Anticipez le besoins en consommation de bâtiments

Projet 4: Moustapha ABDELLAHI - Openclassrooms

Sommaire

- **Problématique**
- **Présentation des données**
- **Préparation des données**
- **Modélisation et Optimisation**
- **Conclusions**

PROBLEMATIQUE DE LA VILLE DE SEATTLE

- **Objectif : Ville neutre en émissions de carbone en 2050**

- **Données de consommation:**

- **Relevés annuels minutieux, effectués en 2015 et 2016, qui sont coûteux à obtenir.**

https://s3.eu-west-1.amazonaws.com/course.oc-static.com/project/Data_Scientist_P4/2016_Building_Energy_Benchmarking.csv

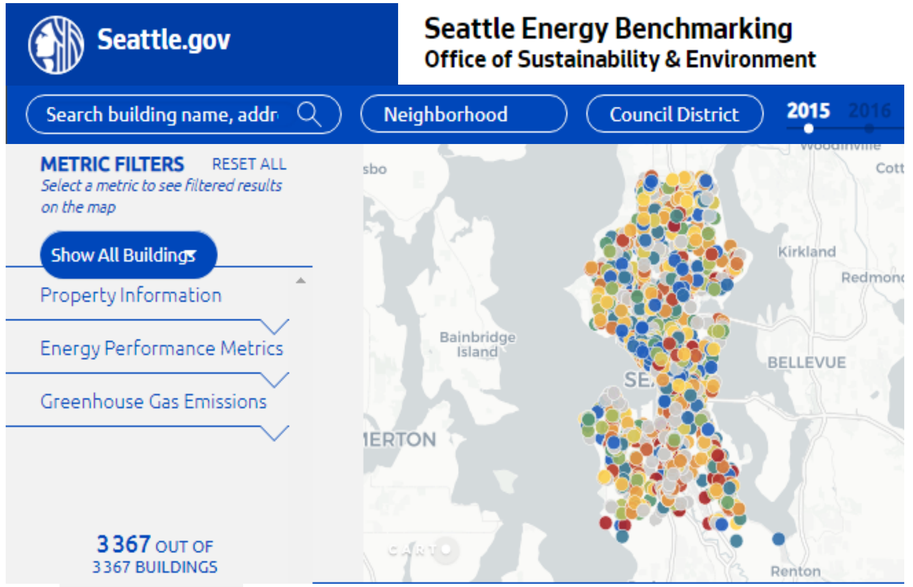
- **Mission :**

- **Utiliser ces données pour prédire les émissions de CO2 et la consommation totale d'énergie.**

- **Evaluer l'intérêt de l'indicateur ENERGY STAR Score**

- **Proposer un modèle de prédiction réutilisable**

ANALYSE DE LA PROBLEMATIQUE



- **Features considérées :**
- **Caractéristiques des bâtiments sans**

la consommation.

- **Targets à prédire:**
- **Consommation totale des bâtiments SiteEnergyUse(kBtu)**
- **Emissions des bâtiments GHGEmissions(MetricTonsCO2e)**

PRESENTATION DES DONNEES

- **2 Datasets séparés de dimension différentes et non alignés (des colonnes différentes et certaines informations sont présentées différemment)**
    - **3 catégories de features : géographiques, caractéristiques et énergétiques**
- **Dataset 2015 :**
- **3340 lignes (établissements)**
- **47 features**

- **Dataset 2016**
- **3376 lignes (établissements)**
- **46 features**
    - **9 features non communes**

PREPARATION DES DONNEES

Harmonisation

- **Harmonisation des variables (noms et modalités)**
- **Fusion des 2 datasets pour créer 1 nouveau dataset.** **dim= (6716 lin., 46 col.)**
- **Après sélection des bâtiments non résidentiels.** **dim= (3318 lin., 46 col)**

Nettoyage

- **Suppressions des valeurs aberrantes des features: (‘NumberofBuildings’, ‘NumberofFloors’, ‘PropertyGFAParking’, ‘PropertyGFABuilding(s)’, ‘Building_age’) > 0**

- **Dataset résultant** **dim= (3309 lin., 12 col.)**

Features Engineering

- **Prise en compte de la variable ENERGYSTARScore**
- **Suppression des NaN des variables targets**
- **Création d’une variable âge d'un bâtiment :**

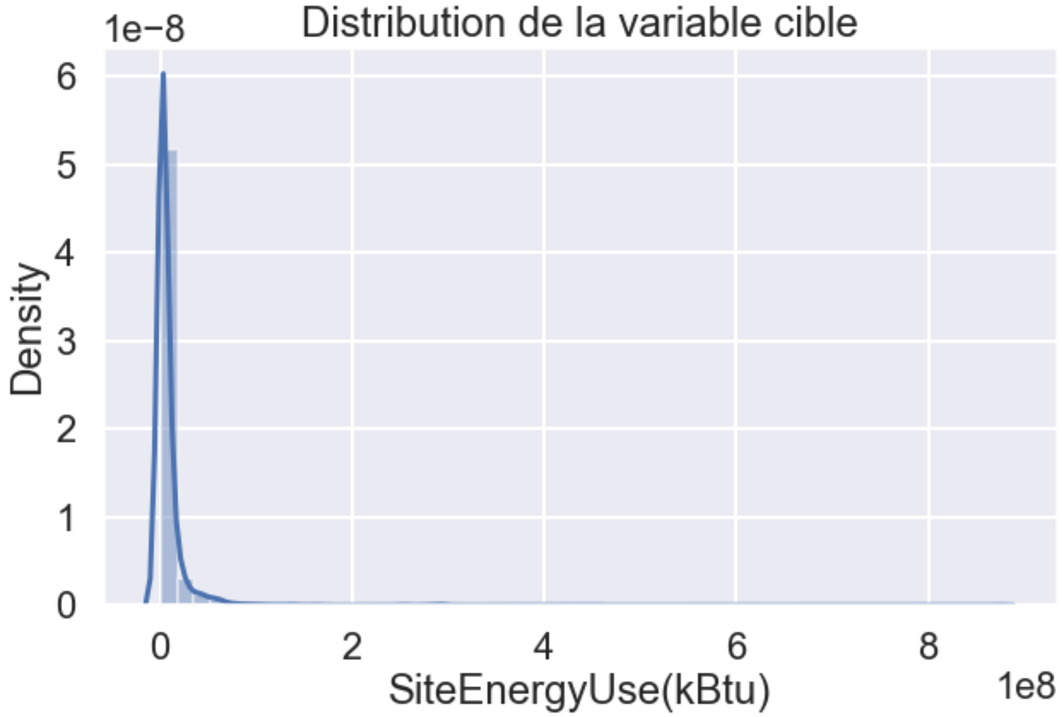
’Building_age’= 'DataYear' - ‘YearBuilt’

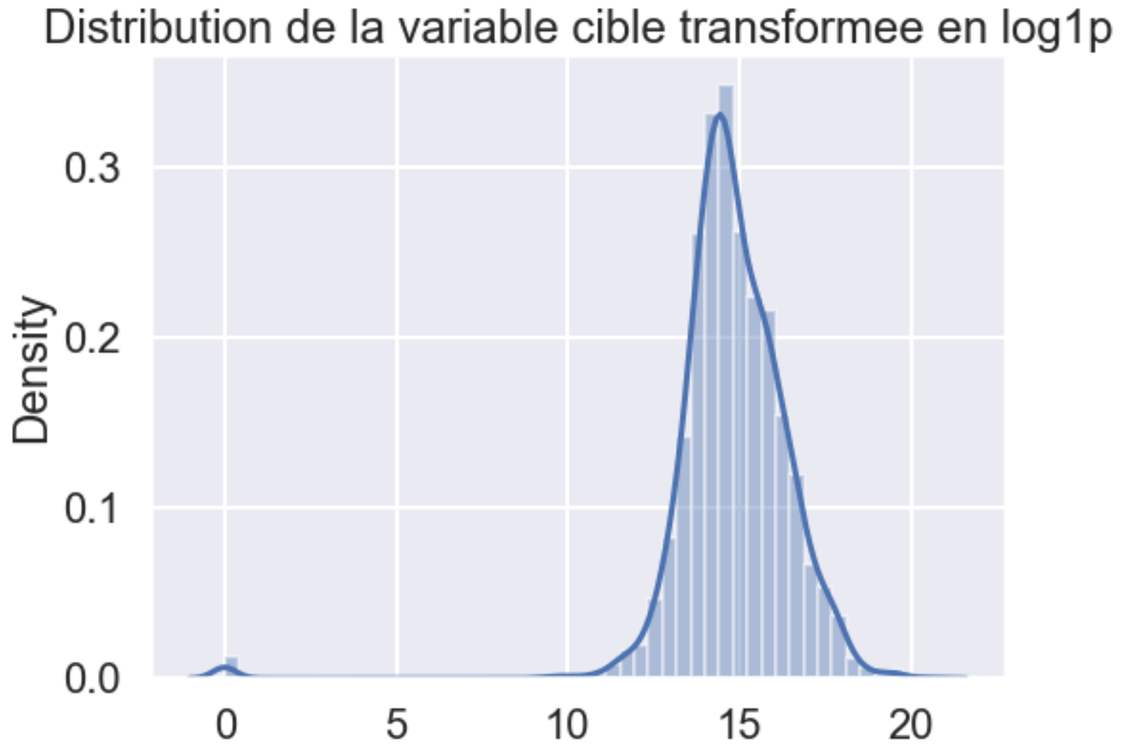
- **Choix des variables pertinentes**

BuildingType
PrimaryPropertyType
NumberofBuildings
NumberofFloors
ENERGYSTARScore
Latitude
Longitude
PropertyGFAParking
PropertyGFABuilding(s)
Building_age

Features Engineering

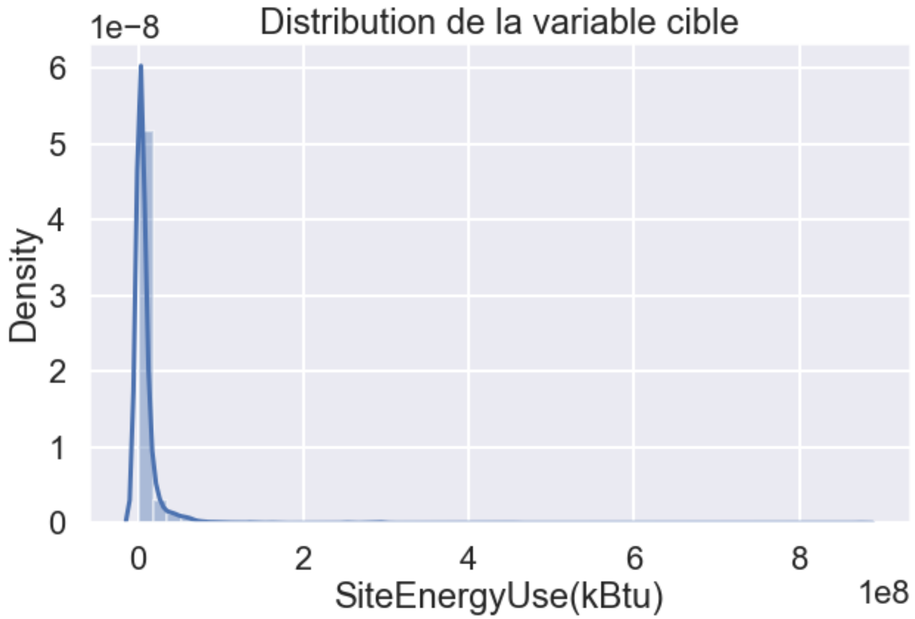
- **Normalisation des features numériques par StandardScaler( )**
- **Encodage des features catégorielles par OneHotEncoder**
- **Imputation des valeurs manquantes à l'aide de KNNImputer**
- **Log1p transformation de la variable target : SiteEnergyUse(kBtu)**
- **Dataset résultant dim= (3309 lin., 13 col.)**

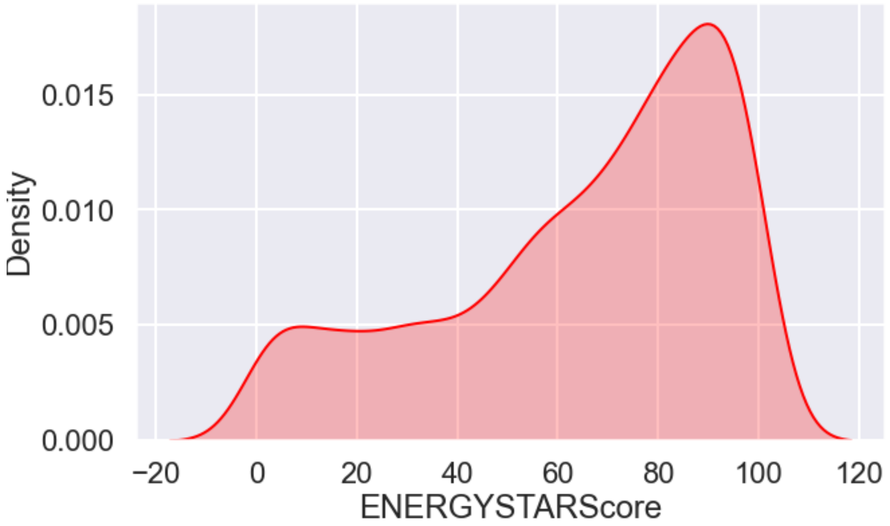


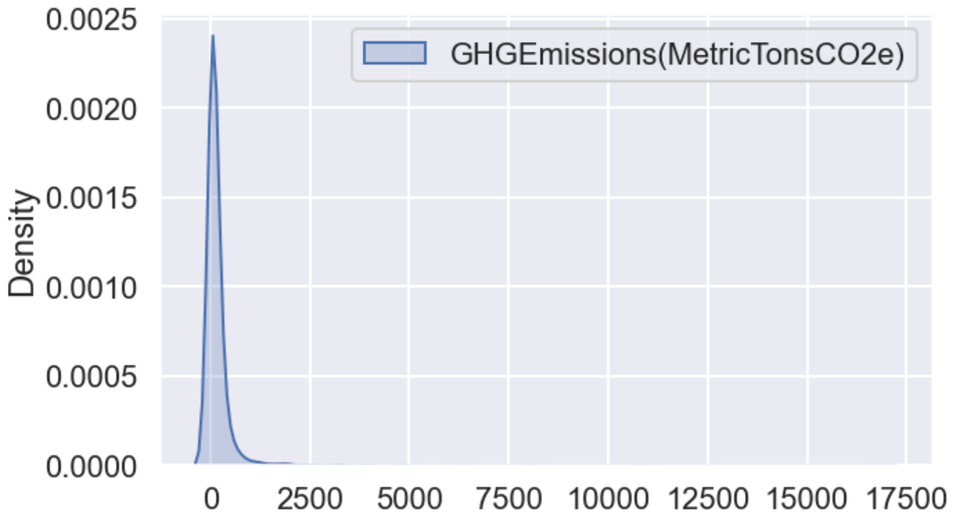


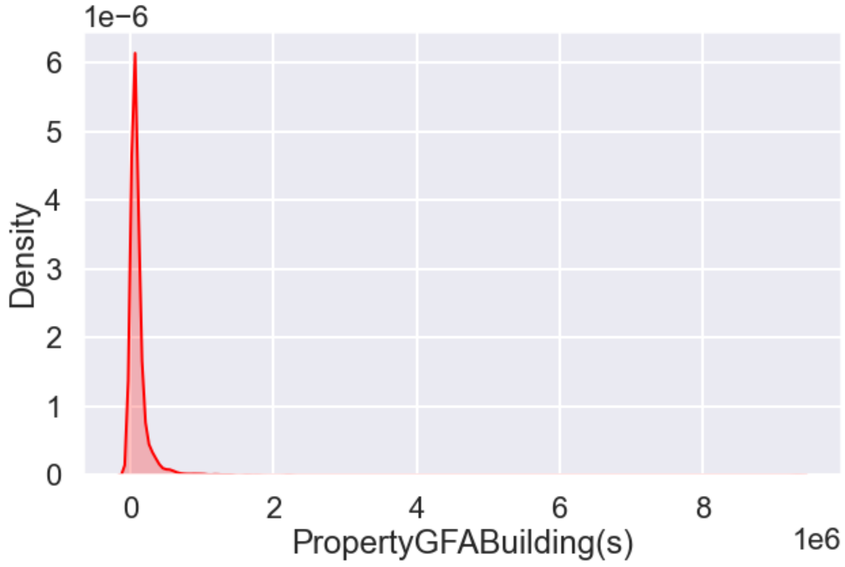
Analyse Exploratoire

- **Distributions des features et des targets non normales**
- **Présence d'outliers**



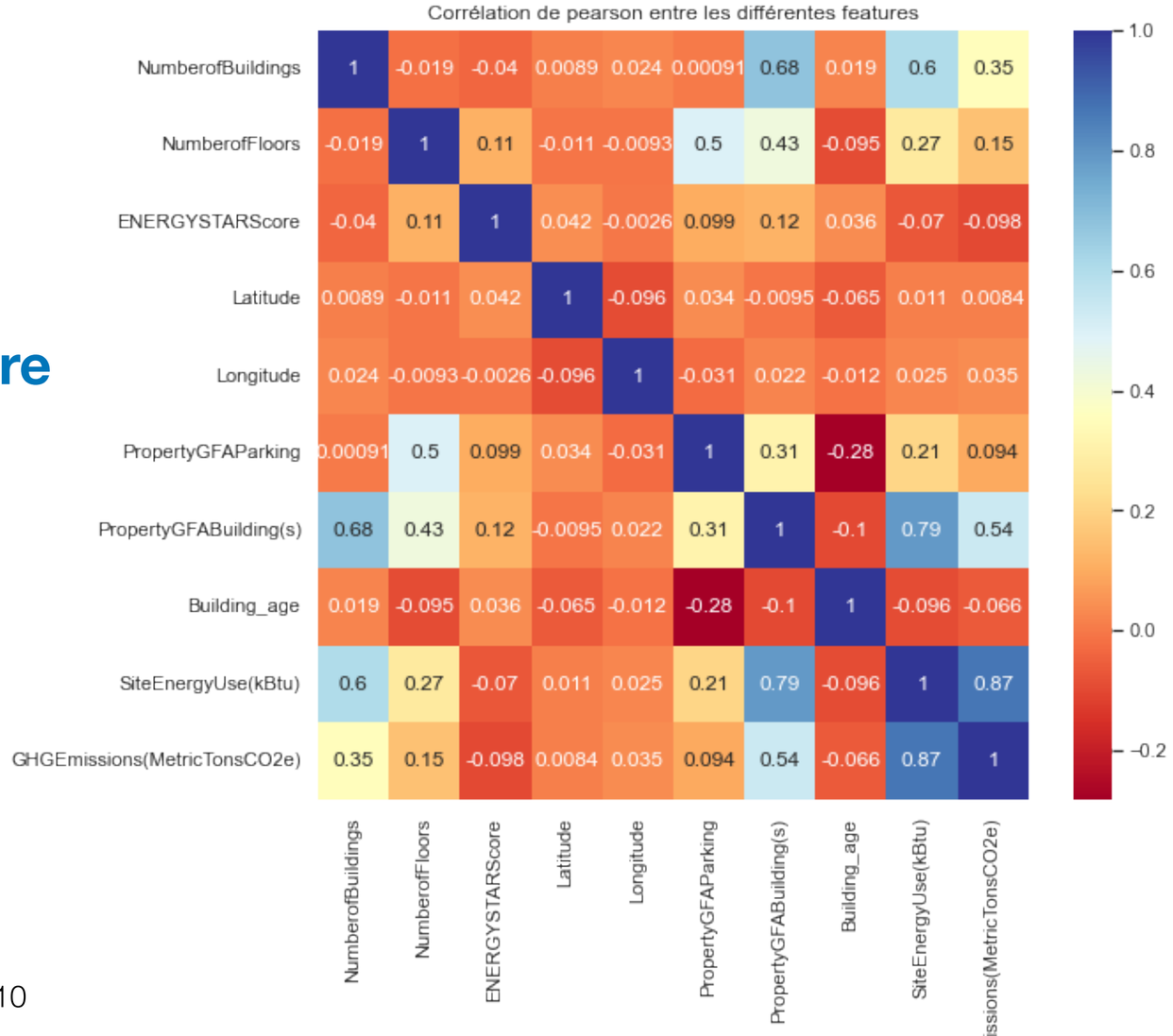






Analyse Exploratoire : corrélation

- **Corrélation importante entre SiteEnergyUse(kBtu) et les variables PropertyGFABuilding(s), NumberofBuildings**
- **Corrélation entre GHGEmissions(MetricTonsCO2e) et les variables PropertyGFABuilding(s), NumberofBuildings**
- **Corrélation forte entre targets SiteEnergyUse(kBtu) et GHGEmissions(MetricTonsCO2e)**
- **Pas de corrélation entre ENERGYSTARScore et les autres variables**



Algorithmes Linéaires

- **Régression Linéaire: Considérons des données, n points en dimension p, représentés par la matrice $X \in \mathbb{R}^{n \times p}$ et leurs étiquettes (cibles observées) à valeurs réelles données par $y \in \mathbb{R}^n$. Le but est de trouver une fonction linéaire $f : \mathbb{R}^p \to \mathbb{R}$**

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_p + \epsilon$$

dont le vecteur des coefficients $\beta = (\beta_0, \beta_1, \beta_2, \ldots, \beta_p)$, $\hat{y}$ la valeur prédite et $\epsilon = y - \hat{y}$ est l'erreur, pour minimiser la somme des carrés des erreurs entre les cibles observées dans l'ensemble de données et les cibles prédites par l'approximation linéaire. Mathématiquement, en utilisant la méthode des moindres carrés, ce problème est formulé ainsi :

$$\min_{\beta} \|\beta X - y\|_2^2$$

- **Ridge: une régression linéaire régularisée avec une contrainte sous forme de norme quadratique (L_2) des paramètres (coefficients) du modèle, qui permet d'agir sur ces paramètres à l'aide d'un hyperparamètre $\alpha \geq 0$. La fonction coût:**

$$\min_{\beta} \|\beta X - y\|_2^2 + \alpha \|\beta\|_2^2$$

Algorithmes Linéaires

- **Lasso : une régression linéaire régularisée avec une contrainte sous forme de norme en valeur absolue (L_1) des paramètres du modèle, dont l'effet est de choisir les variables les plus importants pour le modèle à l'aide d'un hyperparamètre $\alpha \geq 0$**

- **ElasticNet : est une régression qui combine Ridge et Lasso pour tirer les avantages de chaque méthodes.**

ALGORITHMES DE MODELISATION

Algorithmes Ensemblistes

- **Le principe des méthodes ensembliste est d'obtenir un modèle prédictif performant en combinant des modèles de performances faibles. C'est l'effet Wisdom of Crowds.**
- **Random Forest Regressor: Un algorithme utilisant des arbres de décision de classification sur divers sous-échantillons de l'ensemble de données et prenant la moyenne pour améliorer la précision prédictive en contrôlant le overfitting. La taille d'un sous-échantillon est contrôlée avec le paramètre max_samples si bootstrap=True.**
- **Gradient Boost Regressor: Un algorithme qui donne un modèle de prédiction sous la forme d'un ensemble de modèles de prédiction faibles, qui sont généralement des arbres de décision. Il se base sur trois éléments:**

- Une fonction coût à optimiser.
- Un apprenant (prédicteur) faible pour faire des prédictions.
- Un modèle additif pour ajouter des prédicteurs faibles afin de minimiser la fonction coût.
-

ALGORITHMES DE MODELISATION

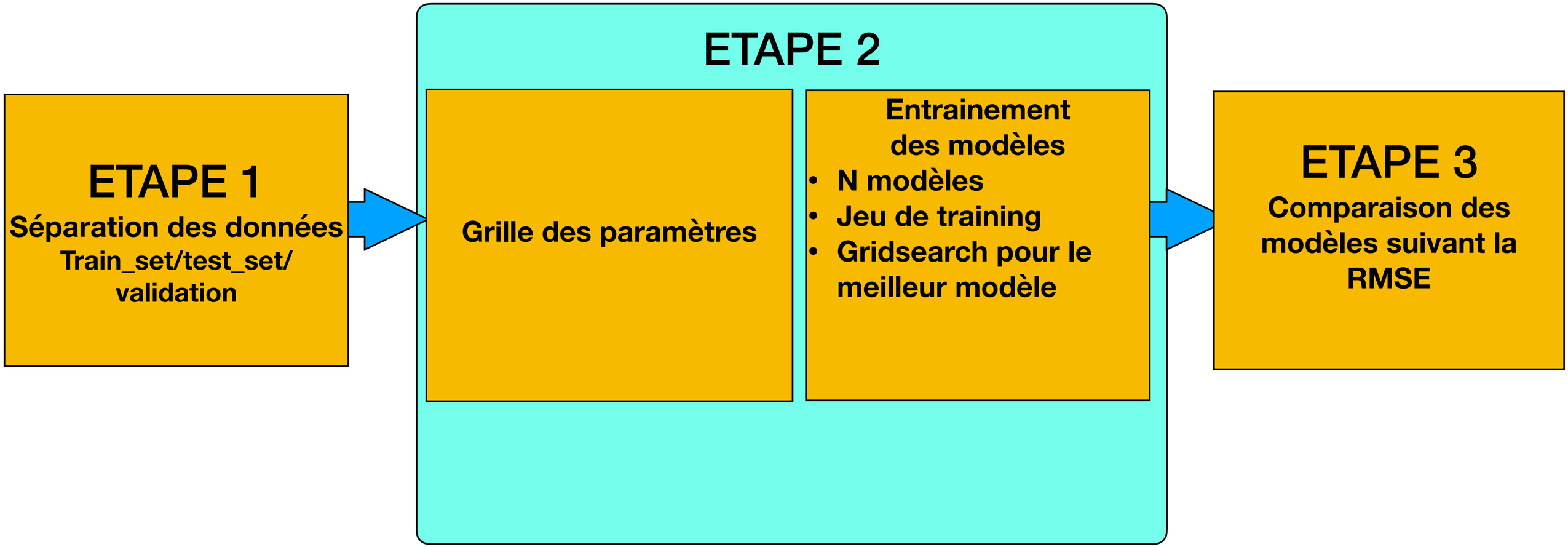
Algorithmes Ensemblistes

- **Bagging : (Bootstrap aggregating), est un méta-algorithme d'ensemble conçu pour améliorer la stabilité et la précision des algorithmes de machine learning utilisés dans la régression et la classification. Le principe est de créer un dataset différent pour entrainer chacun des différents apprenants à l'aide du bootstrap (échantillonner avec replacement notre dataset de taille N afin d'obtenir un nouveau dataset de taille N). Ensuite , il fait une prédiction en effectuant la moyenne pour la régression ou un vote à la majorité pour la classification.**
-

Support Vector Machine

- **SVM : une famille d'algorithmes de machine learning développés dans les années 1990. Son principe est de séparer les données en classes à l'aide d'une frontière, de telle façon que la distance (marge) entre les différents groupes de données et la frontière qui les sépare soit maximale. Les vecteurs de support sont les données les plus proches de la frontière.**

MODELISATION ET OPTIMISATION



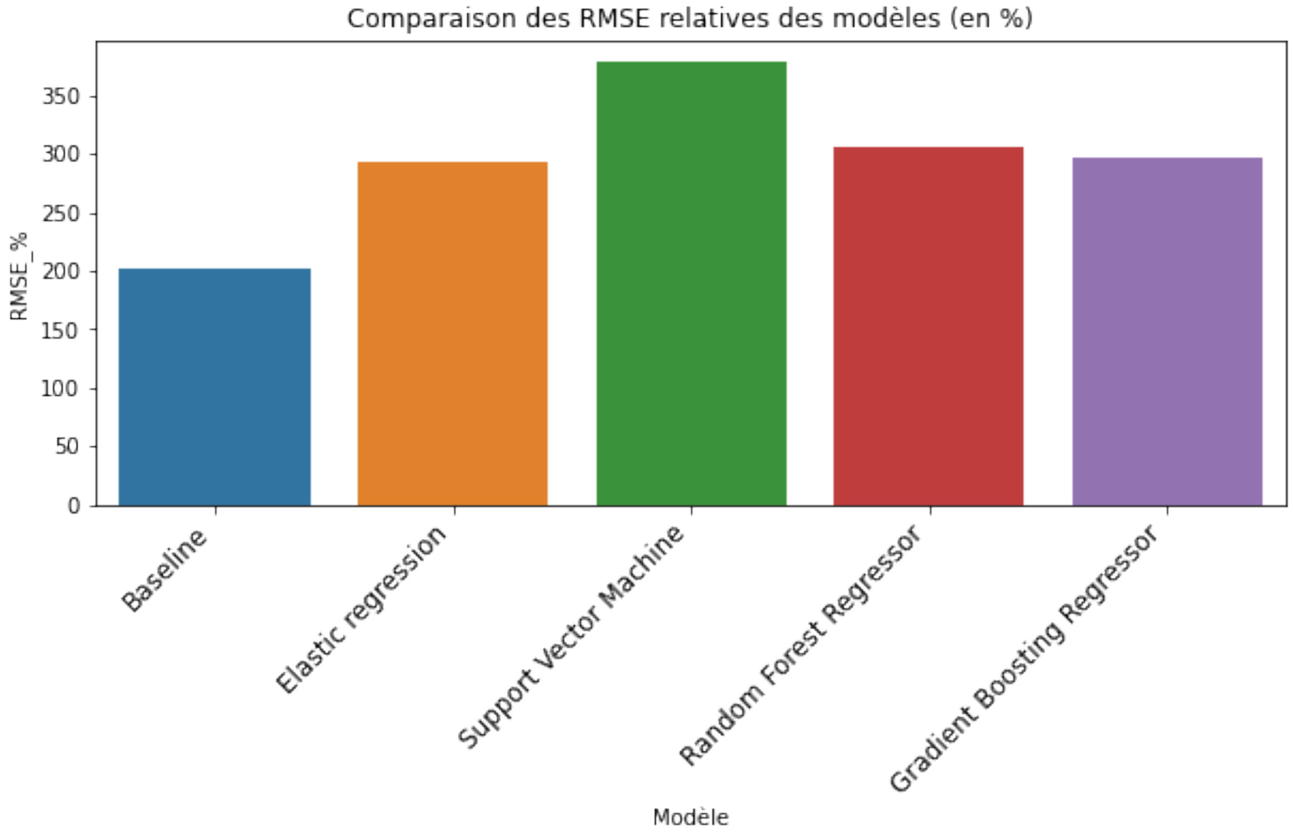
Paramètres du modèle de consommation

ElasticNet Regression	Random Forest Rgressor	SVM	Gradient Boost Regressor
'tol' : [0.01,0.001,0.0001]	n_estimators' : [10,50,100]	gamma' : [1e-3, 1e-1,10]	learning_rate': [0.02,0.03,0.04],
"alpha": [0.001, 0.01, 0.1, 1, 10, 100]	min_samples_leaf' : [1,3,5]	epsilon' : [0.001, 0.01, 0.1]	subsample' : [0.9, 0.5, 0.1],
"l1_ratio": np.arange(0.0, 1.0,0.1)	max_features': ['auto', 'sqrt']	C' : [0.01, 0.1, 10]	n_estimators' : [500,1000, 1500]
			max_depth' : [6,8,10]

Comparaison des modèles de Consommation:

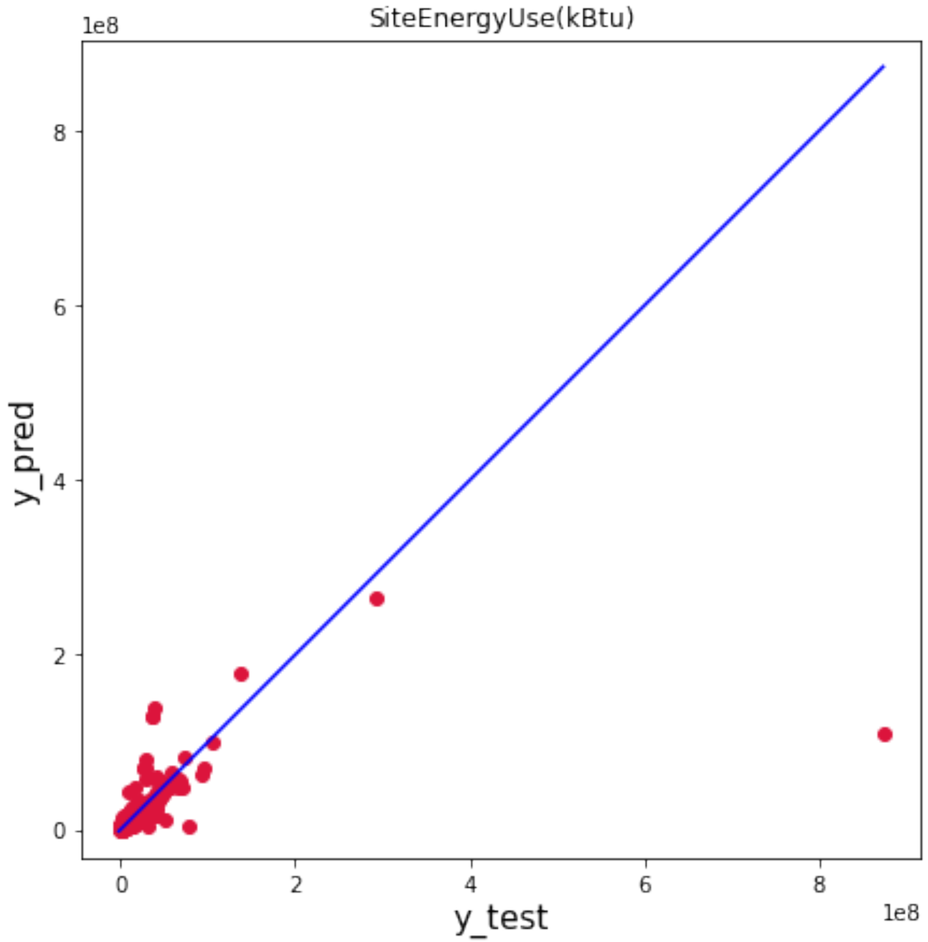
- **La variable target :** **SiteEnergyUse(kBtu)**
- **Baseline : DummyRegressor dont la strategy='mean'**
- **Baseline : RMSE est la plus petite mais le modèle n'est pas bon car R2 <0.**
- **Le meilleur modèle en termes de RMSE et de R2 est ElasticNet regression.**

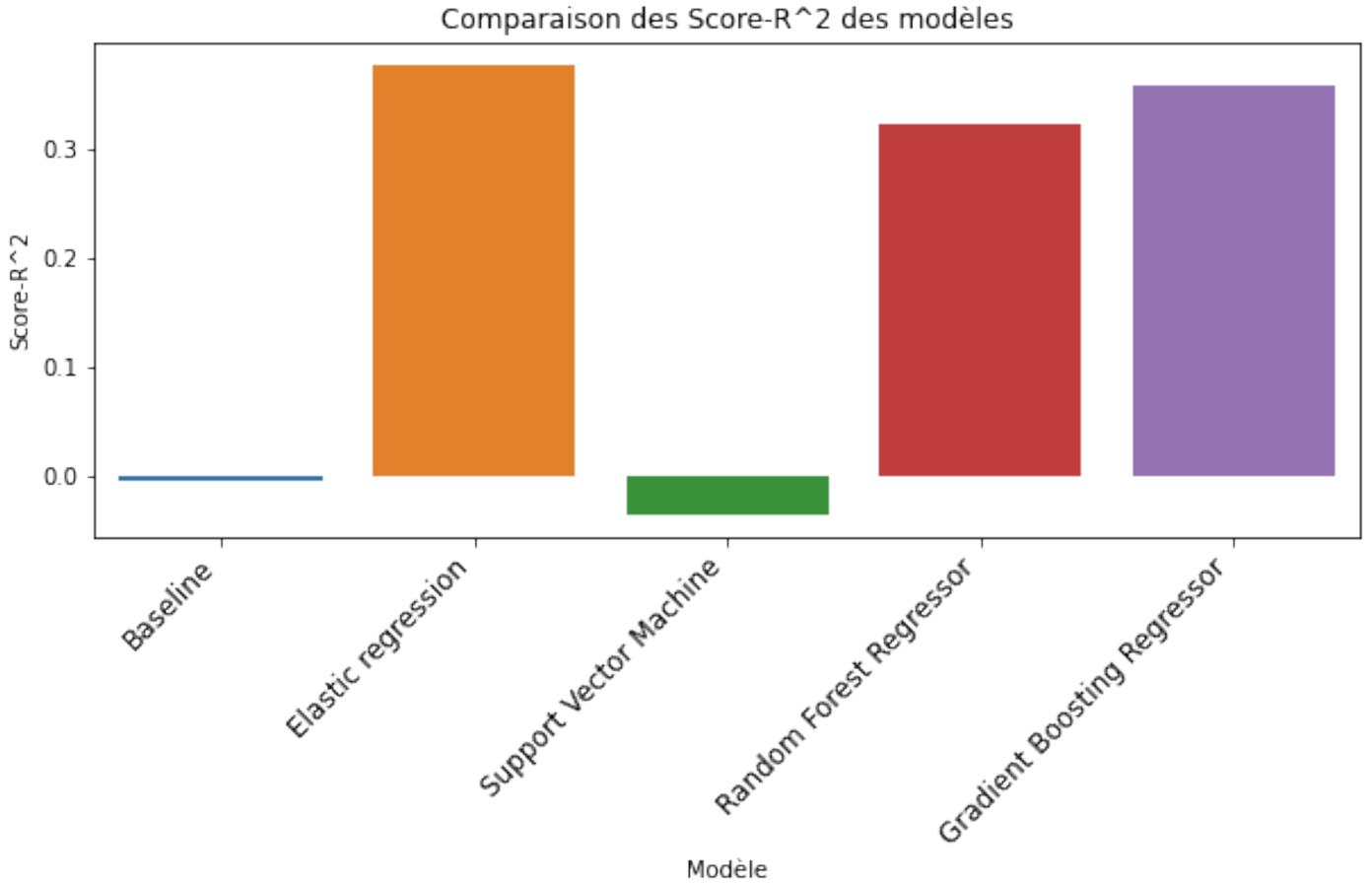
Modèle	Score-R^2	Score_RMSE	RMSE_%
Baseline	-0.003807	1.642956e+07	2.020647
Elastic regression	0.377182	2.385408e+07	2.933779
Support Vector Machine	-0.034897	3.074898e+07	3.781772
Random Forest Regressor	0.322070	2.488711e+07	3.060829
Gradient Boosting Regressor	0.358845	2.420270e+07	2.976654



Comparaison des modèles de Consommation:

- **Modèle ElasticNet : la prédiction est proche de la valeur réelle**
- **Le coefficient R2 de ElasticNet est légèrement plus grand que celui de Gradient Boosting Regressor.**





Comparaison des modèles de Consommation:

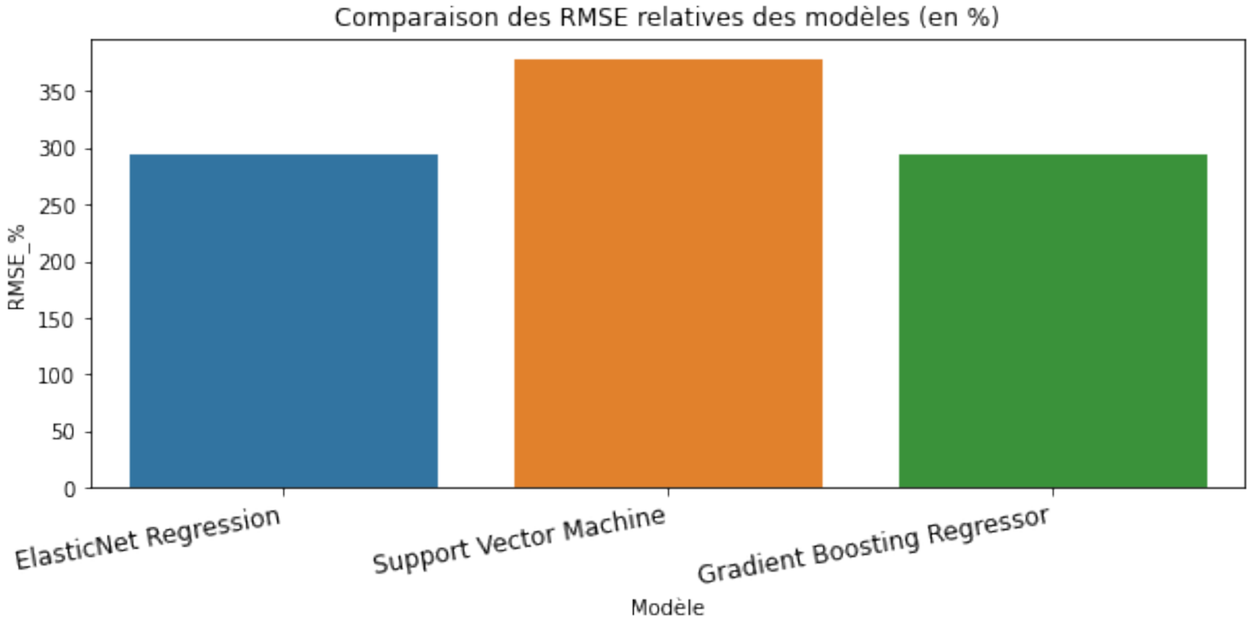
- **La variable target : Log1p(SiteEnergyUse(kBtu))**
- **Le meilleur modèle selon la RMSE est ElasticNet**

Modèle	Score-R^2	Score_RMSE	RMSE_%
ElasticNet Regression	0.377182	2.385408e+07	2.933779
Support Vector Machine	-0.034897	3.074898e+07	3.781772
Gradient Boosting Regressor	0.377364	2.385060e+07	2.933350

```
ElasticNet Regression :
87.4 µs ± 3.23 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)

SVM :
747 ms ± 25.4 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

Gradient Boosting Regressor :
50.9 ms ± 385 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```



Modèles d'émission: avec ENERGYSTARScore

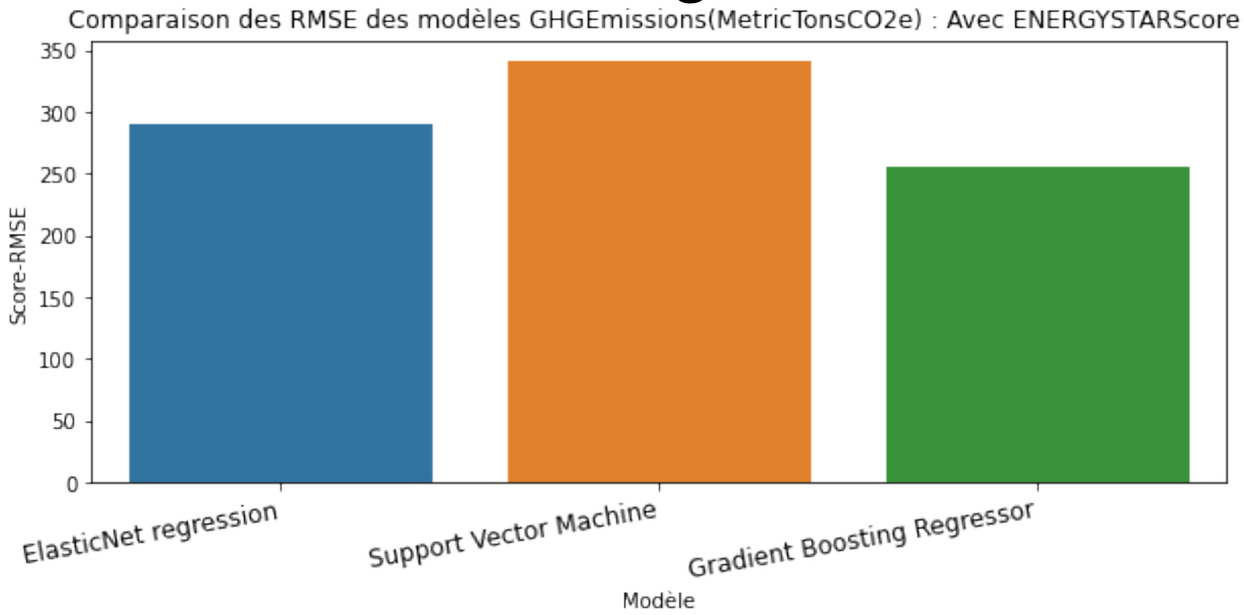
- **La variable target : GHGEmissions(MetricTonsCO2e)**
- **Le meilleur modèle selon la RMSE est Gradient Boot Regressor**

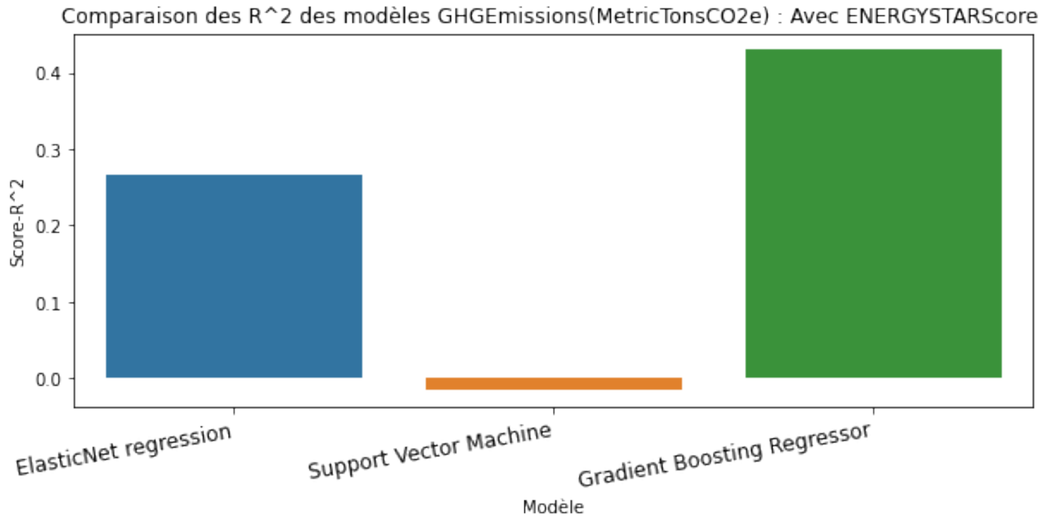
Modèle	Score-R^2	Score-RMSE
ElasticNet regression	0.266380	289.961681
Support Vector Machine	-0.015379	341.129420
Gradient Boosting Regressor	0.429594	255.680257

```
ElasticNet Regression :
93.8 µs ± 6.78 µs per loop (mean ± std. dev. of 7 runs, 10,000
loops each)

SVM :
1.26 s ± 19.9 ms per loop (mean ± std. dev. of 7 runs, 1 loop
each)

Gradient Boosting Regressor :
28.1 ms ± 1.84 ms per loop (mean ± std. dev. of 7 runs, 10 loops
each)
```

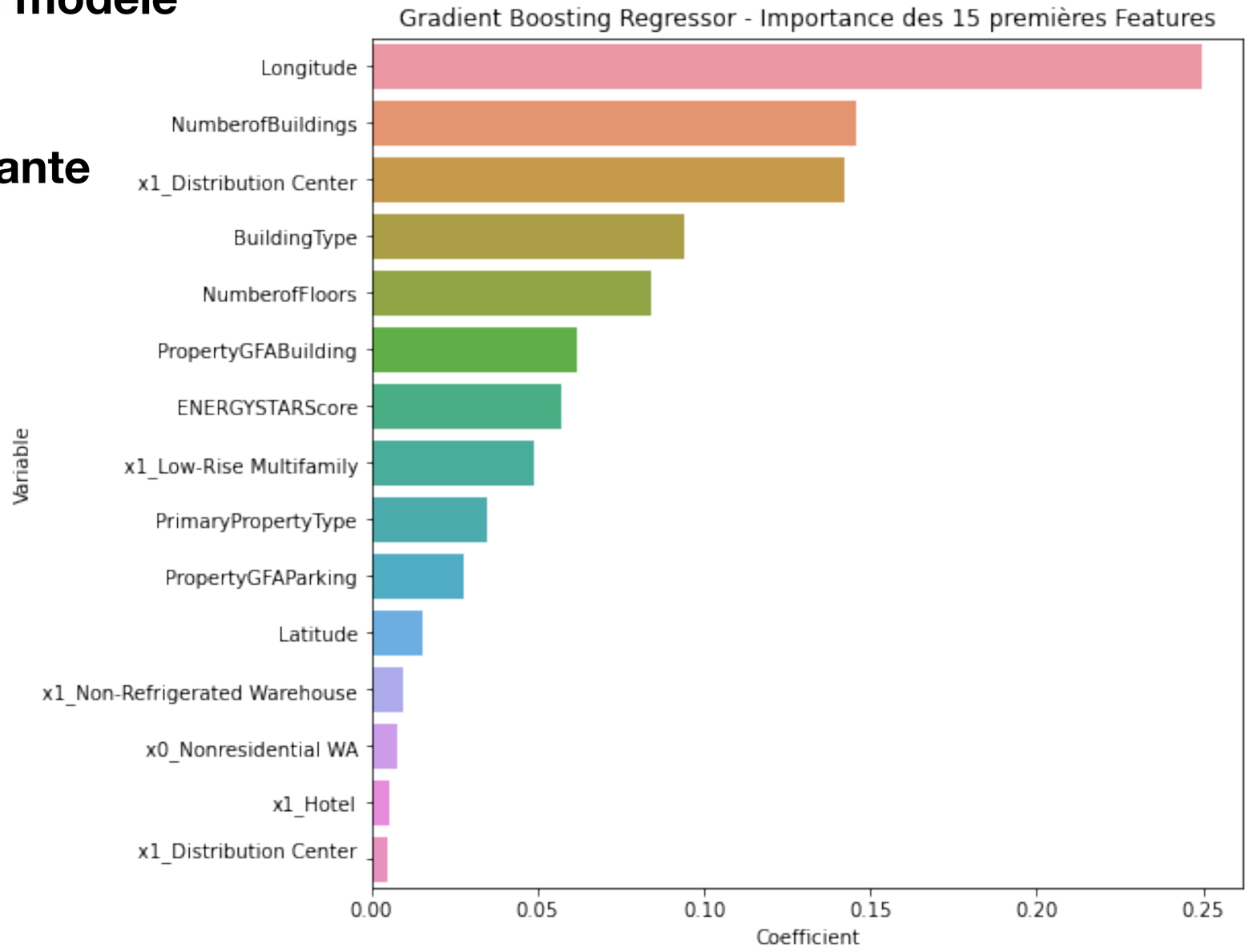




Modèles d'émission: Features Importance

- **Les variable Longitude et NumberofBuildings sont utiles pour la performance du modèle**

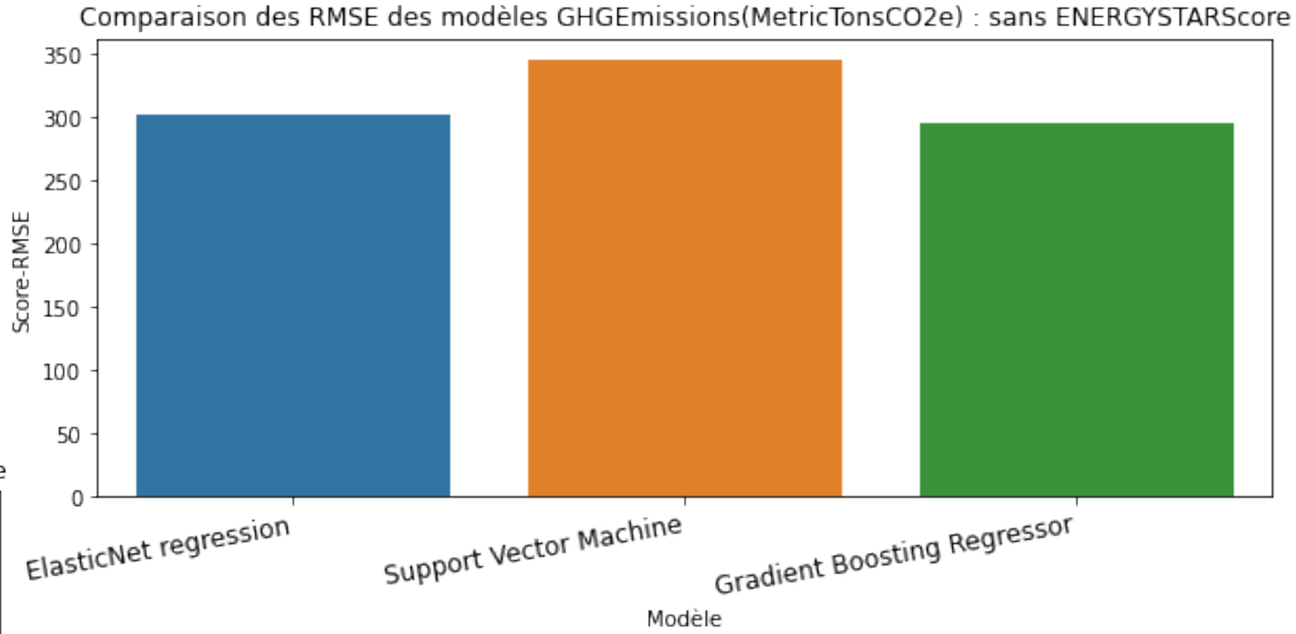
- **La variable ENERGYSTARScore est aussi utile mais moins importante**

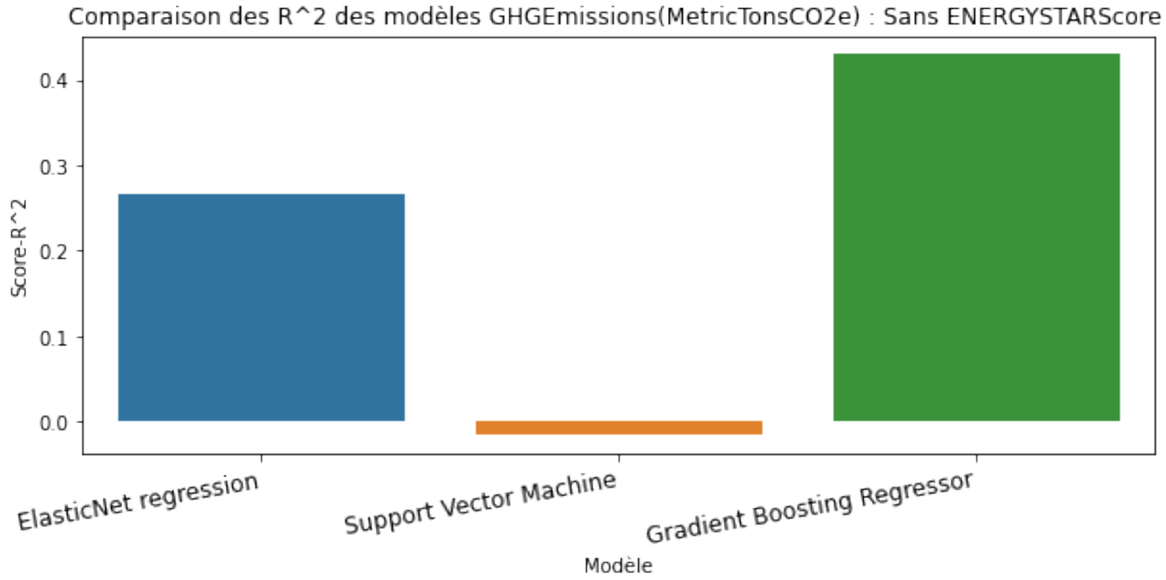


Modèles d'émission: sans ENERGYSTARScore

- **La variable target : GHGEmissions(MetricTonsCO2e)**
- **Le meilleur modèle selon la RMSE est Gradient Boot Regressor**

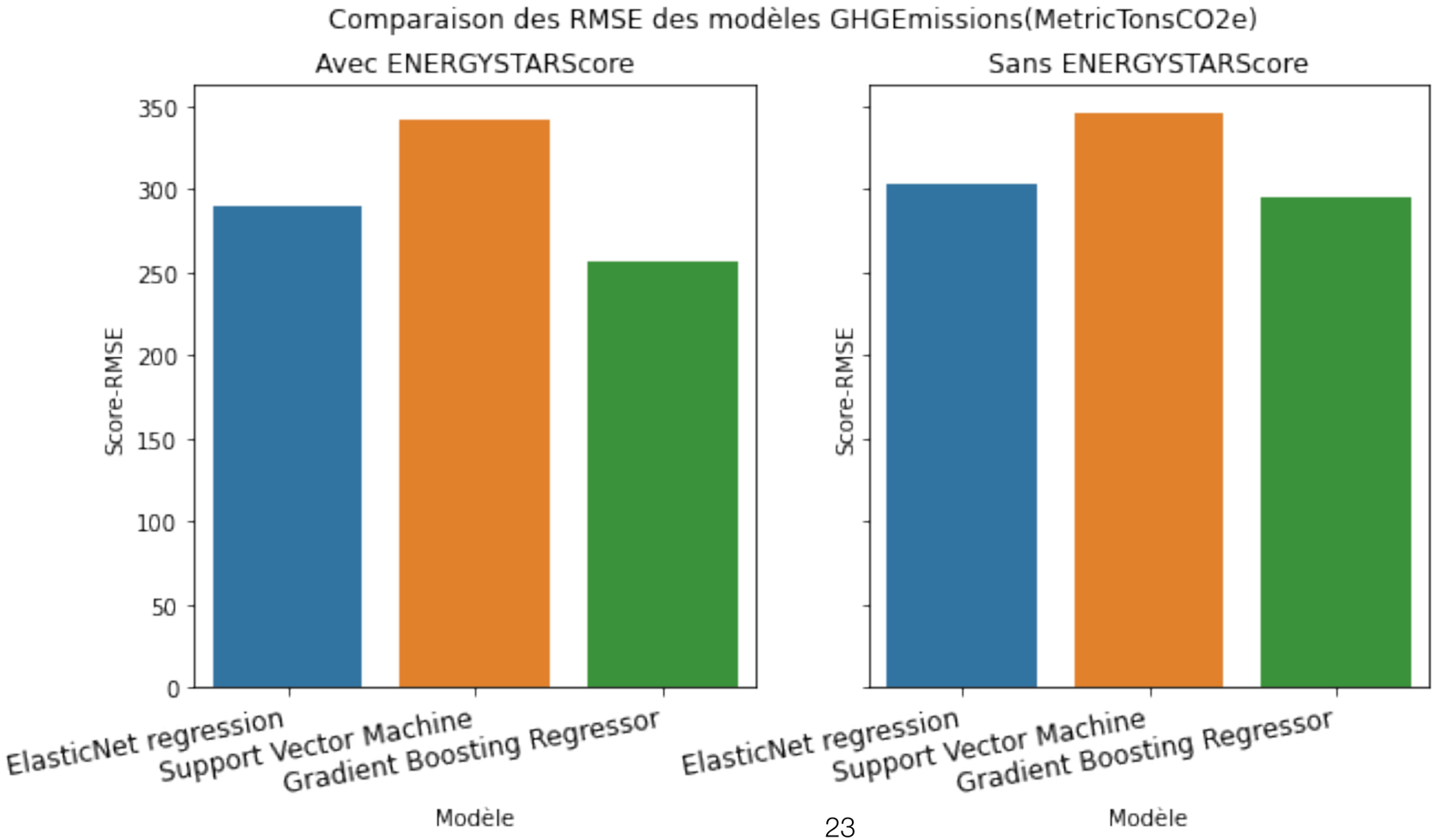
Modèle	Score-R^2	Score-RMSE
ElasticNet regression	0.201962	302.424373
Support Vector Machine	-0.041724	345.526505
Gradient Boosting Regressor	0.237418	295.629872





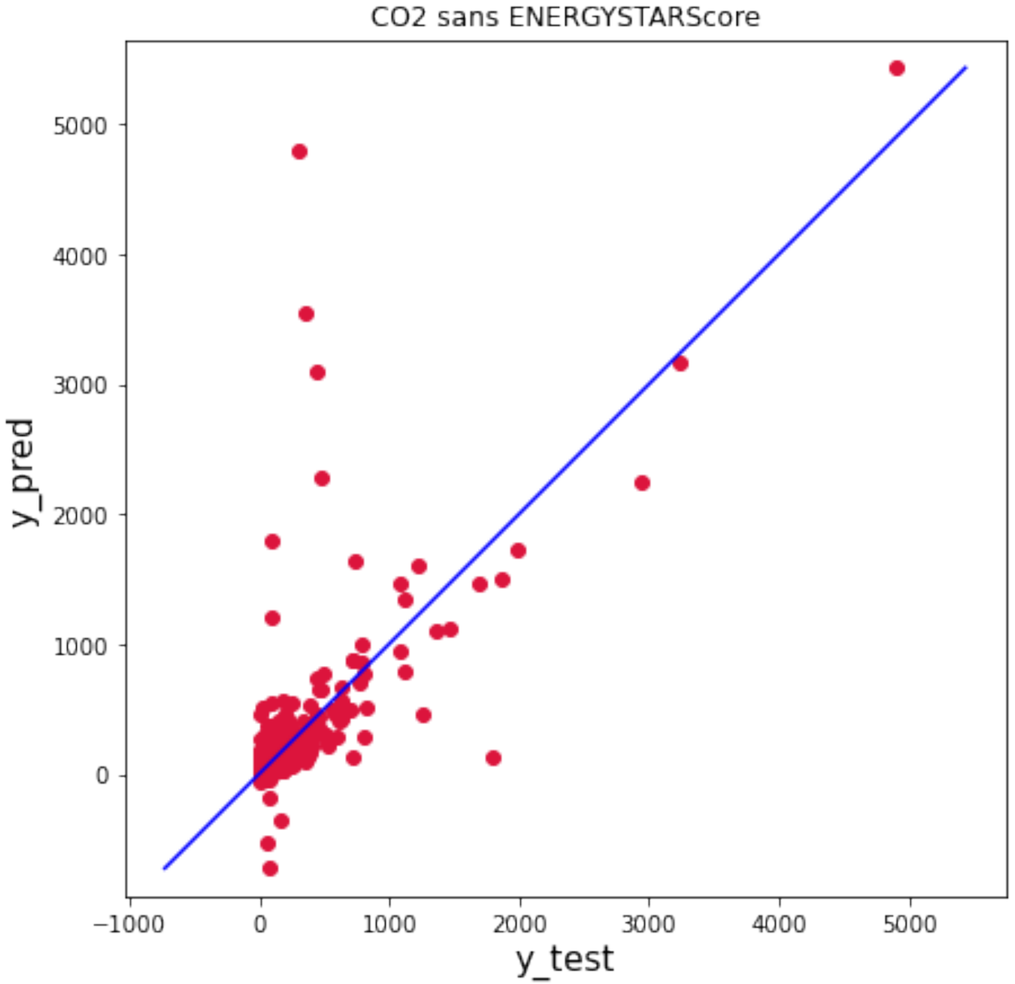
Comparaison des modèles d'émission: avec ENERGYSTARScore et sans ENERGYSTARScore

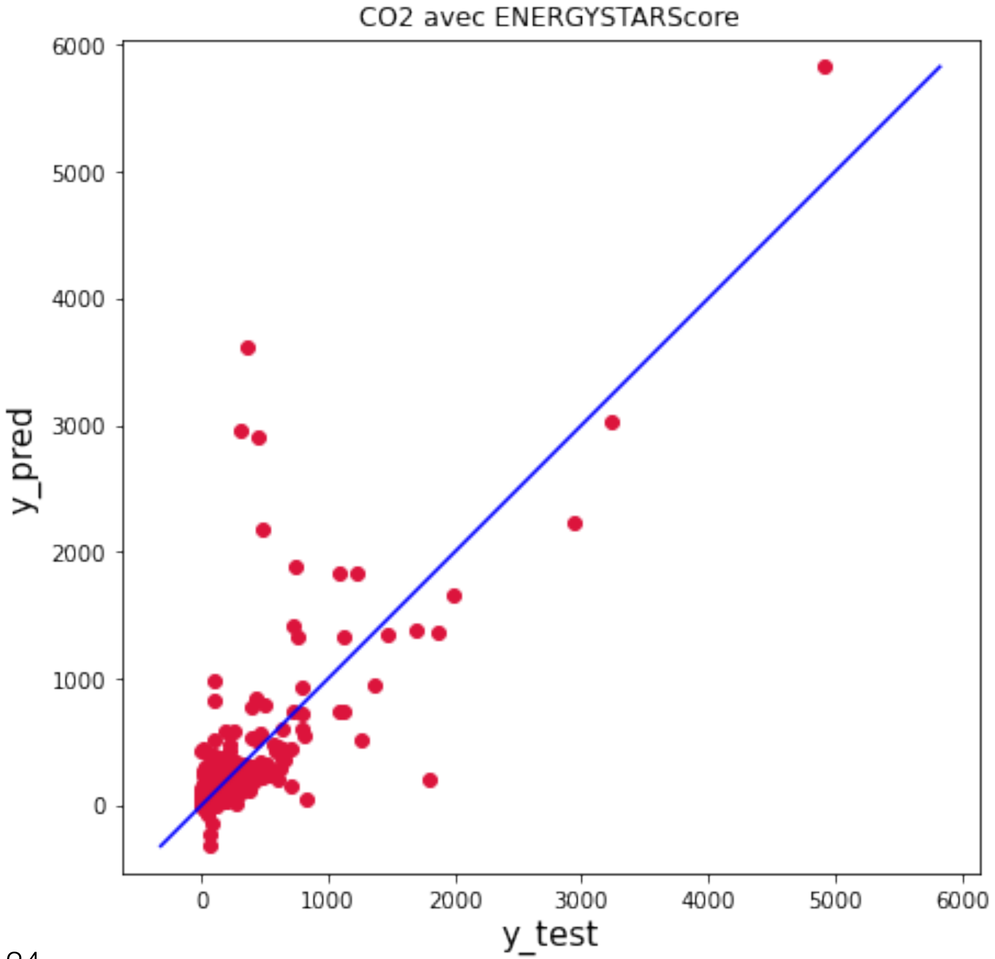
- **Le meilleur modèle dans les deux cas est Gradient Boot Regressor**
- **L'influence de la feature :ENERGYSTARScore sur la performance du modèle est faible**



Comparaison des modèles d'émission: avec ENERGYSTARScore et sans ENERGYSTARScore

- **Les deux modèles sont similaires**
- **L'intéret de la feature ENERGYSTARScore est limité.**





CONCLUSIONS

Modèle de Consommation et Modèle d'émissions

- **Le meilleur modèle pour la consommation est ElasticNet regression et pour l'émission est Gradient Boot Regressor**
- **En présence de l'indicateur ENERGYSTARScore la plus petite RMSE est celle de Gradient Boot Regressor**
- **La suppression de cette feature diminue significativement le nombre de valeurs manquantes.**
- **Le meilleur modèle reste Gradient Boot Regressor pour l'émission avec augmentation de la RMSE du modèle d'émission.**