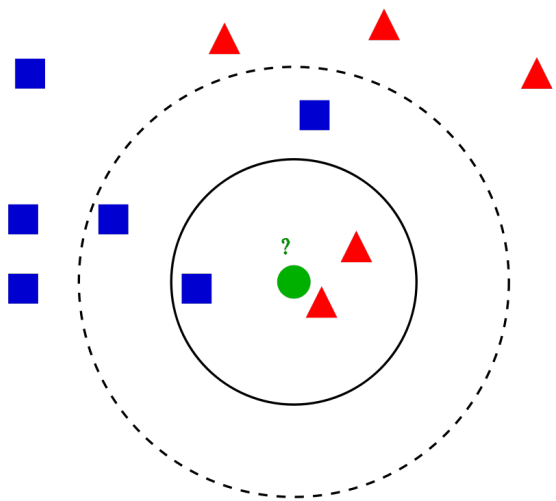
$$\langle w, x_i\rangle + b - y_i \leq \epsilon + \xi_i^*$$



=> **GridSearch :** $C = [-1...2, 5]_{log_{10}}$

K plus proches voisins (KNN)

=> **GridSearch :** $k = [1...40, 40]$



Présentation des modèles non linéaires

- Ridge à noyau
- SVR à noyau
- Réseau de neurones
- Forêts aléatoires
- Gradient boosting

Fonction de redescription ϕ :

Pb non Linéaire		Pb Linéaire
$X \in \mathbb{R}^p$	$\overset{\phi:\mathbb{R}^p \to \mathbb{R}^m}{\Longrightarrow}$	$\phi(X) \in \mathbb{R}^m$

L'astuce du noyau :

$$
\begin{aligned}
K(x,x') &= \langle \phi(x), \phi(x') \rangle &&= \langle x, x' \rangle \ (\textit{SVR linéaire}) \\
K(x,x') &= \langle \phi(x), \phi(x') \rangle &&= (\langle x, x' \rangle + c)^d \\
K(x,x') & &&= \exp\left(-\frac{\langle x - x', x - x' \rangle}{2\sigma^2} \right)
\end{aligned}
$$

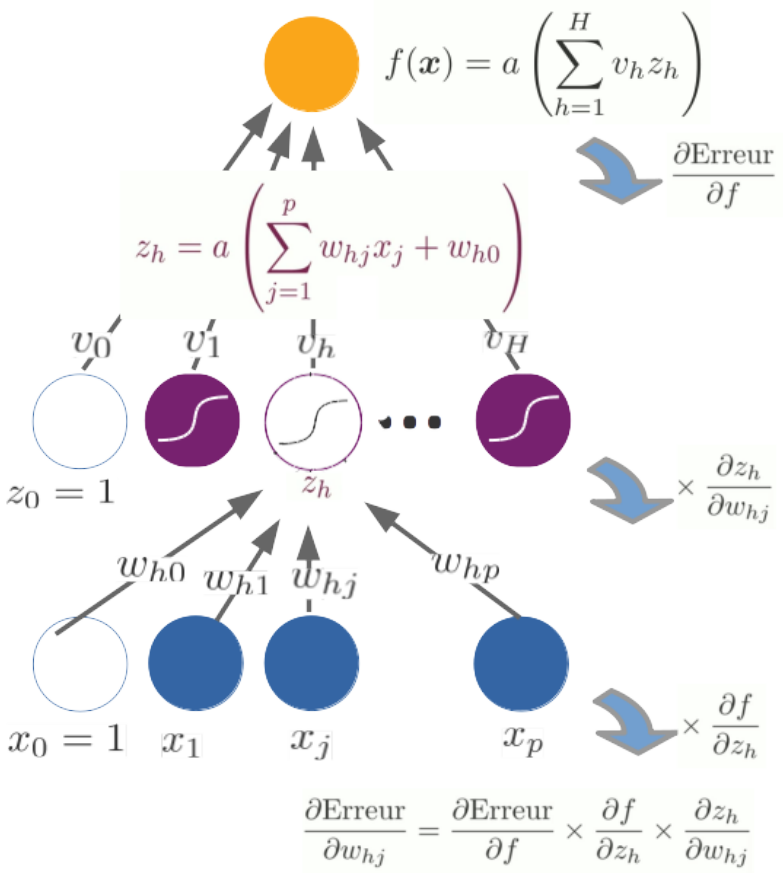
SVR à noyau => **GridSearch :** $\gamma = \frac{1}{2\sigma^2} = [-3.5... - 1.5, 3]_{log_{10}}$,
$\textit{kernel} = \textit{rbf}$, $C = [-1...2, 5]_{log_{10}}$

Regression Ridge (L2) à noyau => **GridSearch :** $\textit{kernel} = \textit{rbf}$,
$\gamma = \frac{1}{2\sigma^2} = [-3.5... - 1.5, 5]_{log_{10}}$, $\lambda = [-5...0, 5]_{log_{10}}$

Réseau de neurone :

- Initialisation poids aléatoires des variables (w_{hj}, v_h)
- Sortie de noeud $z_h = a_h(\sum w_{hj} x_j)$, a_h : fct° d'activation ...

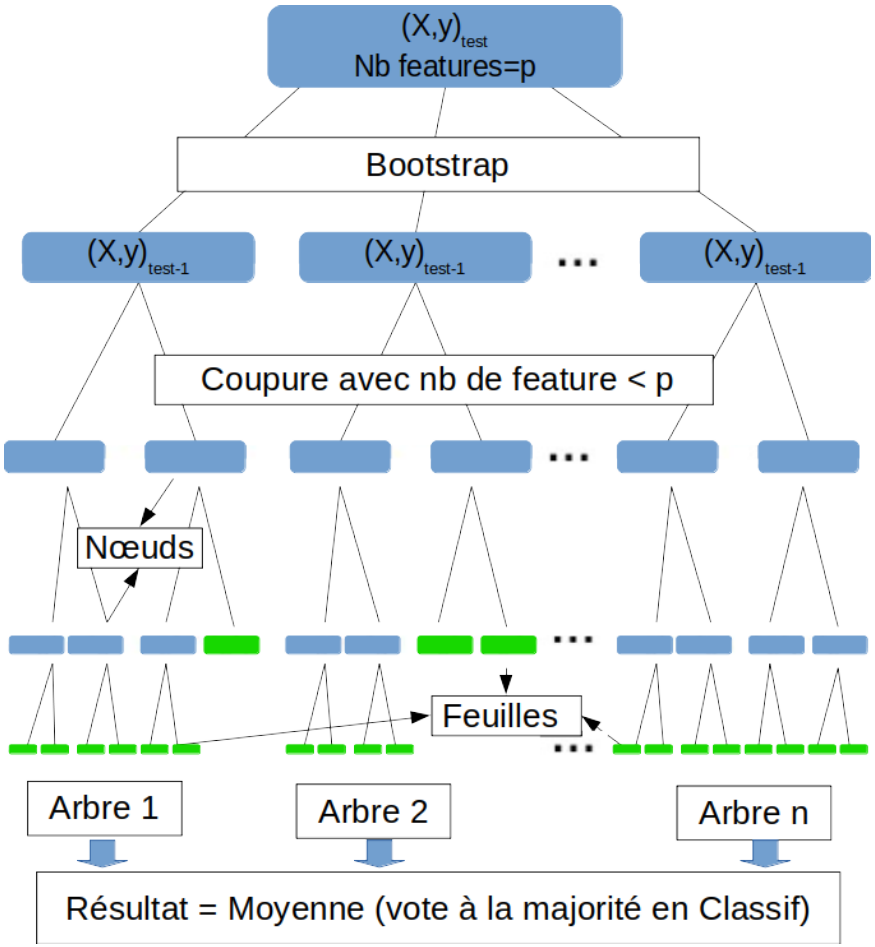
  $f(x) = a(\sum v_h z_h)$
- Descente de gradient par rétropropagation
- (Boucle iter sur toutes les obs) $\times$ nb d'époques



=> **GridSearch avec *mlpregressor* :** $iter = [50...300, 6]$,
$Nb_{neur} = (24) \times [1, 2, 3]$, $lear\_rate = [-4... - 1, 5]_{\log_{10}}$

Forêts aléatoires :

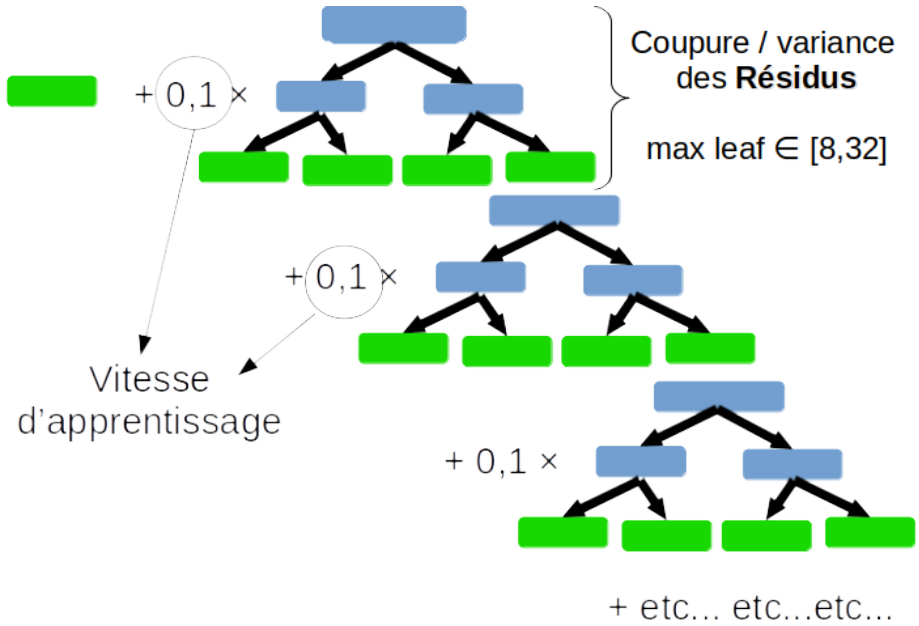
- **Tirage avec remise** (bootstrap)
- Coupure pour **minimiser la variance** (ou l'impureté en Classif) des nœuds fils avec un **nb réduit de feature (p/3)** tirés aléatoirement.
- **Critères d'élagage** pour réduire la taille des arbres : max-depth, min-samples-leaf
- **Résultats par la moyenne** de tous les arbres



=> **GridSearch :** $Nb\_arbres = [50...500, 6]$, $max\_depth = [10...70, 4]$, $min\_samples\_leaf = [1...10, 3]$, $max\_features = [5...20, 3]$

GradientBoost :

- Arbres **petits** de **même poids**
- Arbres construits **sur les résidus** (et non sur la variable cible)
- **Descente de Gradient** avec une vitesse d'apprentissage



=> **GridSearch :** $Nb\_arbres = [100...800, 3]$,
$vitesse\ d'apprentissage = [-3...0, 5]_{\log_{10}}$,
$min\_samples\_leaf = [1...10, 3]$, $max\_depth = [3, 5]$

Table of Contents

Résultats

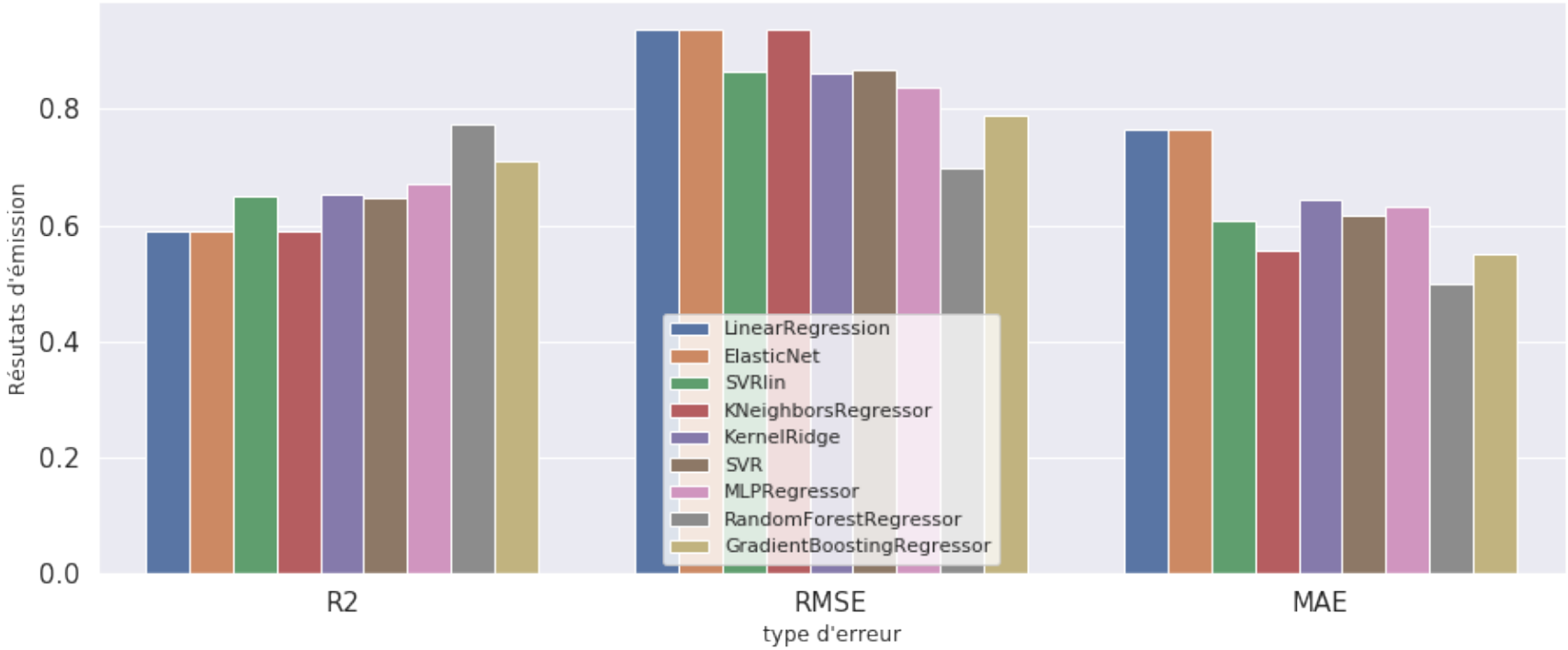
Les résultats calculés sur le set **d'entrainement** se décomposent comme suit :

- 2 études ont été réalisées : une **avec** (observations : 4965) et **sans** (observations : 6515) **ENERGYSTARScore**
- Pour chaque études 2 variables cibles : l'émission et la consommation.
- 3 métriques utilisées $R2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$, RMSE= $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$, MAE= $\frac{1}{n}\sum_{i=0}^{n-1}|y_i - \hat{y}_i|$.
- 9 modèles d'apprentissage supervisés
- 5 validations croisées pour chaque configuration **d'entrainement**.

Plan :

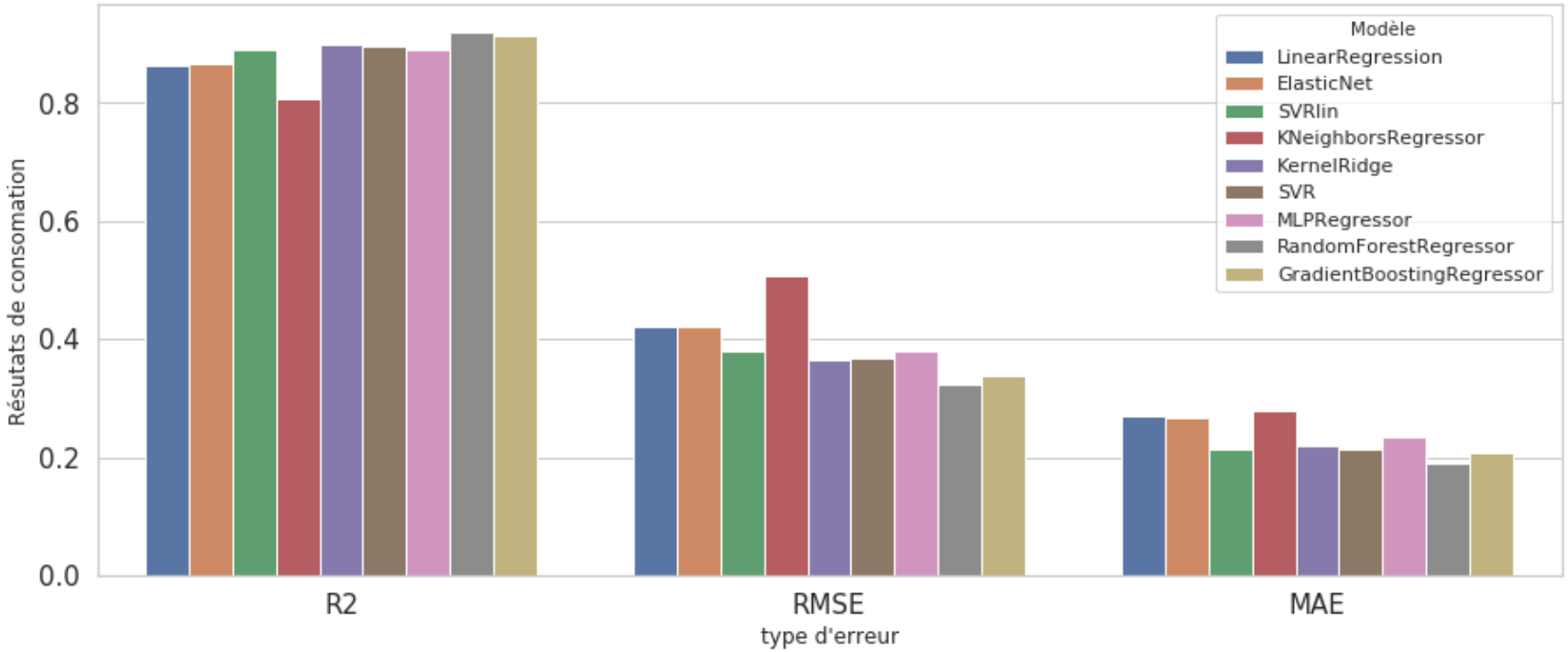
- Présentation des **108** calculs sous forme de 4 histogrammes
- Garder ou non **ENERGYSTARScore**

Emissions **AVEC** ENERGYSTARScore :



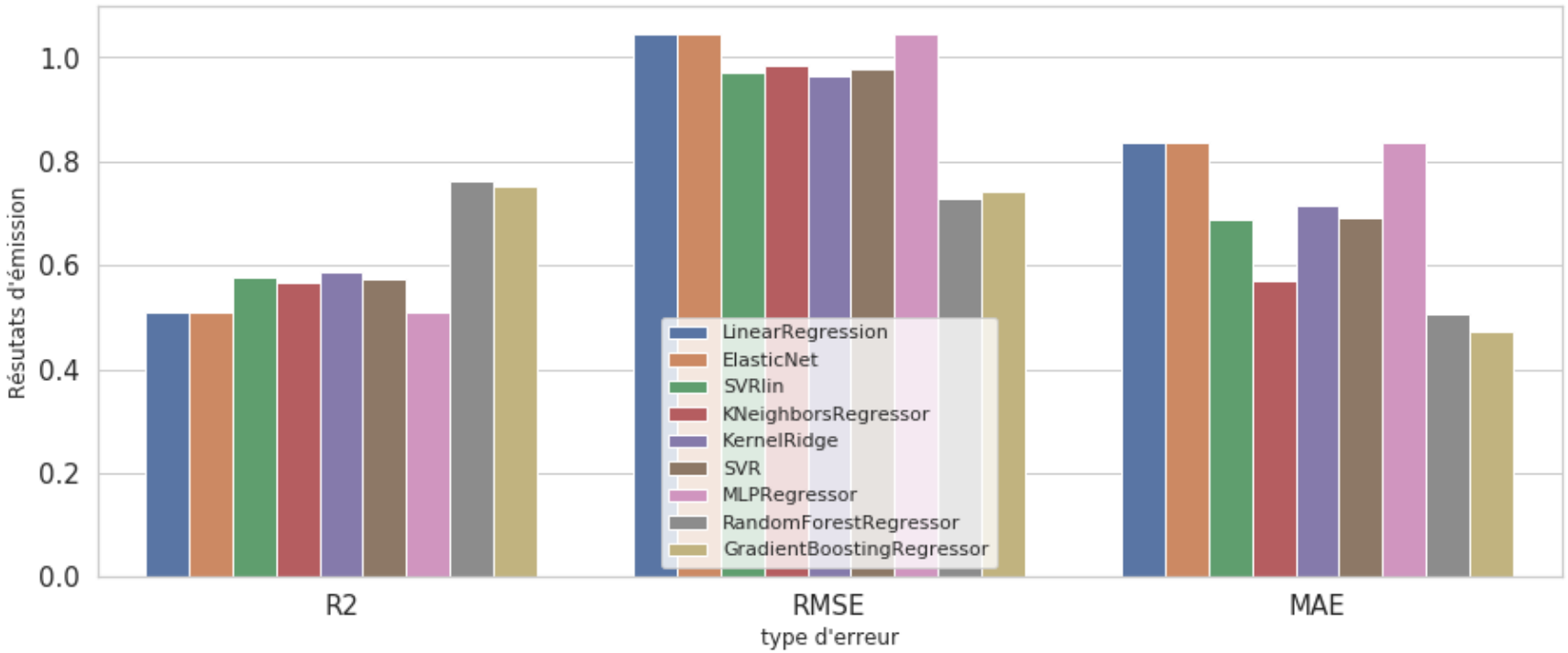
=> **Vainqueur : La forêt aléatoire !**

Consomation AVEC ENERGYSTARScore :



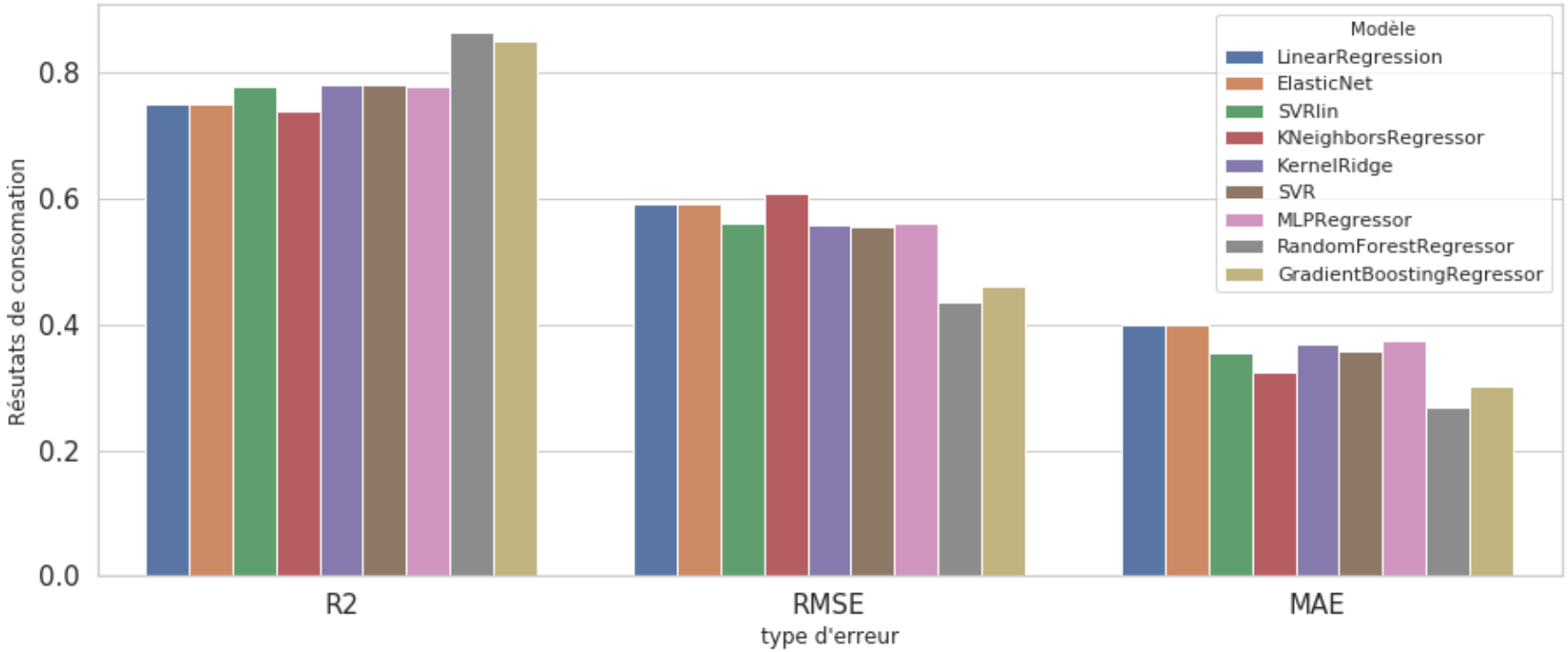
=> **Vainqueur : La forêt aléatoire !**

Emissions **SANS** ENERGYSTARScore :



=> **Vainqueur : Le Gradient boosting !** (mais tps CPU = 13.4 s / 3.78s pour les forets aléatoires)

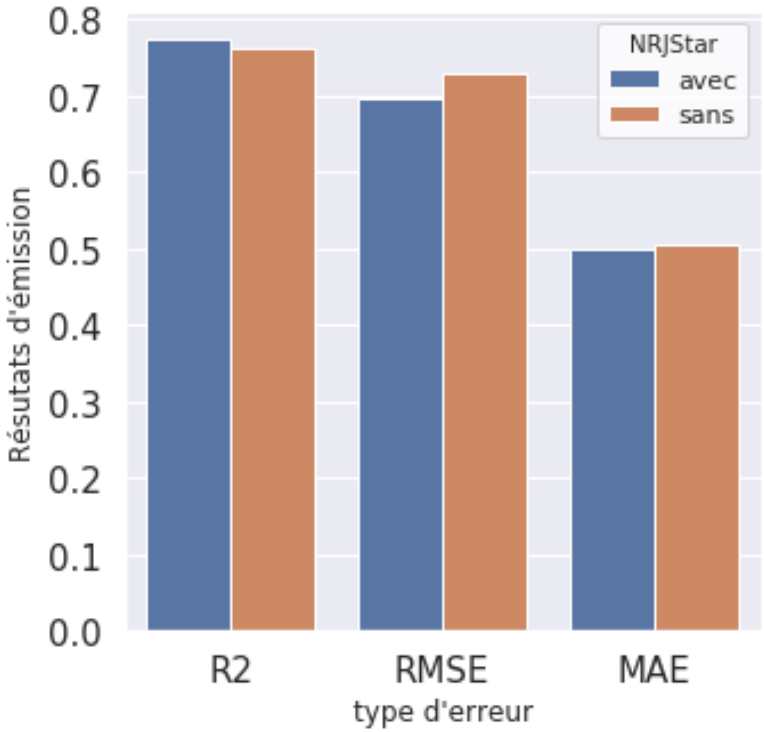
Consomation **SANS** ENERGYSTARScore :

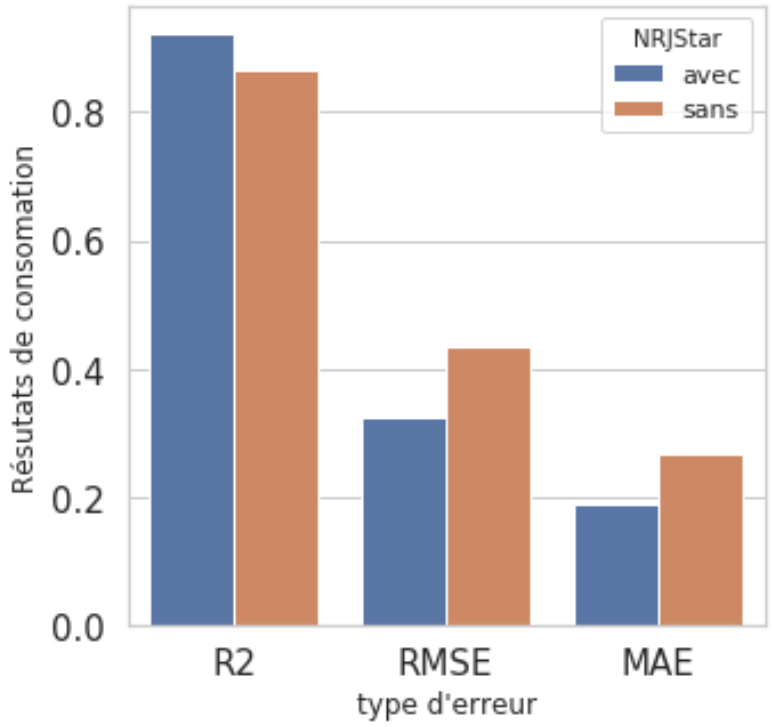


=> **Vainqueur : La forêt aléatoire !**

=> **Le modèle de forêts aléatoires est retenu**

Garder ou non ENERGYSTARScore



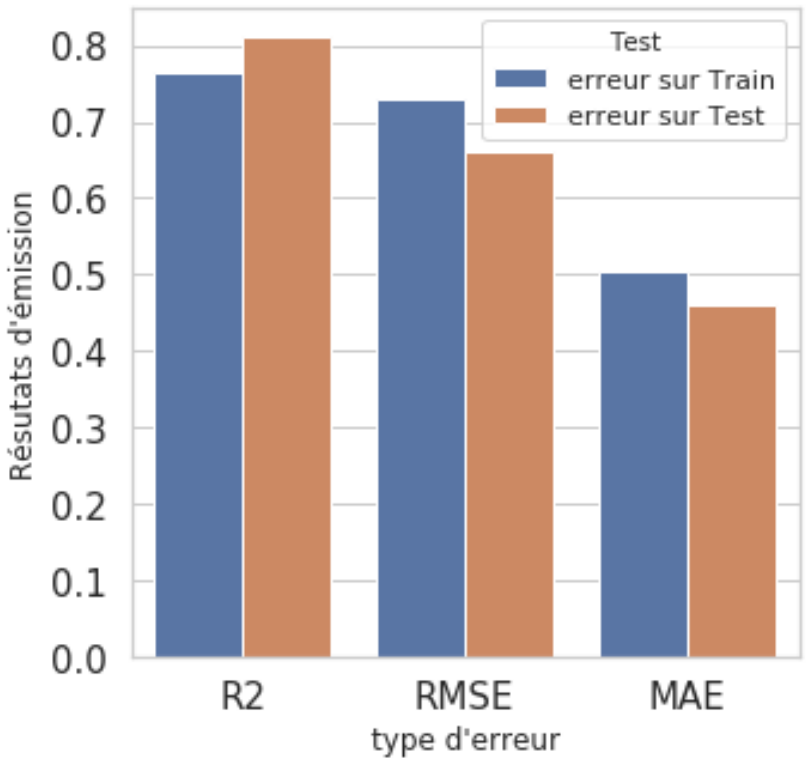


Résultats sans ENERGYSTARScore moins précis (-6% sur R2) mais satisfaisants

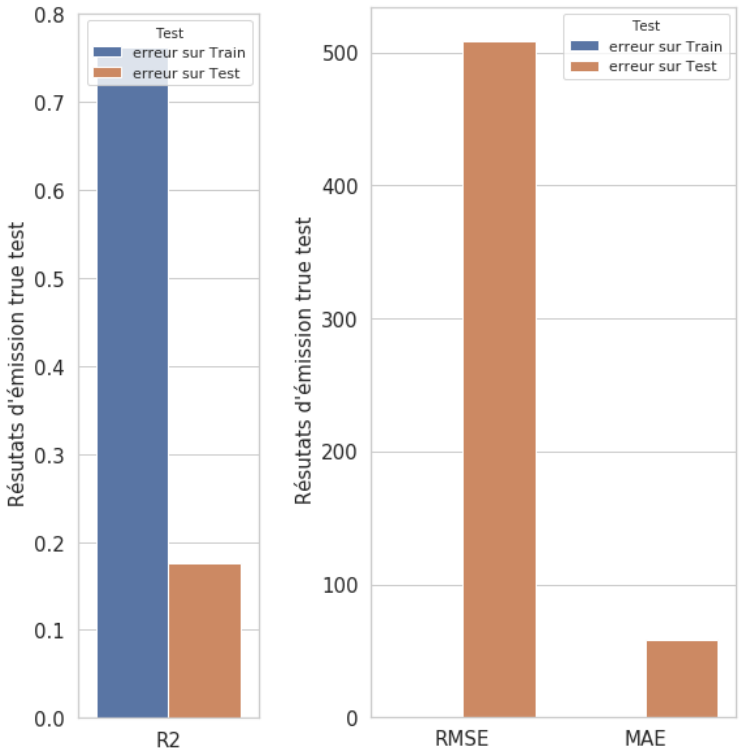
=> ENERGYSTARScore n'est pas retenue.

Erreur sur l'échantillon test : émissions

erreur sur log test



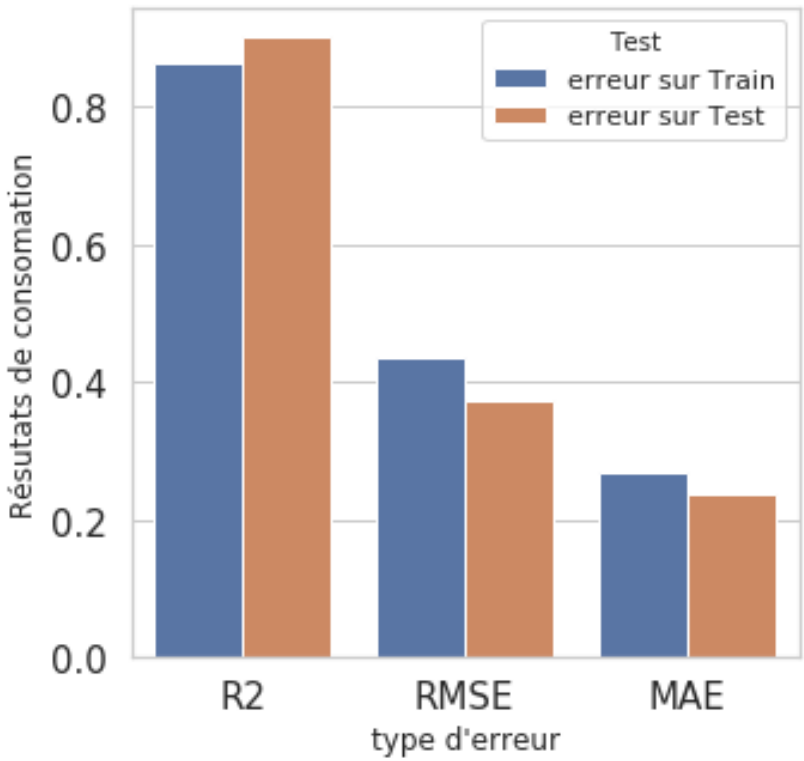
erreur sur true test



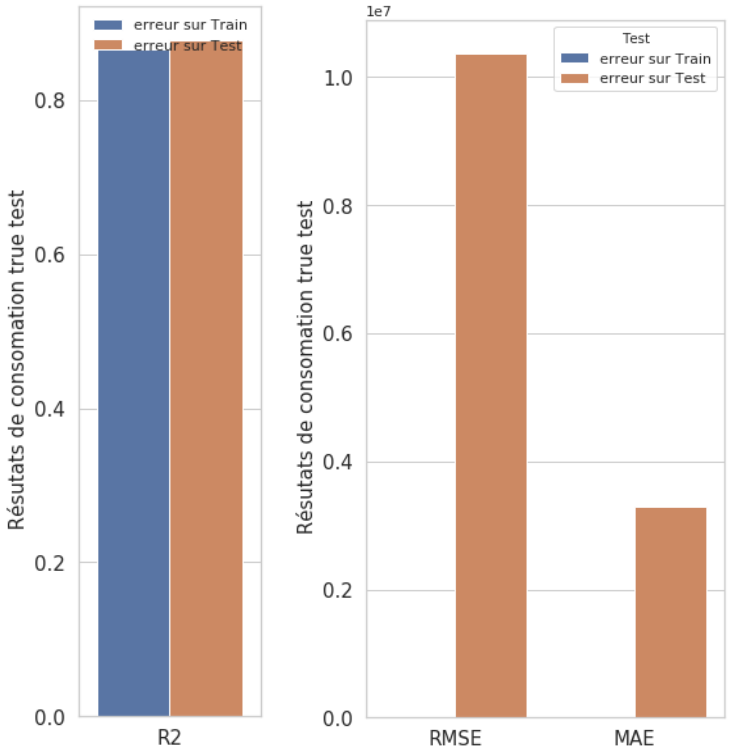
=> Scores sur log test sont légerement meilleur. Scores sur true test reflètent les vraies erreurs physique des prédictions.

Erreur sur l'échantillon test : consommation

erreur sur log test



erreur sur true test

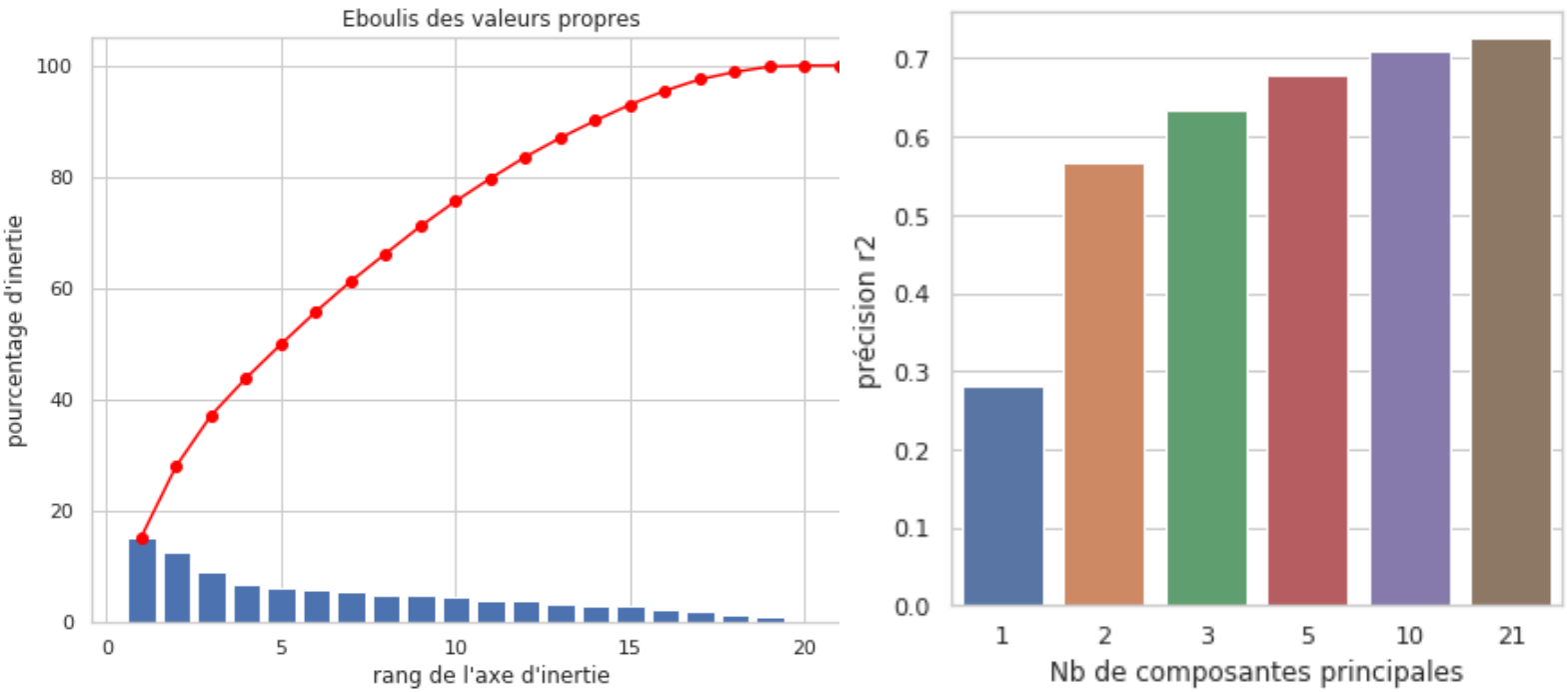


=> Scores sur log test sont légerement meilleur. Scores sur true test reflètent les vraies erreurs physique des prédictions.

Table of Contents

Analyse par composante principales sur les émissions

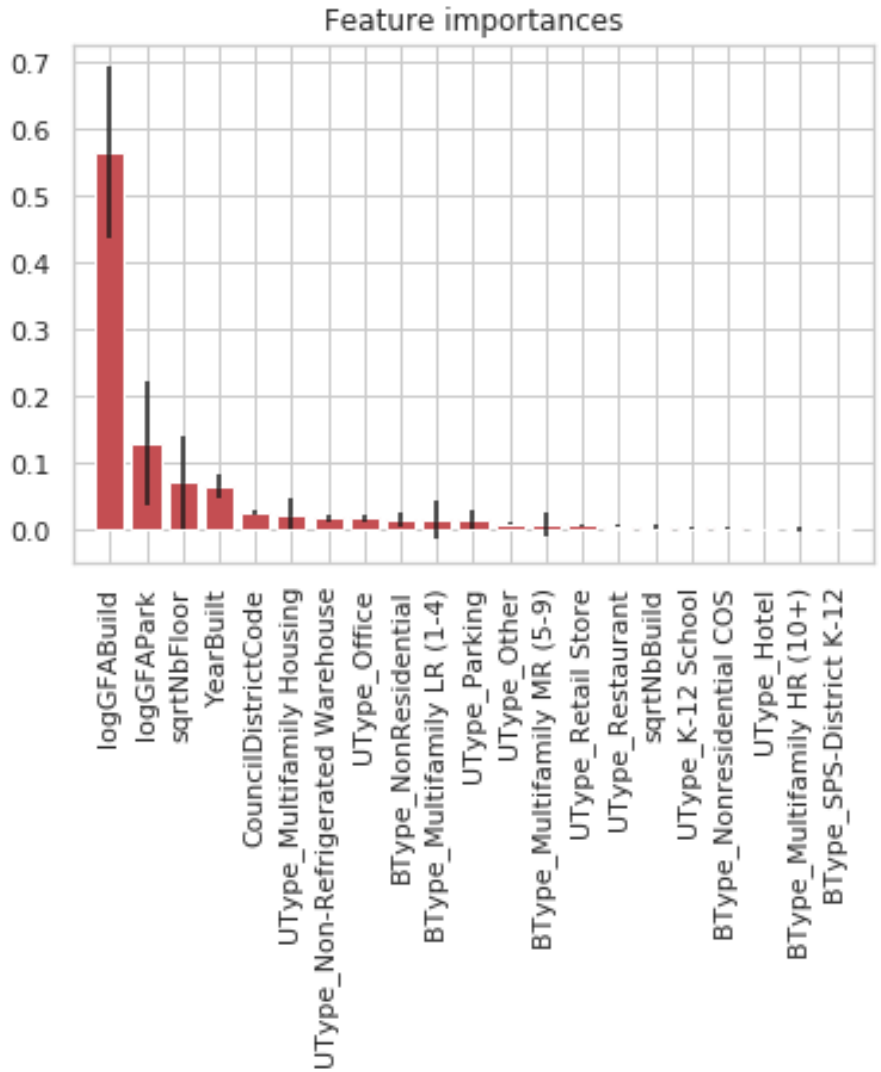
- Avantage : Evaluer l'importance des composantes principales
- Inconvénient : l'ACP ne selectionne pas directement les variables initales



=> Une analyse avec uniquement les 2 axes principaux donne un score r2 de **57%** (-16% / 73%).

Selection de variables par les forêts aléatoires sur la concomation

- Avantage : Evalue l'importance des variables sur [0,1]. Importance <=> Positions dans les arbres
- Inconvénient : les variables catégorielles sont éclatées



Selection forêt :

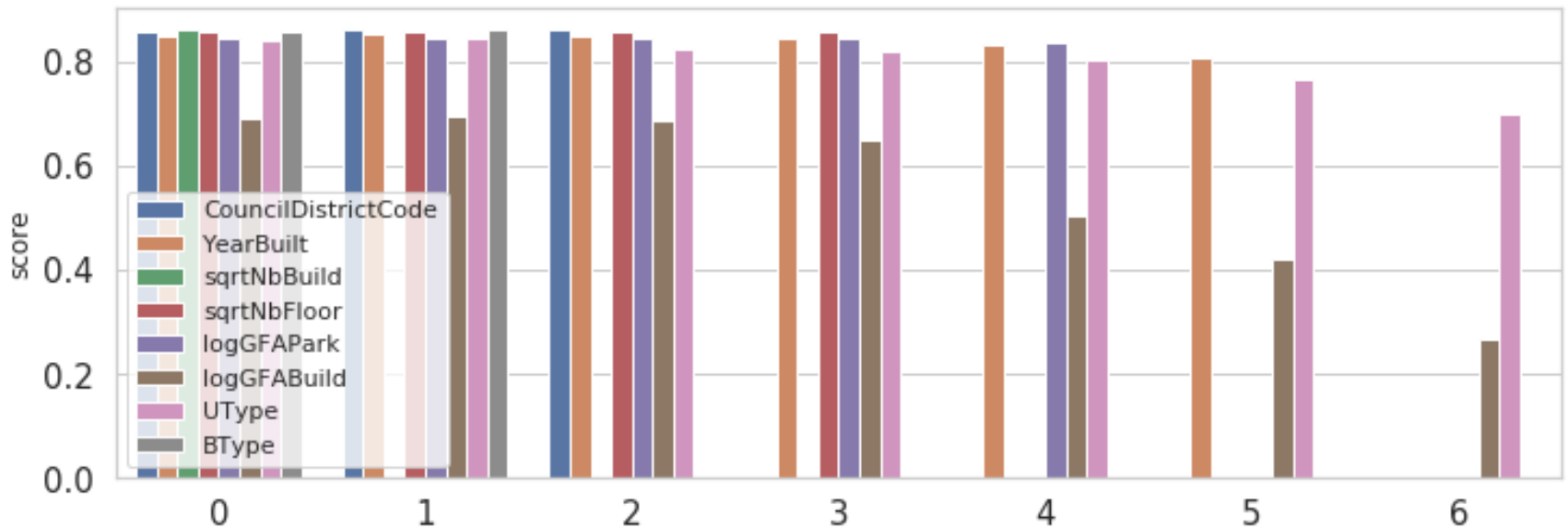
=> **GFABuild + GFAPark** ont une importance cumulée de **69%**

=> Analyse GFABuild + GFAPark : **R2 = 75%** (-12% / 87%)

Algorithme de descente de selection de variables sur la consommation

- Avantage : Evalue l'importance des variables. Groupe les catégories des variables qualitatives en une seule variable.
- Inconvénient : peut être gourmant en tps de calcul

Score r2 pour chaque variable supprimée :



Selection descente de variable :

=> **GFABuild+Utype** sont les variables les plus importantes.

=> Analyse GFABuild + Utype :
R2 = 81% (-6% / 87%)

Moyenne du BestScore-ScoreVariable sur toute les itérations :

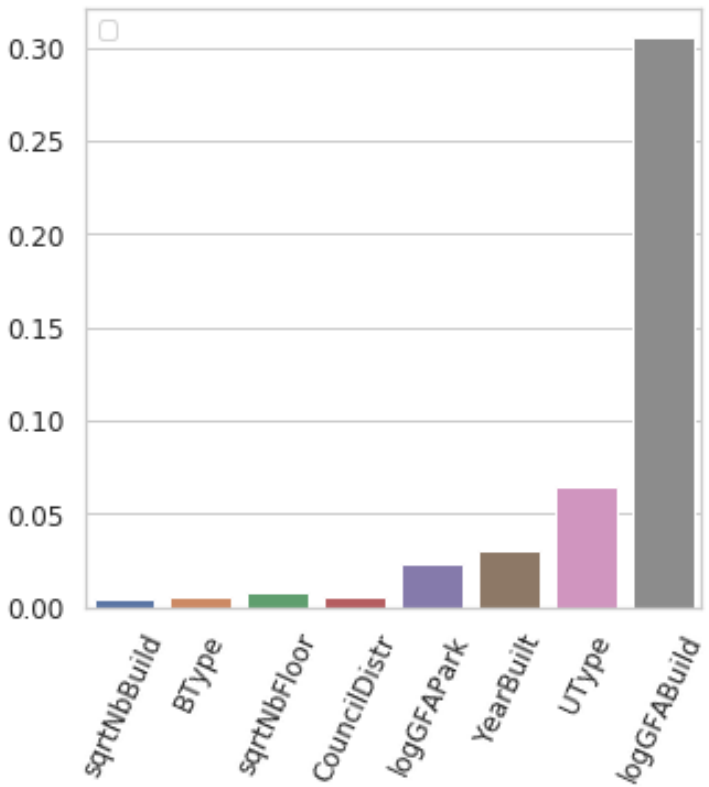


Table of Contents

Conclusion

- Feature Engineering : selection et transformation de :
  - 2 variables cibles pour l'énegie et la consommation
  - 7 variables quantitatives
  - 4 variables qualitatives
- Présentation des 9 modèles de régression supervisés
- Résultats

  => Vainqueur : modèle des forêts aléatoires

  => Le score ENERGYSTAR peut être écarté sans trop pénaliser les prédictions (-6% sur R2)

  => Les scores sur le jeu test sont légerement meilleurs que sur le train (+5% sur R2)
- Selection de variables
  - ACP :
    - \+ Une analyse avec seulement les 2 premières composantes principales donne un bon score r2 : 57% (-16% / 73%).
    - – l'ACP ne selectionne pas directement les variables initales.

- Selection des forêts :
  - \+ **GFABuild+YearBuild** ont une importance cumulée de **60%**
  - \+ Analyse **GFABuild+GFAPark** sur consommation : R2 = 75% (-12% / 87%).
  - – Les variables qualitatives sont éclatées (OHE)
- Algo de descente :
  - \+ Variables qualitatives groupées
  - \+ **GFABuild+Utype** sont selectionnées
  - \+ Analyse **GFABuild+Utype** sur consommation : R2 score = 81% (- 6% / 87%).

=> Le modèle des forêts aléatoires pourra être utilisé avec les variables GFABuild et UseType pour prédire la consommation.

Perspectives

- Appliquer ce modèle aux prochaines données terrain.
- Etudier ensuite l'évolution des émissions et consommations suivant la surface au sol ou le type d'usage des batiments

Merci de votre attention