PROJET 4
Anticipez les besoins en consommation électrique de bâtiments

Problématique et pistes recherches

L’objectif de la ville de Seattle est d’être neutre en émissions de carbone en 2050.

Pour cela des relevés par des agents ont été exécutés en 2015 et 2016. Ces relevés sont fastidieux et demandent beaucoup de travail, la ville voudrait tenter de prédire les données pour les bâtiments non destinés à l’habitation et non encore relevés.

Nous allons chercher à prédire la consommation d'énergie et la quantité de CO2 émis. Nous allons aussi évaluer l ’intérêt du calcul du score « ENERGY STAR » pour prédire les émissions de gaz à effet de serre.

Ces prédictions devront se baser sur les données déclaratives du permis d’exploitation commerciale (taille et usage des bâtiments, date de construction).

Le score « ENERGY STAR » étant calculé en fonction de la consommation d'énergie, de l’utilisation du bâtiment, on pourrait envisager d’utiliser la prédiction de consommation d'énergie pour prédire l’émission de CO2. Nous ferons une comparaison avec et sans cette donnée.

Analyse exploratoire des données

Analyse de la forme

Deux variables targets : consommation d'Energie, émission de CO2

Nous avons deux jeux de données 2015 et 2016 contenant un peu plus de 3000 lignes et 46 colonnes. Les bâtiments sont identifiés par un numéro unique ce qui va permettre de fusionner les deux fichiers. Je vais faire une moyenne entre les deux valeurs pour le jeu de données final.

il y a 20 colonnes de données quantitatives et 26 colonnes qualitatives.

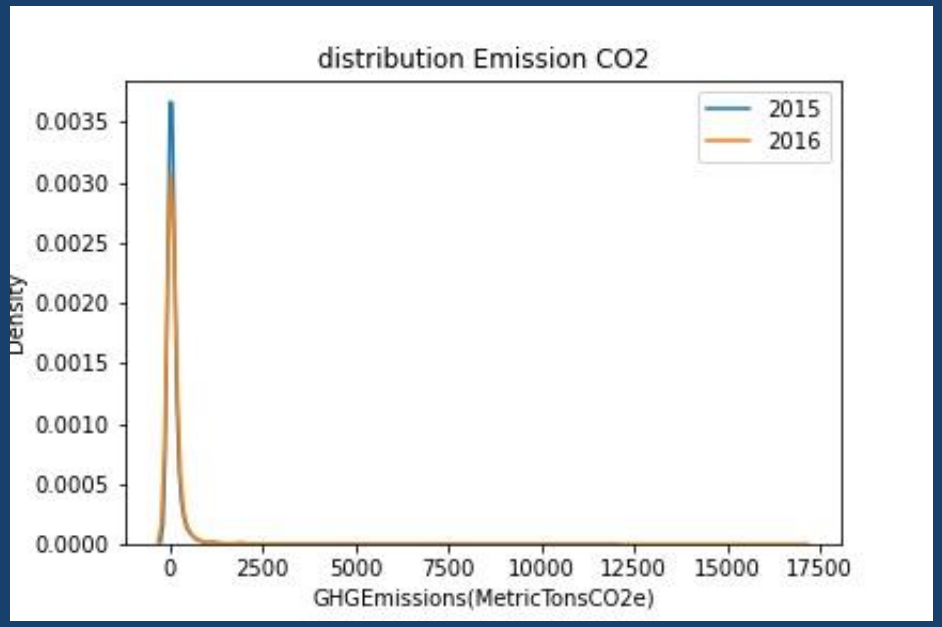
peu de valeur manquantes pour les features que nous allons utiliser sauf EnergyStarScore (remplie dans 75% des données)

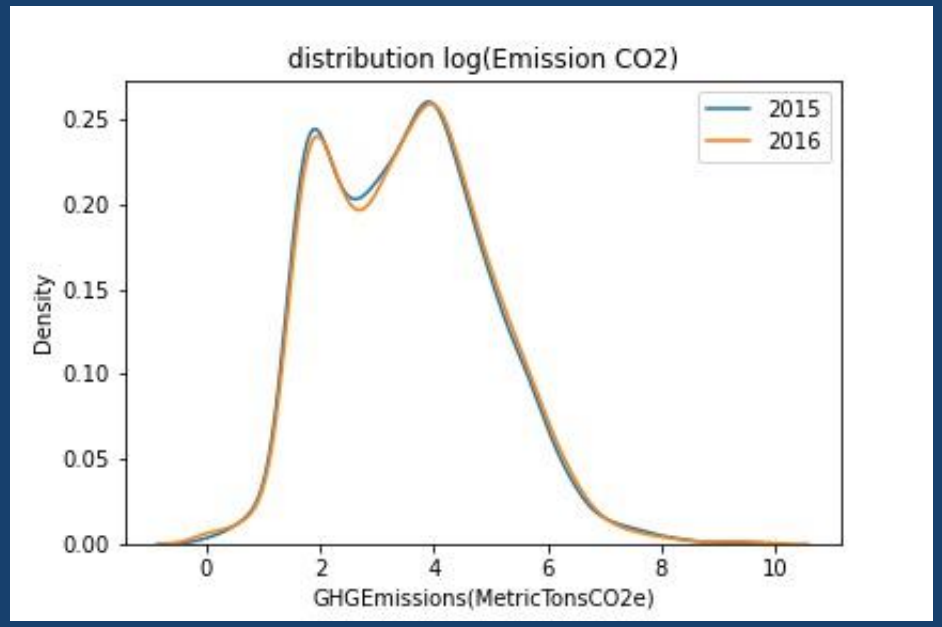
Analyse exploratoire des données

Analyse de Fond : target émission gaz effet de serre CO2

On supprime les quelques valeurs nulles ou négatives.

Moyenne 110 kBtu avec ecart type 410 en passant au log moyenne 3,5 avec ecart type 1,4



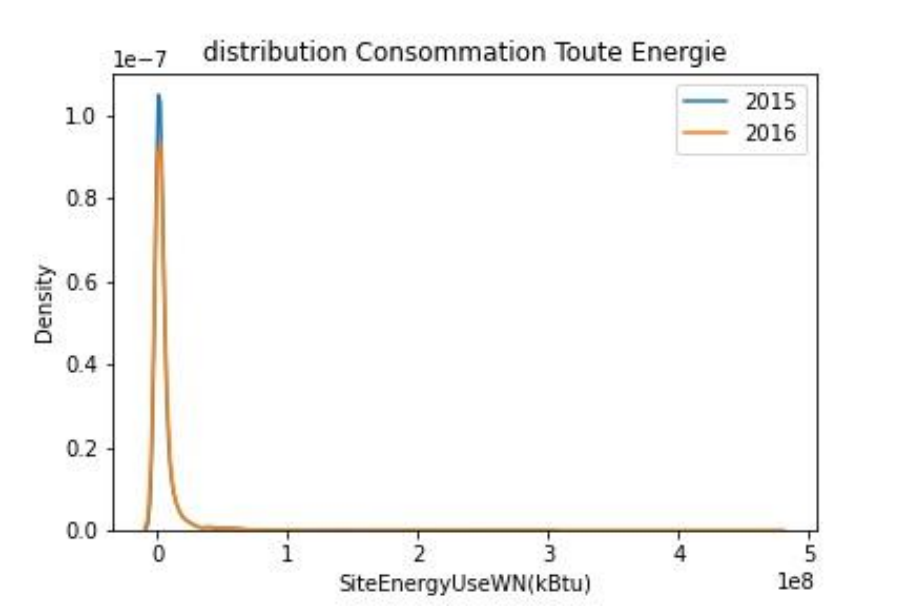


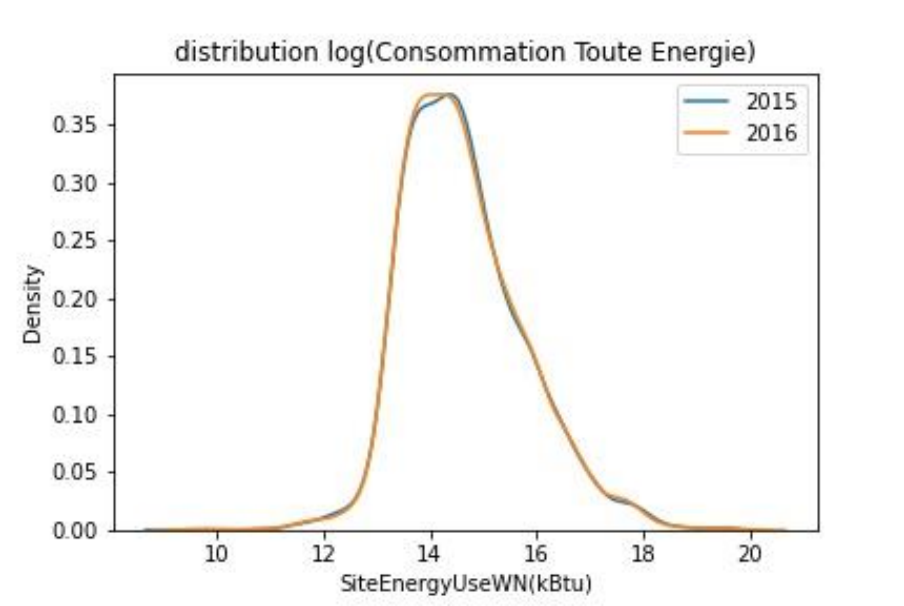
Analyse exploratoire des données

Analyse de Fond : target Consommation Energie

On supprime les quelques valeurs nulles ou négatives

Moyenne $5*10^6$ kBtu avec ecart type $14*10^6$ en passant au log moyenne 14,6 avec ecart type 1,2





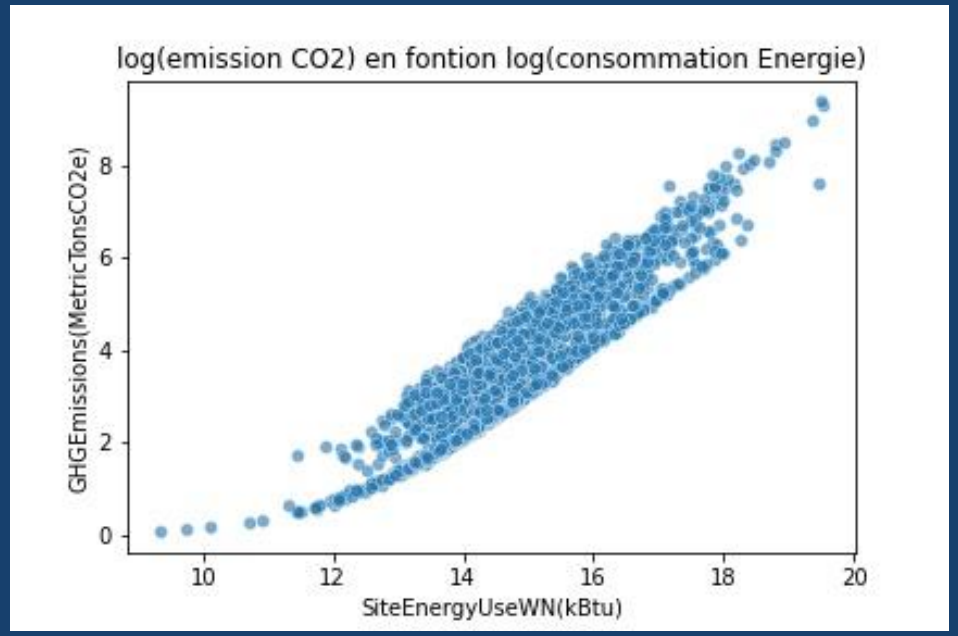
Analyse exploratoire des données

Analyse de Fond : corrélation entre target

Coefficient de corrélation entre la consommation énergétique et émission gaz effet de serre

Pour 2015 : 0,897

Pour 2016 : 0,921



Analyse exploratoire des données

Analyse de Fond : features quantitatives conservés pour la modélisation

'NumberofFloors’ : nombres d’étages

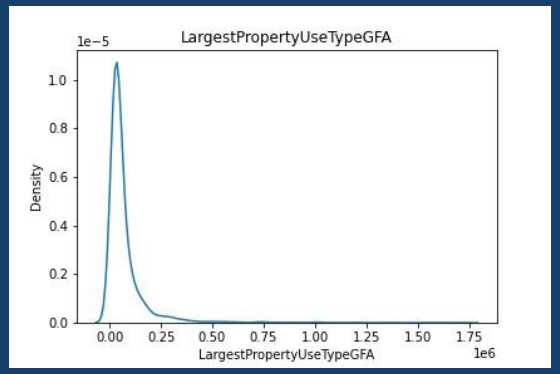
'PropertyGFAParking’ : surface totale de parking

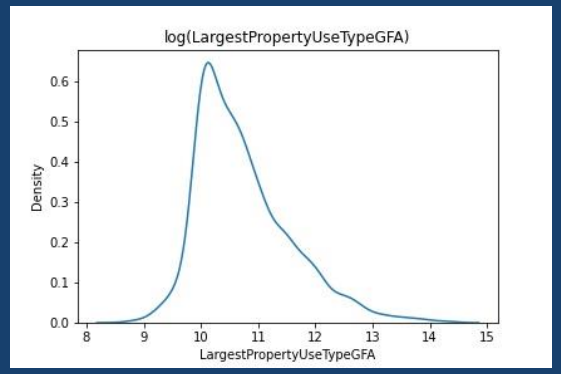
'LargestPropertyUseTypeGFA’ : surface utilisé pour l’utilisation principale du bâtiment

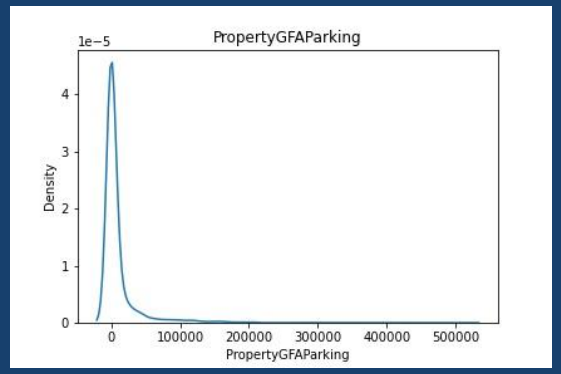
'ENERGYSTARScore’ : score EnergySTAR

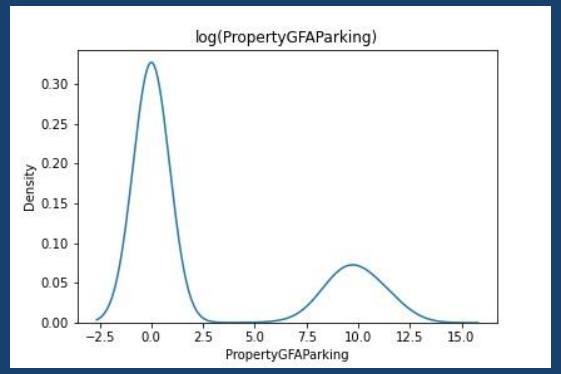
Analyse exploratoire des données

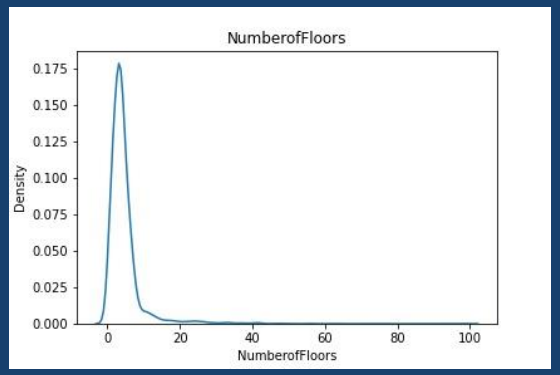
Analyse de Fond : features quantitatives conservés pour la modélisation

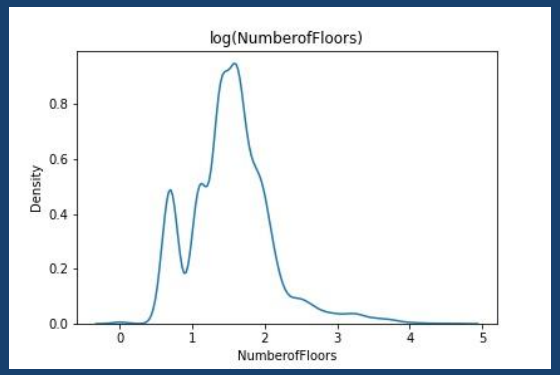


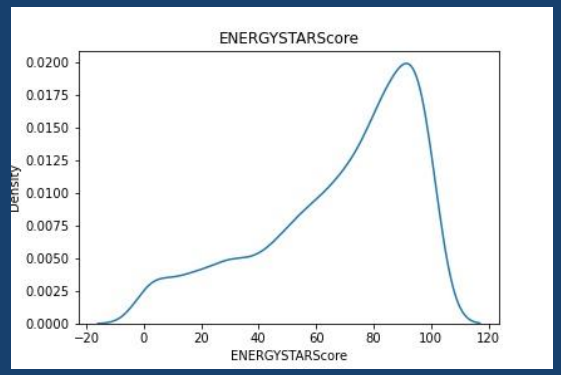


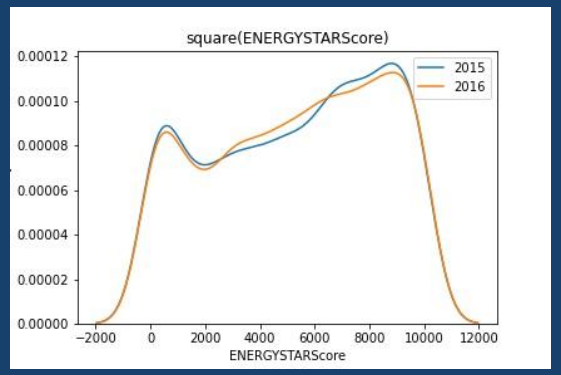












Analyse exploratoire des données

Analyse de Fond : features qualitatives conservés pour la modélisation

‘YearBuilt’ : année de construction qui va nous permettre de calculer l’âge du bâtiment

‘LargestPropertyUseType’ : utilisation principale du bâtiment (on a remarqué que le second type d’utilisation est principalement parking qui influence peu la consommation Energie et émission CO2)

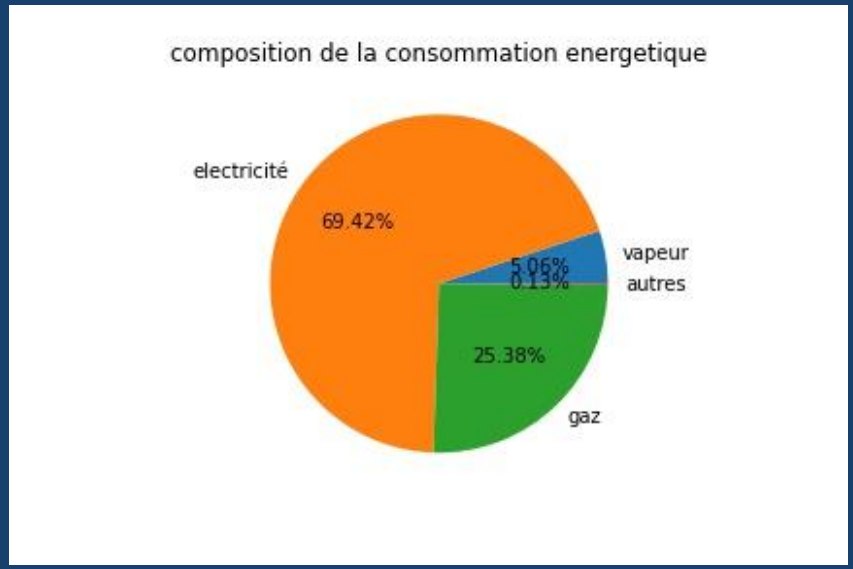
‘Neighborhood’ : quartier dans Seattle du bâtiment

Analyse exploratoire des données

Analyse de Fond : features calculés

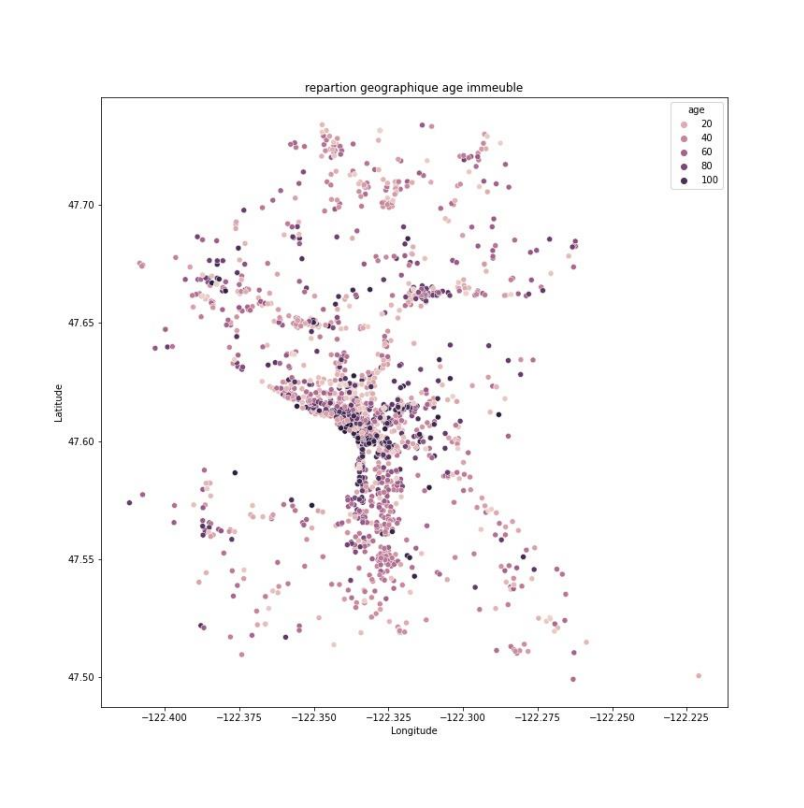
- âge : différence entre 2015 et année construction

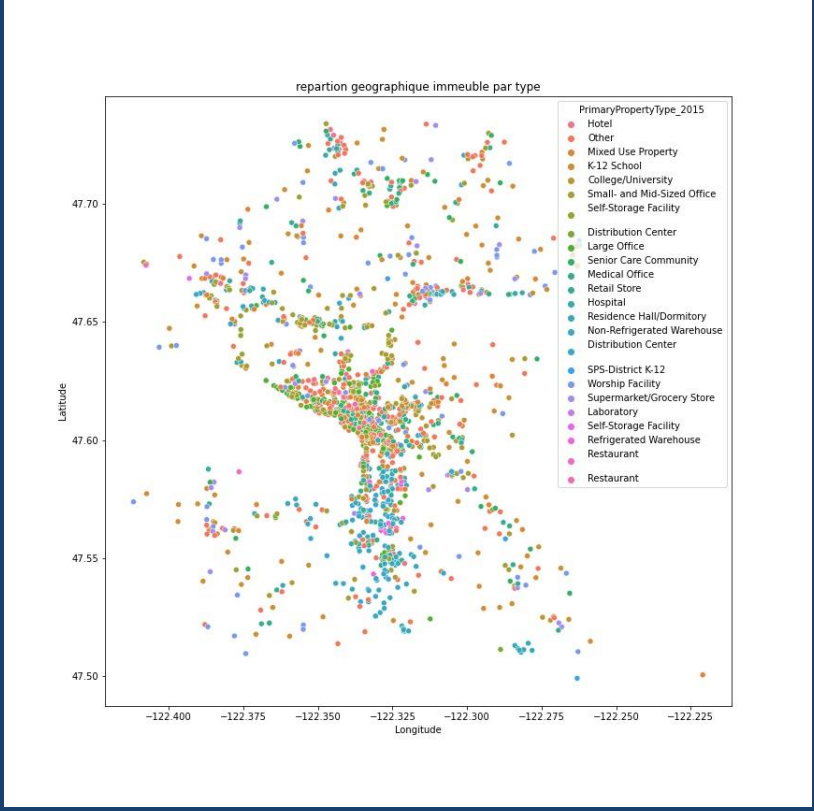
- répartition type énergie par type d'utilisation principale



Analyse exploratoire des données

Analyse de Fond : feature geographique ?





PREPROCESSING

- passage au log pour les features :
  - NumberofFloors
  - LargestPropertyUseTypeGFA
  - PropertyGFAParking
- OneHotEncoding (passage de variable qualitative en quantitative) pour les features :
  - Neighborhood
  - PrimaryPropertyType

- Création d’un Train set et Test set pour comparer les modèles

MODELISATION

- Régression linéaire, Régression Ridge et Lasso
- DecisionTree
- KNeighbors
- RandomForest
- ADABoost
- GradientBoost
- XGBOOST

MODELISATION

Recherche des valeurs pour les hyperparamètres :

1. d’abord un RandomizedSearchCV
2. À partir de ces paramètres on fait un GridSearchCV pour affiner. On mesure le temps de calcul pour trouver un compromis temps / optimisation

MODELISATION

Quelles sont les hyperparamètres :

- Régression linéaire : aucun
- Régression Ridge : alpha, force de la pénalisation (carré de la pente)
- Régression Lasso : alpha, force de la pénalisation (valeur absolue la pente)

MODELISATION

Quelles sont les hyperparamètres :

- DecisionTree :
  1. Criterion : fonction pour mesurer la qualité du split
  2. Max_depth : profondeur maximale de l'arbre

- Kneighbors
  1. n_neighbors : nombre de voisins recherchés

MODELISATION

Quelles sont les hyperparamètres :

- RandomForest
  1. n_estimators : nombre d'arbres dans la forêt
  2. max_features : nombre de features conservés pour chaque arbre
  3. max_depth : profondeur maxi des arbres
  4. min_samples_split : minimum d'échantillon dans une feuille pour éclater un nœud interne
  5. min_samples_leaf : minimum d'echantillon pour devenir une feuille

MODELISATION

Quelles sont les hyperparamètres :

- ADABoost
  1. n_estimators : nombre d'arbres dans la forêt
  2. learning_rate : taux d'apprentissage (réduction de la contribution de chaque arbre)
  3. loss : fonction utilisé pour calculer le nouveau poids de chaque échantillon

MODELISATION

Quelles sont les hyperparamètres :

- GradientBoost
  1. n_estimators : nombre d'arbres dans la forêt
  2. learning_rate : taux d'apprentissage (réduction de la contribution de chaque arbre)
  3. Criterion : fonction pour mesurer la qualité du split
  4. max_depth : profondeur maxi des arbres
  5. min_samples_leaf : minimum d'échantillon pour devenir une feuille
  6. min_samples_split : minimum d'échantillon dans une feuille pour éclater un nœud interne
  7. max_features : nombre de features conservés pour chaque arbre
  8. Loss : fonction utilisé pour calculer le nouveau poids de chaque échantillon

MODELISATION

Quelles sont les hyperparamètres :

- XGBOOST
  1. n_estimators : nombre d'arbres dans la forêt
  2. learning_rate : taux d'apprentissage (réduction de la contribution de chaque arbre)
  3. Gamma : paramètre de régularisation pour faire un nouveau split
  4. min_child_weight : nombre minimum d'échantillon pour former une feuille
  5. max_depth : profondeur maxi des arbres
  6. colsample_bytree : nombre de features utilisés pour nouvel arbre
  7. Subsample : sous échantillonnage pour chaque nouveau tree
  8. reg_lambda : paramètre de régularisation de type L2 (type Ridge)

RESULTAT Modélisation Consommation

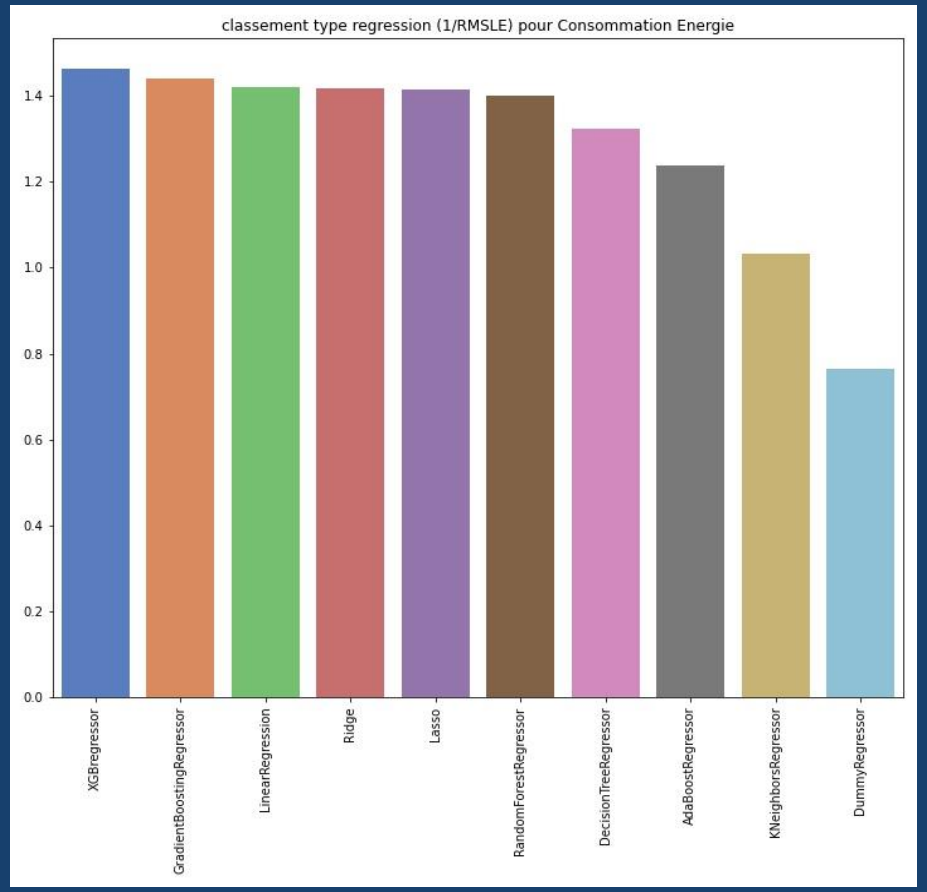
Modele retenu XGBRegressor avec les hyperparametres

subsample = 0.7, reg_lambda = 1.1, gamma = 0.5,

n_estimators = 150, max_depth = 4, min_child_weight = 3,

learning_rate = 0.1, colsample_bytree = 0.8

modele	score R2	RMSLE	temps (sec)
XGBregressor	0,7260	0,6843	0,13
GradientBoostingRegressor	0,7178	0,6944	161,8
LinearRegression	0,7101	0,7039	0,02
Ridge	0,7088	0,7055	0,01
Lasso	0,7070	0,7077	0,01
RandomForestRegressor	0,7018	0,7139	0,61
DecisionTreeRegressor	0,6658	0,7557	0,1
AdaBoostRegressor	0,6178	0,8082	5,71
KNeighborsRegressor	0,4523	0,9675	0,02
DummyRegressor	-0,0029	1,3092	0

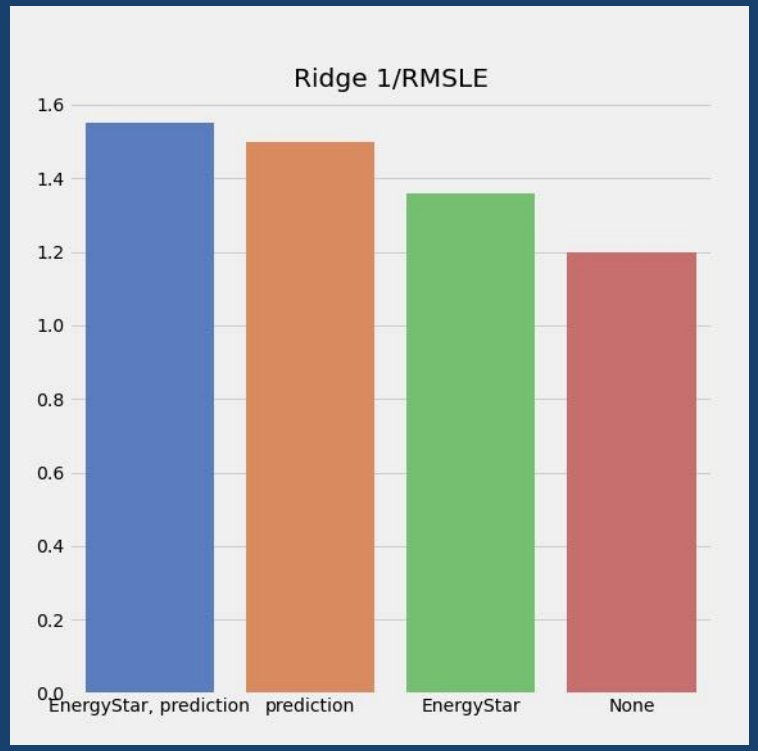


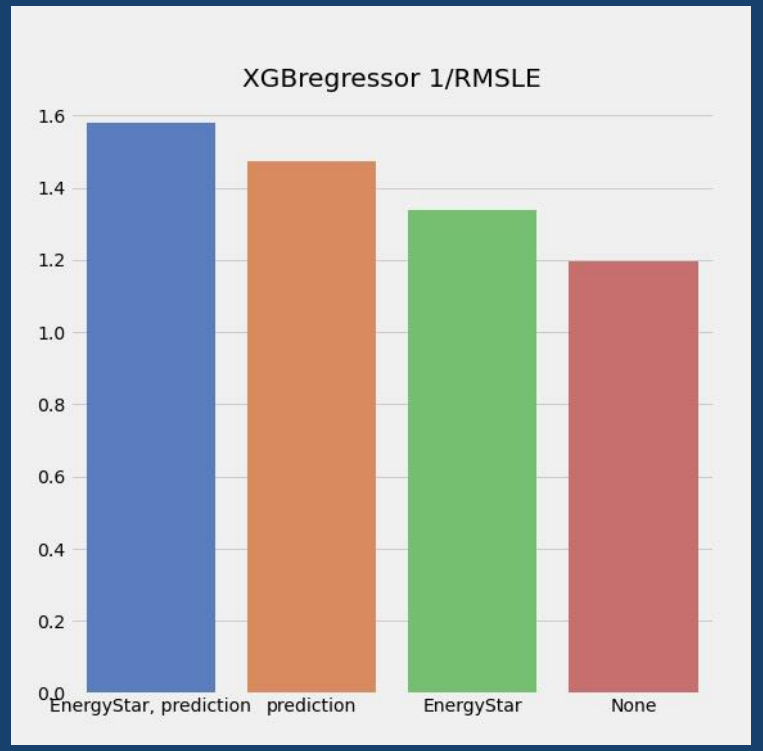
MODELISATION Emission CO2

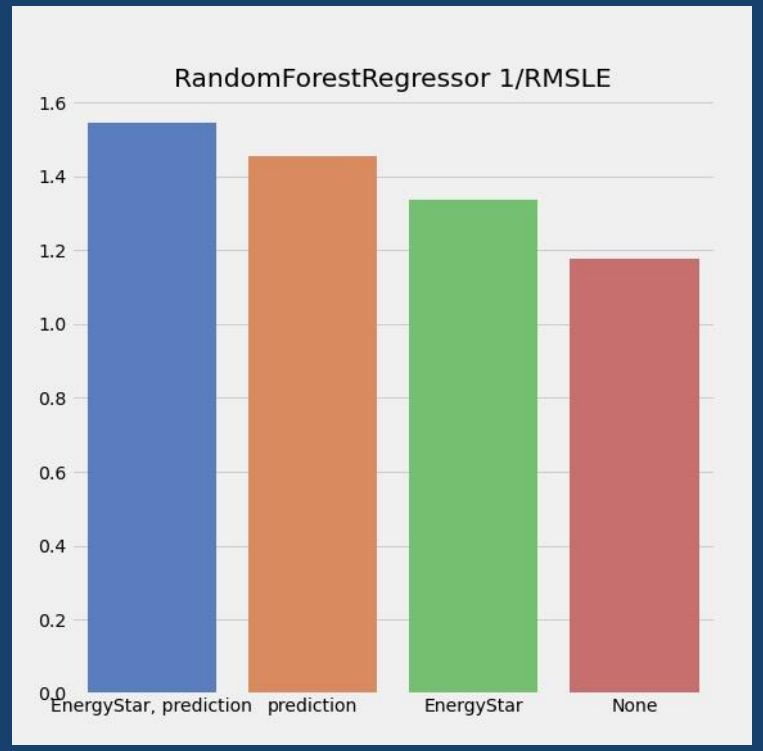
Essai avec les modèles ci-dessous avec comparaison de l'utilisation de 'ENERGYSTARscore' et prédiction consommation Energie à partir du modèle XGBOOST précèdent

- Régression linéaire, Régression Ridge et Lasso
- DecisionTree
- KNeighbors
- RandomForest
- ADABoost
- GradientBoost
- XGBOOST

RESULTAT Modélisation Emission CO2







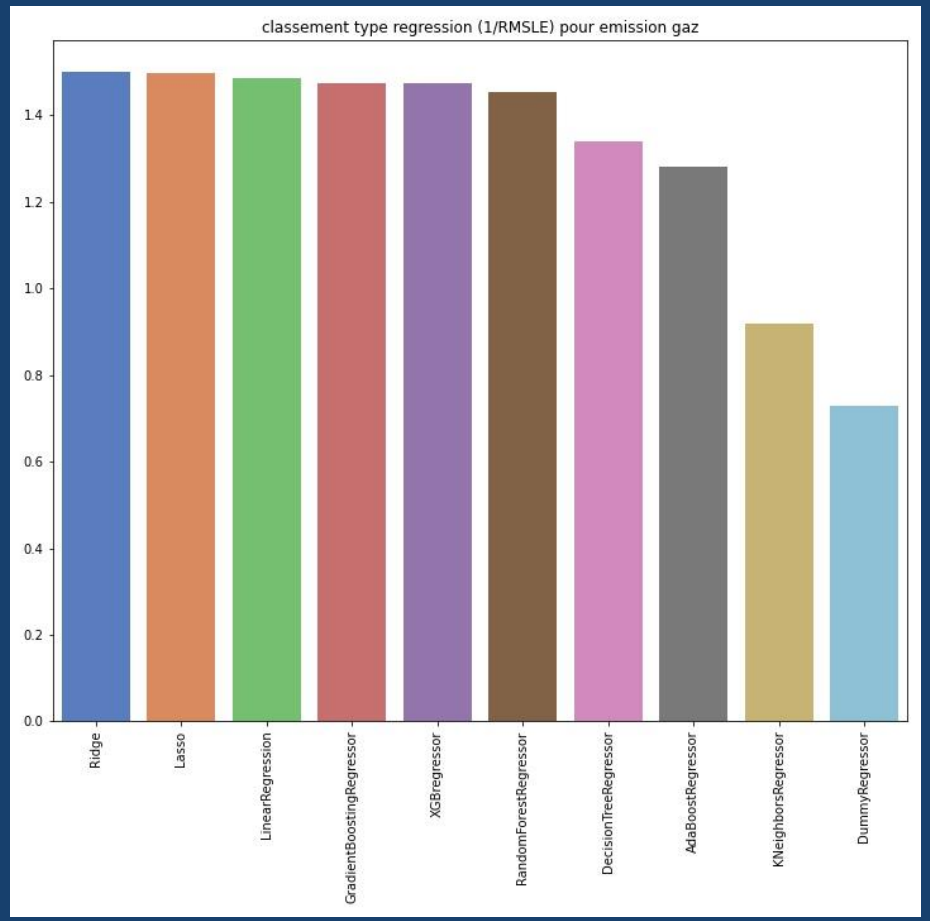
RESULTAT Modélisation Emission CO2

Modele retenu Regression Ridge

avec alpha = 1,45 et prédiction

de la consommation par XGBOOST

modele	score R2	RMSLE	temps (sec)
Ridge	0,7631	0,6670	0,01
Lasso	0,7629	0,6672	0,01
LinearRegression	0,7587	0,6731	0,01
GradientBoostingRegressor	0,7550	0,6783	2,46
XGBregressor	0,7547	0,6787	0,17
RandomForestRegressor	0,7477	0,6884	1,37
DecisionTreeRegressor	0,7029	0,7469	0,01
AdaBoostRegressor	0,6759	0,7801	0,09
KNeighborsRegressor	0,3679	1,0894	0,01
DummyRegressor	-0,0010	1,3711	0



MERCI

Questions et Réponses