Soutenance
Projet 4

Anticipez les besoins en consommation électrique de bâtiments

Seattle

OPENCLASSROOMS - Formation Data Scientist

Michel Blazevic

Ordre du jour

1. Introduction
2. Analyse et exploration
3. Modélisation pour la prédiction
4. Analyse du modèle choisi
5. Conclusion

1 Introduction

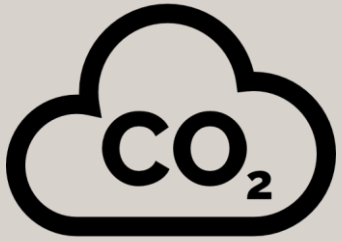
1. Problématique
2. Base de données
3. Les pistes envisagées

1 Introduction

1. Problématique

Seattle

Objectif Seattle: ville neutre en 2050



Volonté de connaître:

⇒ Emission CO2
⇒ Consommation total d'énergie

Problème: relevés couteux

Solution:

Pour connaître les informations CO2 et Energie sur des bâtiments non résidentiels grâce à la prédiction avec les données déclaratives du permis d'exploitation

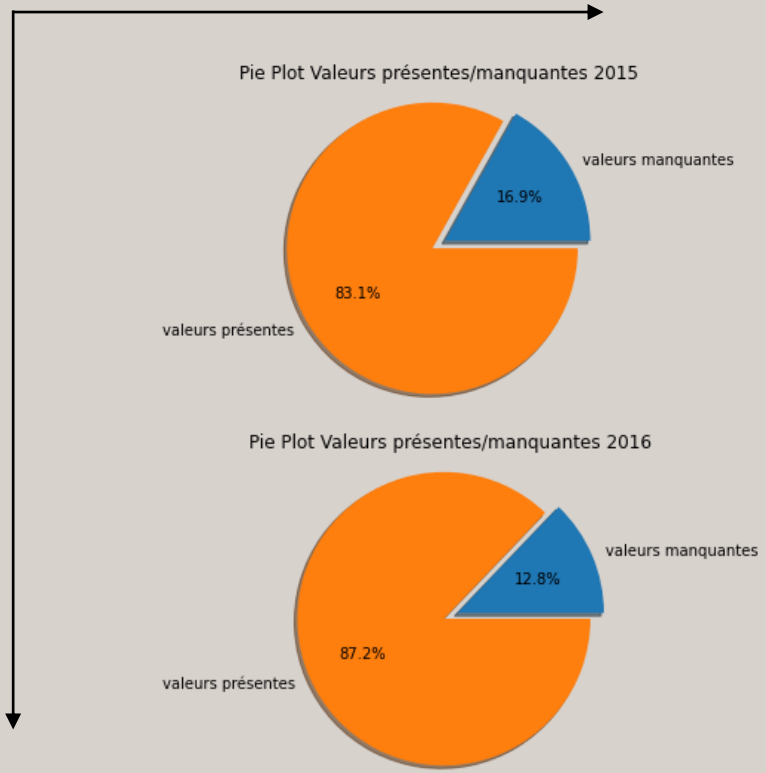
1 Introduction

2.Base de données

→ 2015 et 2016

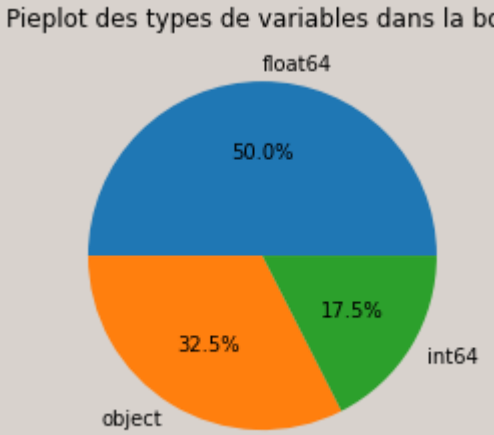
47 (2015)
46 (2016)
} Variables (type, utilisation, surface, consommation etc...)

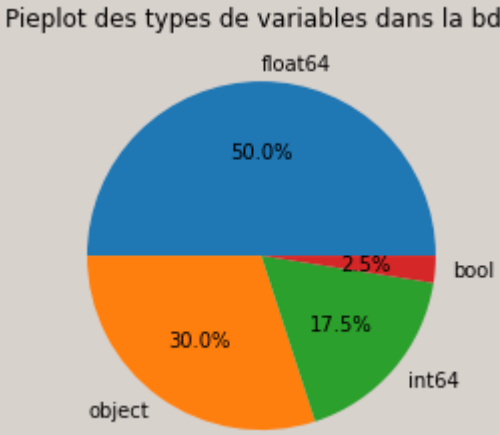
3340 (2015),
3376 (2016)
Bâtiments



Choix d'une première modification:

- On garde les individus communs entre 2015 et 2016
- On garde les variables communes et utilisables entres les deux années
- On change les types des données pour qu'elles soient communes entre les deux bdd





1 Introduction

3.Les pistes envisagées

- Observer les données
- Faire un choix sur les données qu'on veut utiliser pour la problématique
- Arranger les données pour les envoyer dans un modèle
- (feature engineering & pipeline)
- Entrainement de plusieurs modèles et évaluer sur plusieurs métrics
- Analyse du meilleur modèle, comparaison du même travail avec EnergyStarScore

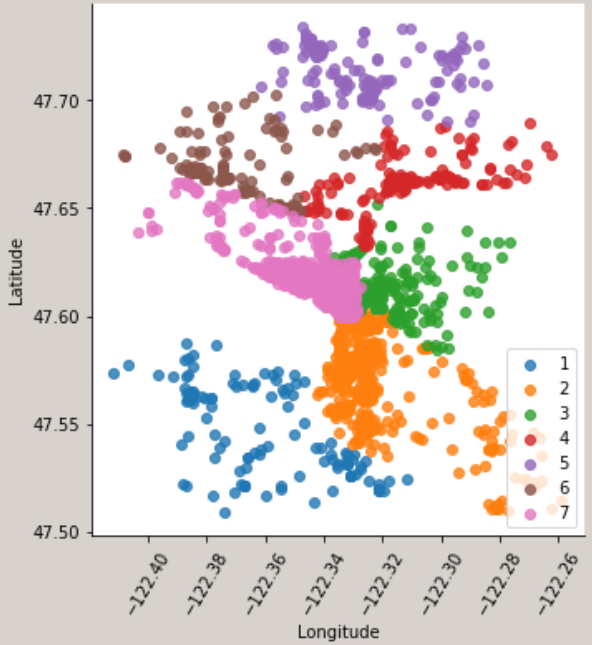
2 Analyse et exploration

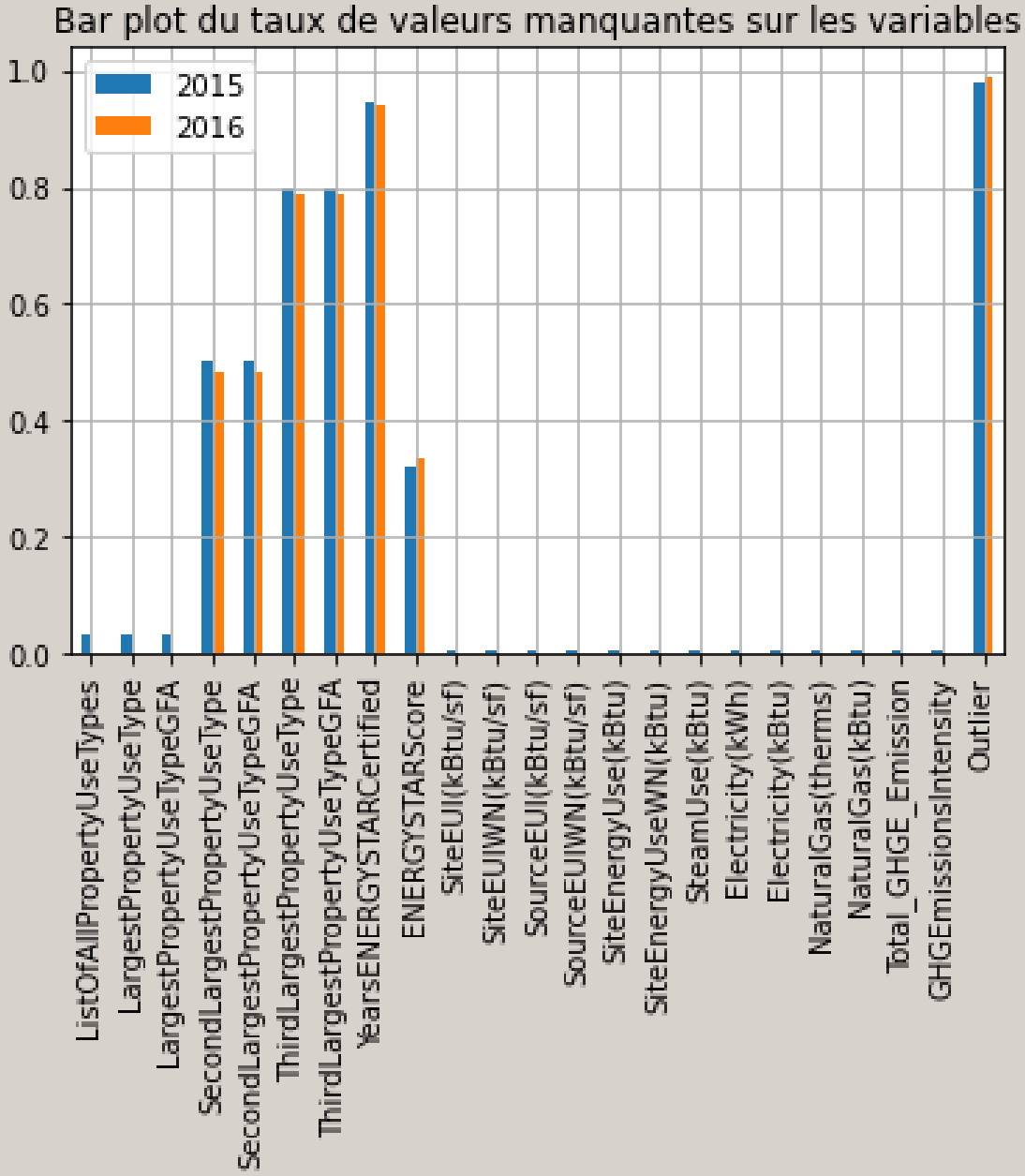
1. Nettoyage
2. Feature engineering
3. Exploration

2 Analyse et exploration

1.Nettoyage

- Suppression des variables où il y a beaucoup de valeurs manquantes
- Réarrangement des variables d'utilisation du bâtiment en 7 catégories (Autre, Commerce, service public, Entertainment, Enseignement, bureau, logement-hôtel)
- Choix de garder CouncilDistrictCode et non Neighborhood

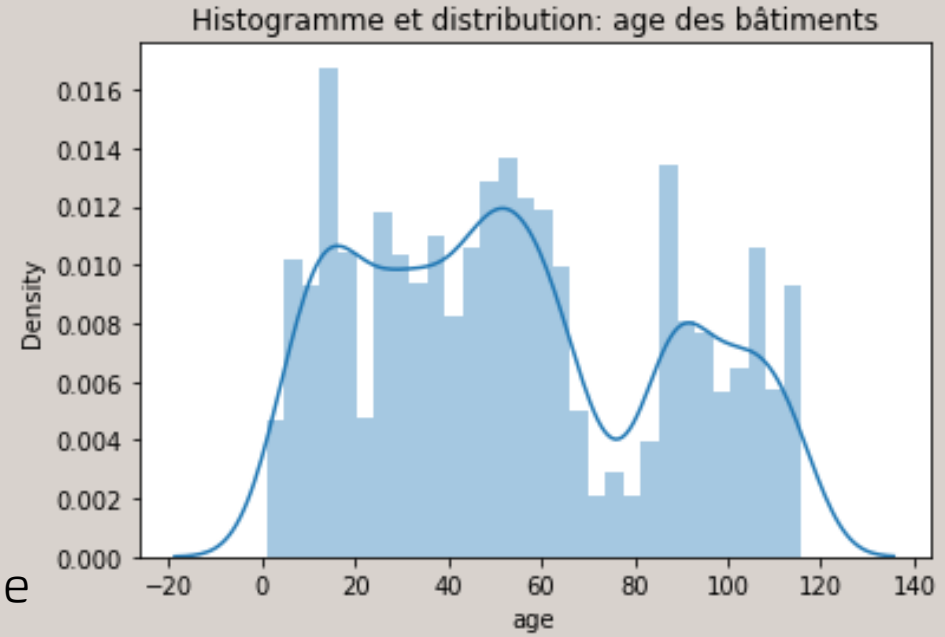


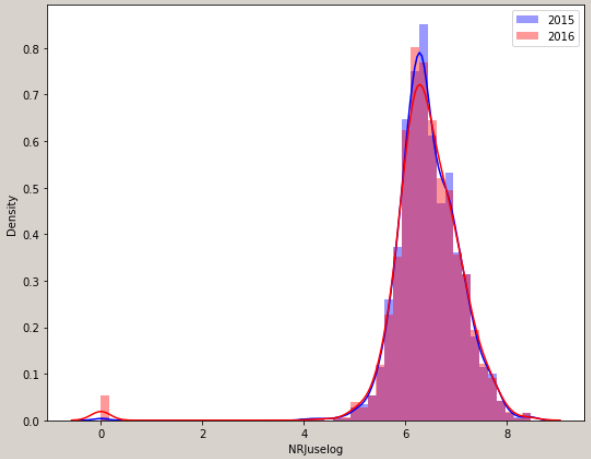


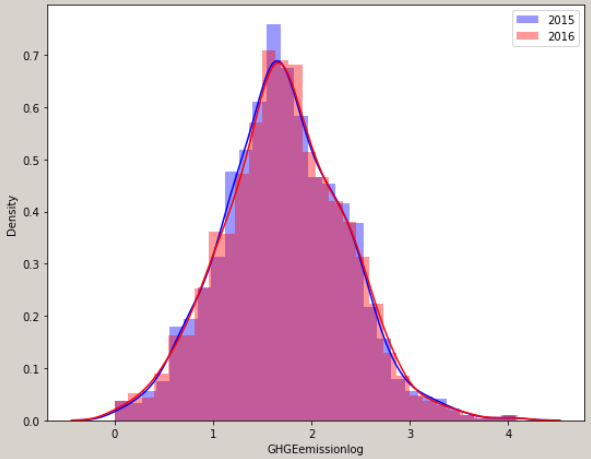
2 Analyse et exploration

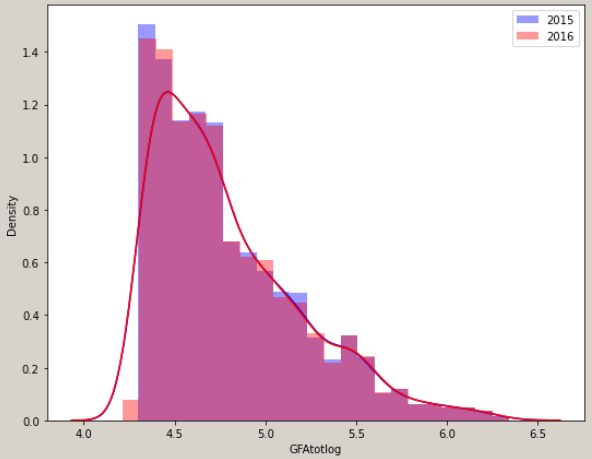
2.Feature engineering

- Création de d'une transformation logarithmique (Bijection) pour surface total, consommation d'Energie totale et émission de CO2
- Calcul du pourcentage des surfaces Parking, buildings etc
- Calcul de l'âge du bâtiment en fonction de la date de construction et l'année de collecte d'information
- Pourcentage de consommation d'énergie par rapport au type de source









2 Analyse et exploration

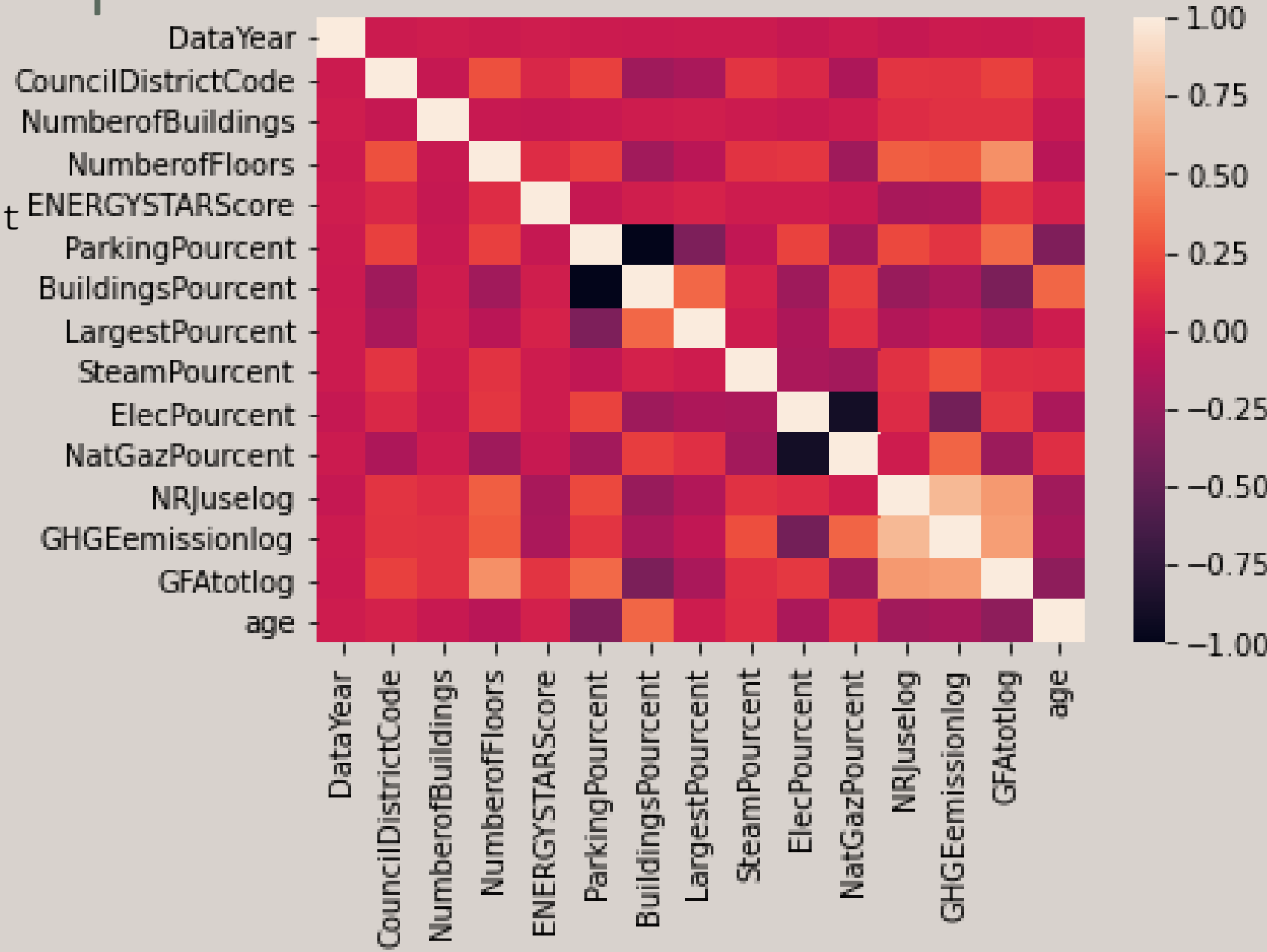
3.Exploration Corrélation

- ParkingPourcent/BuildingPourcent
  &
  ElecPourcent/GazNatPourcent

  ⇒ Fortement inv. corrélées

- Il existe une corrélation intéressante entre:

  ⇒ Age
  ⇒ GFAtotlog
  ⇒ GHGEemissionlog
  ⇒ Nrjuselog



3 Modélisation

1. Rappel de la problématique et du Dataset
2. Séparation Données entraînement/Validation
3. Pipeline modification données pour entrainement
4. .Différents modèles envisagés et leurs hyperparamètres testés
5. Choix du modèle le plus performant selon plusieurs critères

3 Modélisation

1.Rappel de la problématique et du Dataset

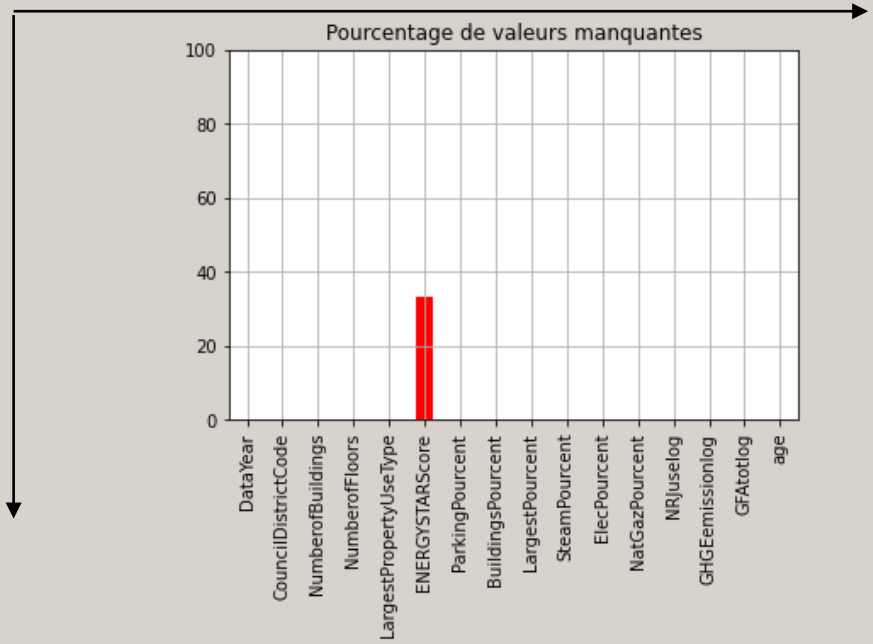
Rappel problématique

- Prédiction CO2 & Energie consommée
- Choix du modèle le plus performant
  - Comparaison par métriques
  - Évaluation des modèles et choix des hyperparamètres
- Interprétation du modèle sur le jeu de test

Rappel Dataset

16 variables

3082 individus



Choix pour les données:

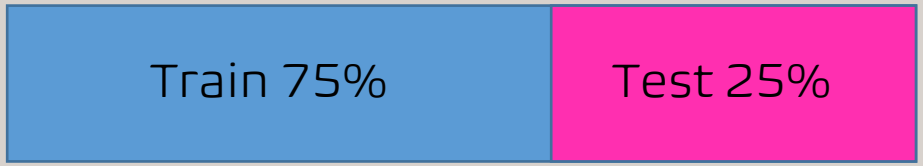
- On ne prend pas les différents types d'énergie %
- Nos Targets sont GHGEemissionlog et NRJuselog
- Avec et sans ENERGYSTARSCORE

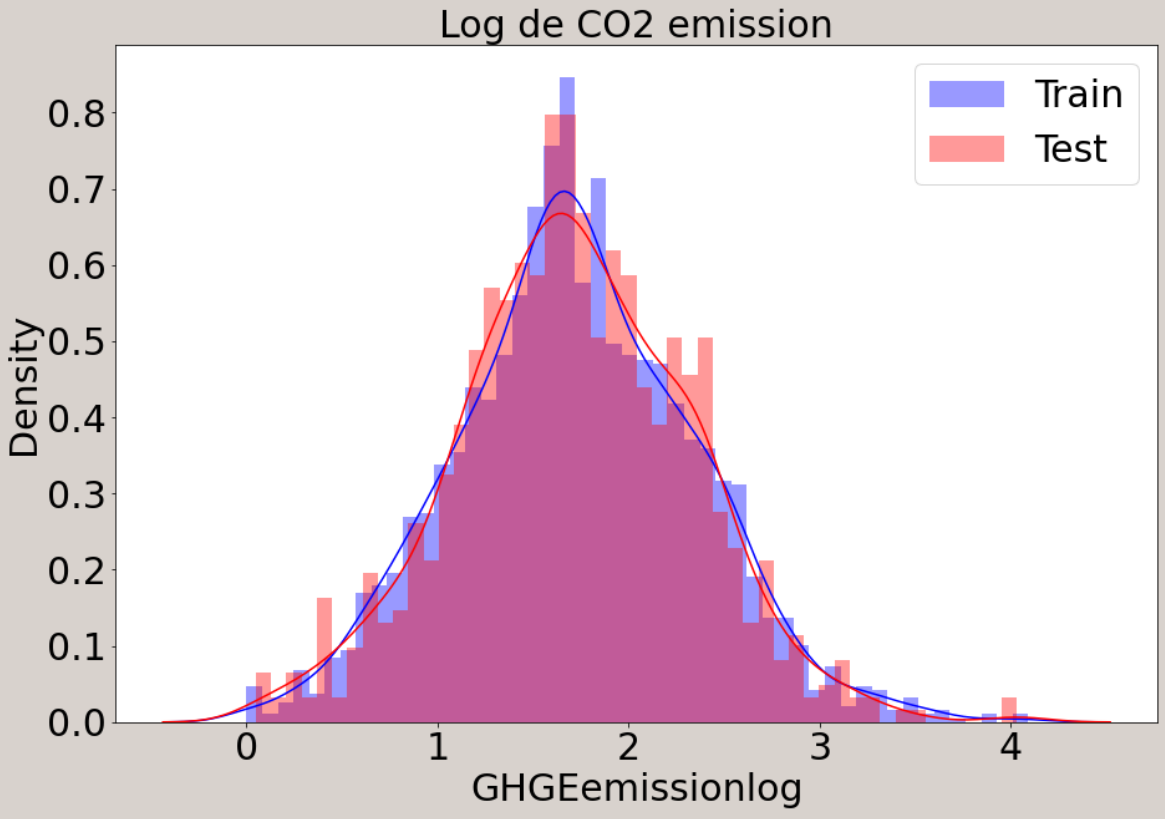
3 Modélisation

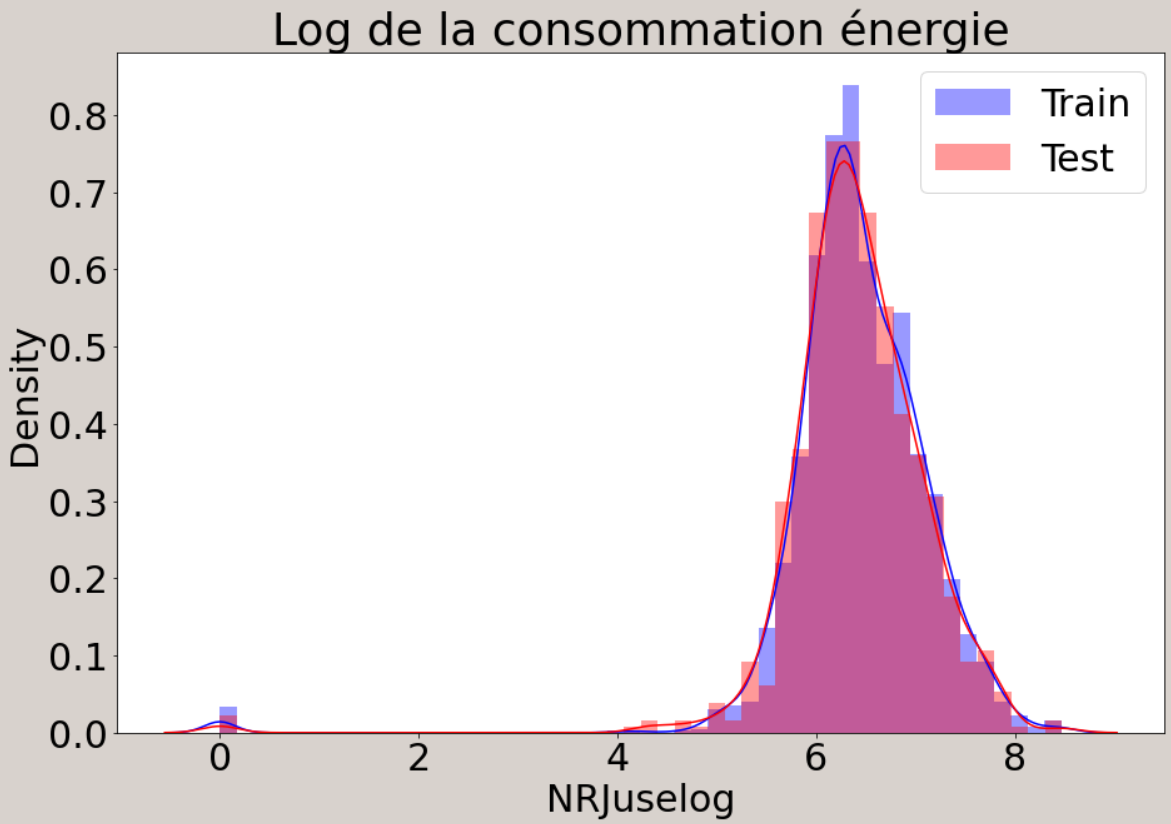
2. Séparation Données entraînement/Validation:

train_test_split de Scikit-Learn

Stratify: DataYear, CouncilDistrictCode, LargestPropertyUseType

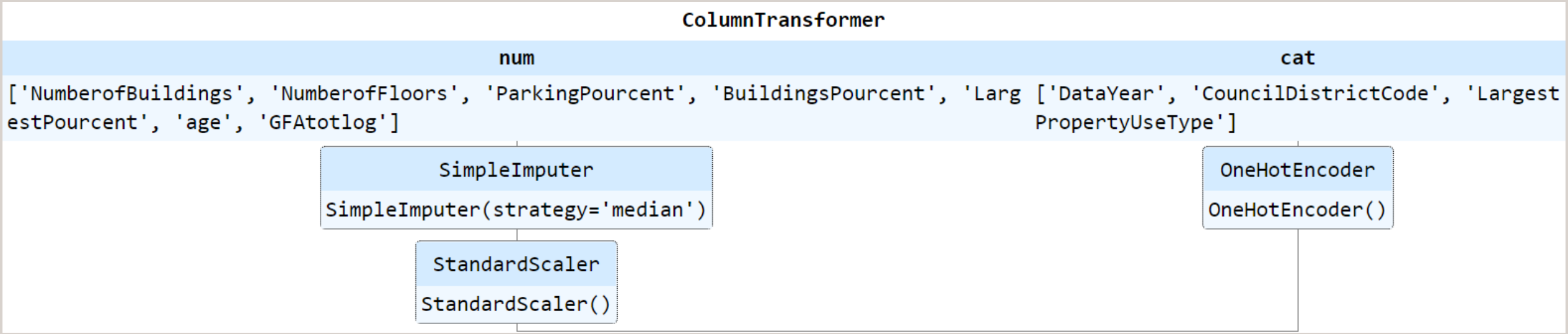






3 Modélisation

3.Pipeline modification données pour entrainement:



Remarque: avec EnergyStarScore lors de la modélisation en prenant compte.

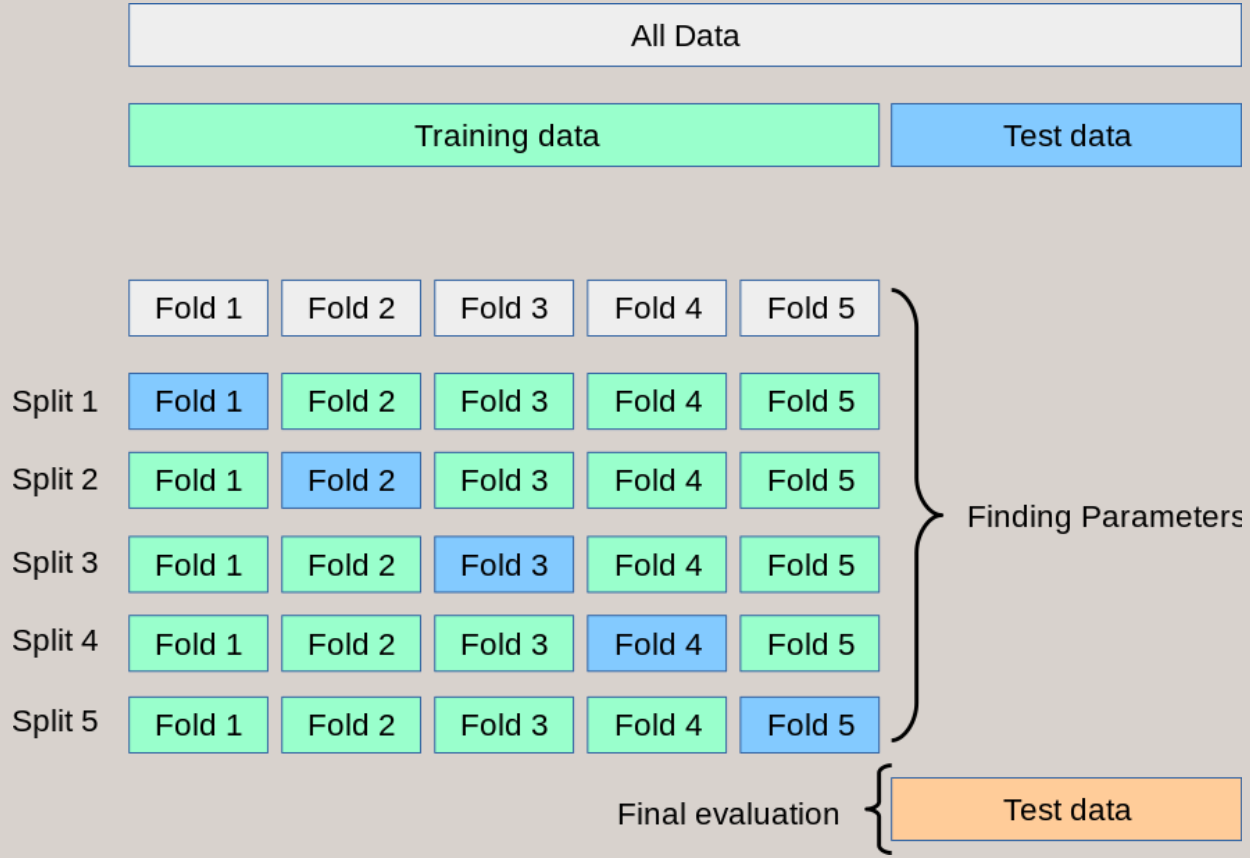
3 Modélisation

4.Différents modèles envisagés et leurs hyperparamètres testés

Modèles évalués et hyperparamètres:

- LinearRegression
- ElasticNet:
  Alpha, max_iter, l1_ratio
- KnnRegressor:
  N_neighbors
- SVR:
  RBF, C, gamma
- RandomForestRegressor:
  Bootstrap, maxdepth, max_features, Min_samples_leaf, min_samples_split N_estimator
- GradientBoostingRegressor:
  Learning rate, max_depth, n_estimator, subsample
- MLPregressor:
  Early_stopping, hidden_layer_size, Learning rate

GridSearch et CrossValidation:



3 Modélisation

5.Choix du modèle le plus performant selon plusieurs critères

Métriques pour comparaison des performances:

- Variance expliquée
- Erreur max
- R2
- RMSE
- MAE

Résultats pour prédiction CO2

	explained_var	test_exp_var	error_max	test_error_max	r2	test_r2	RMSE	test_RMSE	MAE	test_MAE
regression lineaire	0.5136	0.4493	1.5992	1.7049	0.5136	0.4425	0.4278	0.4350	0.3311	0.3364
elasticnet	0.5129	0.4526	1.6110	1.6890	0.5129	0.4459	0.4281	0.4337	0.3324	0.3359
SVR_c100_eps0.1_gamma_0.003	0.5624	0.4663	1.6753	1.7925	0.5621	0.4599	0.4059	0.4282	0.2989	0.3200
KNeighborsRegressor(n_neighbors=12)	0.5510	0.4546	1.8917	1.8843	0.5479	0.4542	0.4124	0.4305	0.3184	0.3284
GradientBoosting_lr_0.01_md_8	0.9968	0.8130	0.2504	1.7984	0.9968	0.8101	0.0349	0.2539	0.0262	0.1598
RandomForestRegressor(max_depth=90, max_features=3, min_samples_leaf=3, min_samples_split=8, n_estimators=200)	0.6844	0.5235	1.5396	1.7724	0.6844	0.5208	0.3446	0.4033	0.2678	0.3081
MLPRegressor(early_stopping=True, hidden_layer_sizes=100, learning_rate='invscaling', verbose=True)	0.6246	0.4977	1.5684	1.7878	0.6246	0.4963	0.3758	0.4135	0.2871	0.3112

Résultats pour prédiction Energie

	explained_var	test_exp_var	error_max	test_error_max	r2	test_r2	RMSE	test_RMSE	MAE	test_MAE
regression lineaire	0.6918	0.5556	1.7282	6.2335	0.6918	0.5524	0.3071	0.4107	0.2178	0.2303
elasticnet	0.6912	0.5569	1.7303	6.2489	0.6912	0.5537	0.3074	0.4101	0.2182	0.2294
SVR(C=100.0, gamma=0.0031622776601683794)	0.7245	0.5695	1.6631	6.2453	0.7245	0.5666	0.2903	0.4041	0.1971	0.2171
KNeighborsRegressor(n_neighbors=12)	0.6859	0.5300	1.8438	6.3437	0.6836	0.5300	0.3111	0.4209	0.2258	0.2421
GradientBoostinginch	0.9992	0.7086	0.0658	6.2788	0.9992	0.7059	0.0160	0.3329	0.0121	0.1245
RandomForestRegressor(max_depth=110, max_features=3, min_samples_leaf=3, min_samples_split=8)	0.7754	0.5528	1.3331	6.3256	0.7754	0.5514	0.2621	0.4112	0.1885	0.2309
MLPRegressor(early_stopping=True, hidden_layer_sizes=1000, learning_rate='invscaling', verbose=True)	0.7858	0.5808	1.5059	6.3000	0.7857	0.5807	0.2560	0.3975	0.1758	0.2154

4 Modèles finals

1.Modèles et hyperparamètres:

2.Comparaison prédiction CO2 avec et sans EnergystarScore

3.Importance des variables du modèle pour CO2

4 Modèles finals

1.Modèles et hyperparamètres:

Modèle pour le CO2

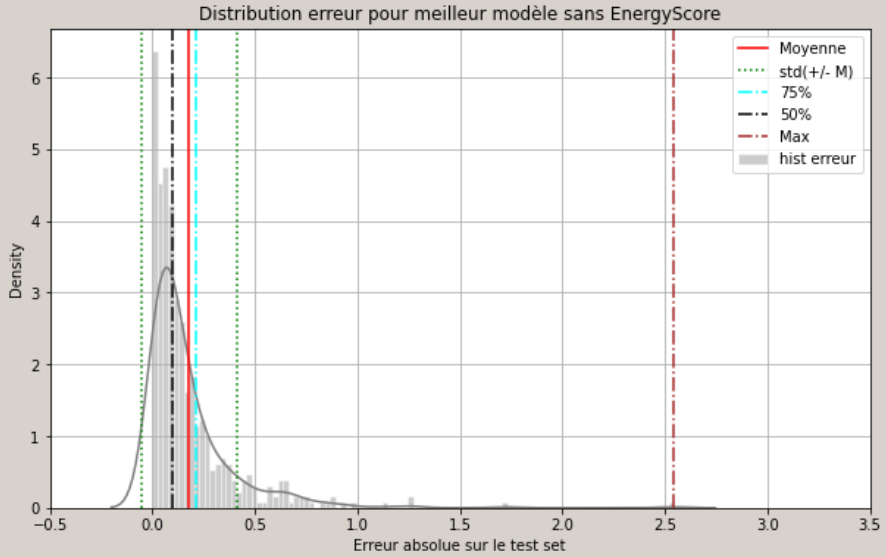
```
GradientBoostingRegressor
GradientBoostingRegressor(learning_rate=0.01, max_depth=8, n_estimators
=1700,
                          subsample=0.85)
```

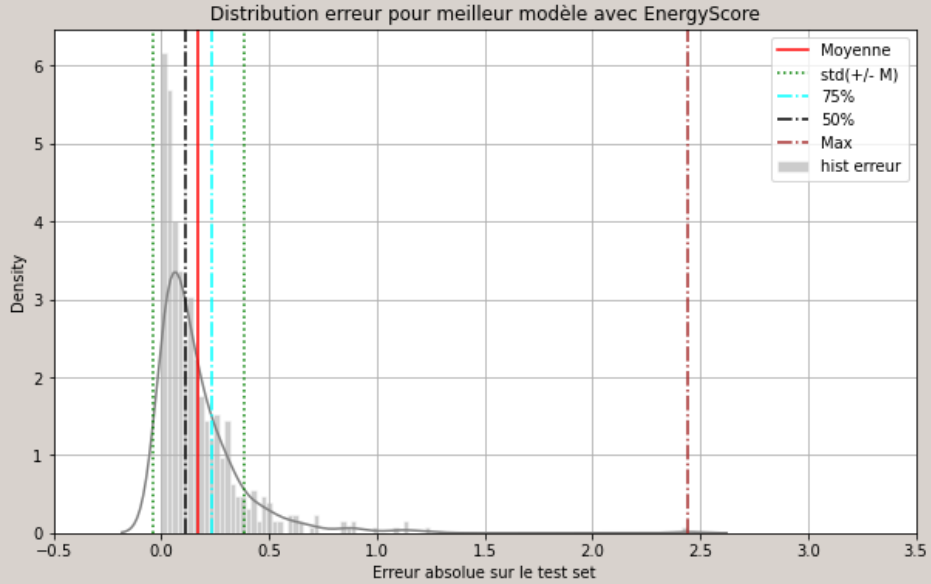
Modèle pour l'énergie

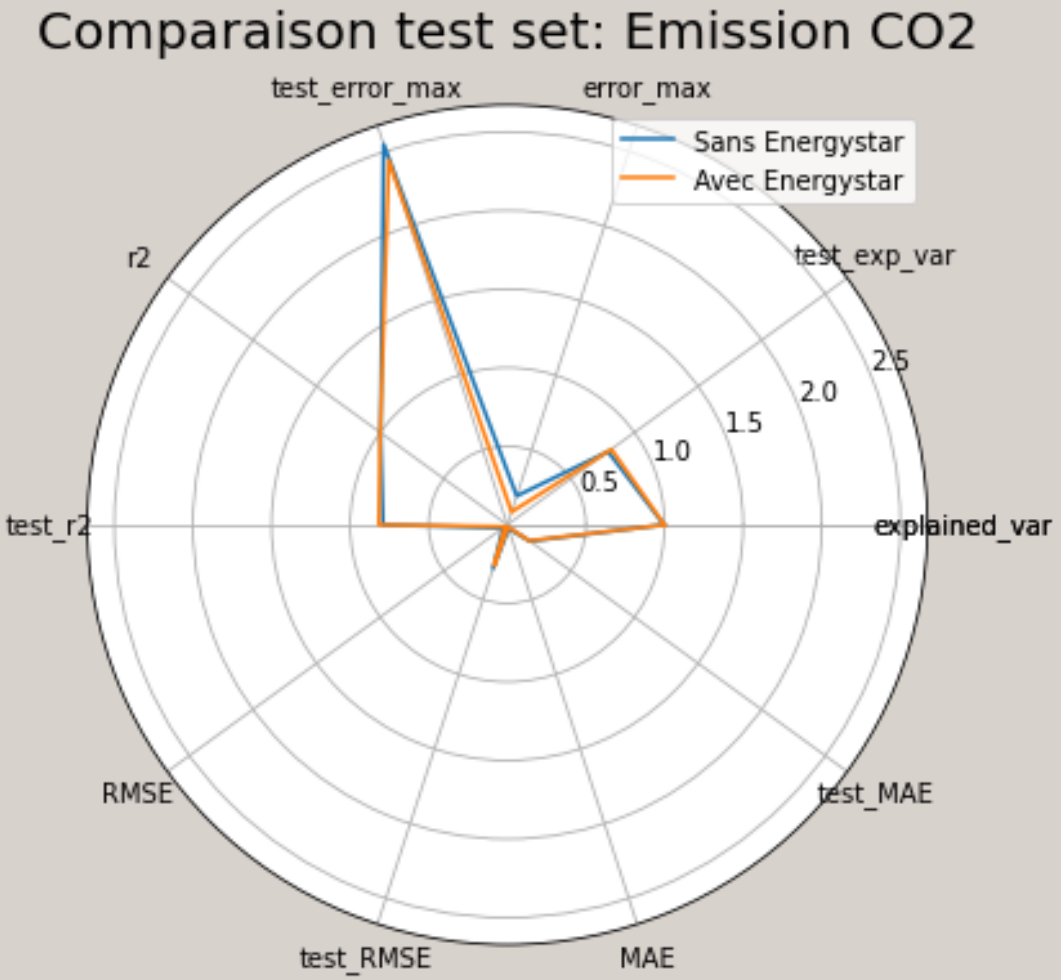
```
GradientBoostingRegressor(learning_rate=0.01, max_depth=8, n_estimators=2000,
                          subsample=0.85)
```

4 Modèles finals

2.Comparaison prédiction CO2 avec et sans EnergystarScore

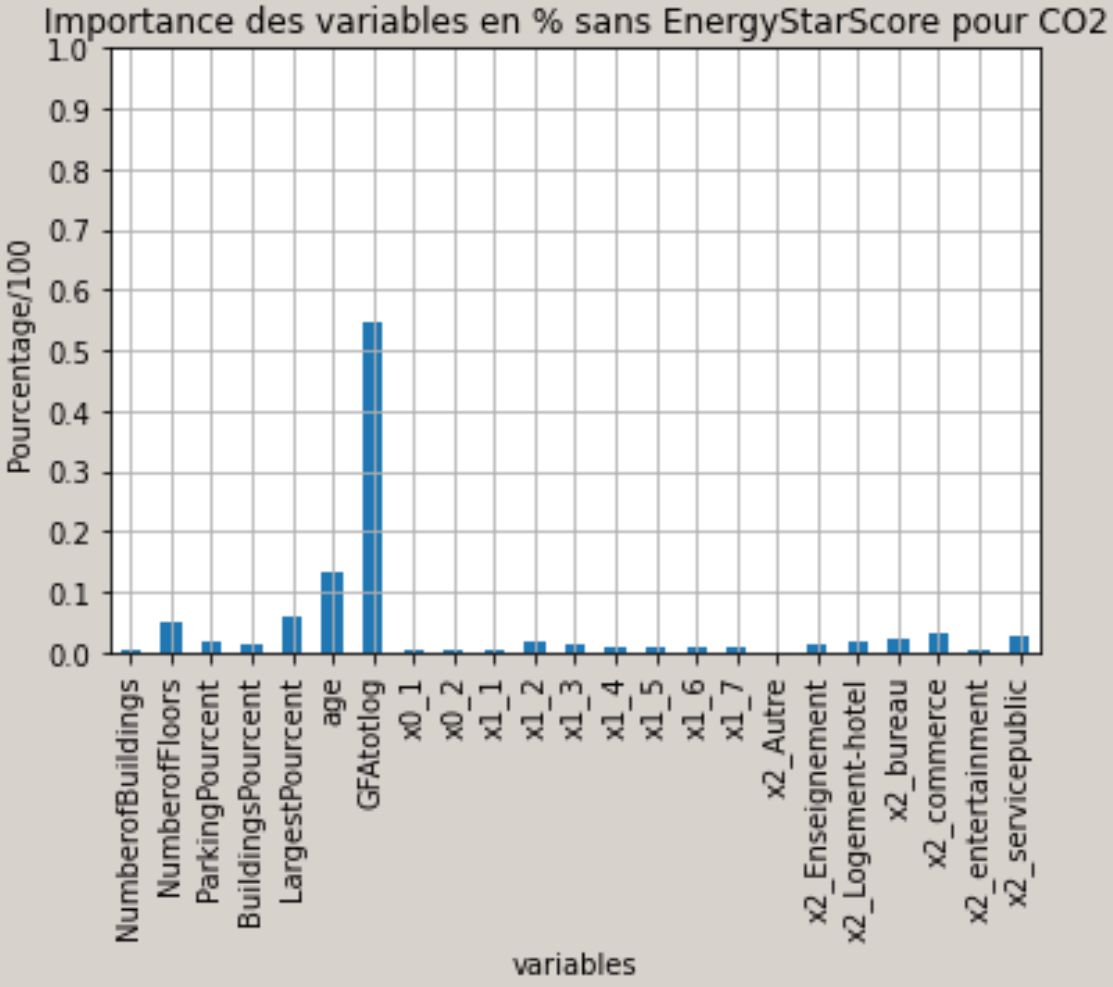


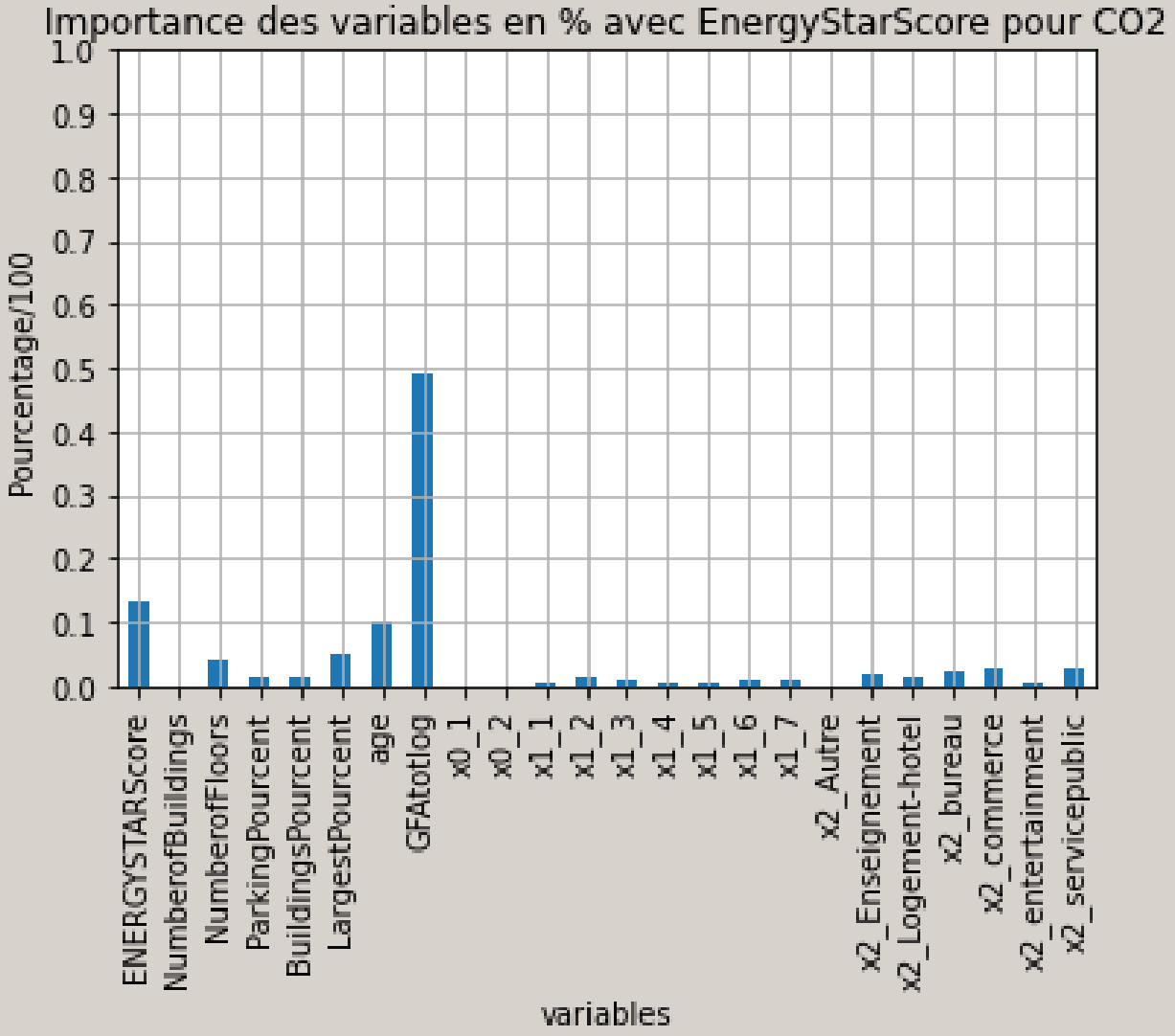




4 Modèles finals

3.Importance des variables du modèle pour CO2





Conclusion

- découverte et transformation du Dataset
- Exploration avec feature engineering, création de nouvelles variables
- Recherche des meilleurs hyperparamètres avec une validation croisée et un gridsearch
- Comparaison des modèles sélectionnés selon plusieurs métriques
- Analyse du modèle le plus performant en l'interprétant

Aller plus loin

- Explorer plus profondément la base de données notamment avec le résultats de la modélisation
- Prendre en compte d'autres variables
- Changer dans pipeline le Centrage/Normalisation (Standardscaler) -> MinMaxScaler
- Recherche encore plus affinée des Hyperparamètres sur le training set

Merci!

Question?