ANSD
Agence Nationale de la Statistique
et de la Démographie

École Nationale de la Statistique
et de l'Analyse Économique

Projet de Machine Learning

Prédiction de l'émission du CO_2 dans la ville de Seattle

Bâtiments Non-Résidentiels

Présentateurs

Aissatou GUEYE
Marianne DAIFERLE
Justin BOGNON
Ndeye Salla TOURE
Mohamadou Dia

Présentatrice

Mme Fatou SALL

Table des matières

Introduction

Contexte scientifique

Ce projet de machine learning s'inscrit dans le cadre de la transition énergétique urbaine aux États-Unis. La ville de Seattle impose depuis plusieurs années le *benchmarking* énergétique de l'ensemble de ses bâtiments commerciaux et institutionnels via le programme *Seattle Building Energy Benchmarking*. Ce programme génére chaque année un dataset public contenant des informations structurelles — surface, date de construction, type d'occupation, géolocalisation — ainsi que des données de consommation énergétique post-opérationnelles.

Notre objectif est de détreminer si les émissions de gaz à effet de serre (GHG) d'un bâtiment peuvent être *anticipées* à partir de ses caractéristiques physiques connues. Une telle capacité permettrait aux autorités municipales d'estimer l'impact climatique d'un projet immobilier *avant* même que le bâtiment soit construit.

Objectifs du projet

L'étude poursuit trois objectifs distincts. Le premier est de construire un modèle prédictif *pur* (Modèle 1) utilisant exclusivement les variables autorisées, sans fuite d'information. Le second est d'évaluer la valeur ajoutée du score ENERGY STAR comme variable supplémentaire (Modèle 2). Le troisième est de quantifier rigoureusement le gain — ou l'absence de gain — entre ces deux scénarios à travers une méthodologie de comparaison contrôlée.

Stratégie anti-data leakage

Le concept central du projet est la **distinction temporelle** entre les variables. Une variable est autorisée uniquement si elle est connue au stade du permis de construction, avant que le bâtiment ne soit mis en service. Cette frontière temporelle dicte la partition des 46 variables du dataset en trois groupes :

— **Variables autorisées** (22) : connues au permis — surface, année de construction, type d'occupation, coordonnées géographiques, nombre d'étages, etc.
— **Variables de consommation** (12) : mesurées seulement après mise en service (kWh, kBtu, EUI…) — **exclues de tout modèle**.
— **ENERGYSTARScore** (2 variables) : nécessite une certification post-construction — utilisé uniquement dans le Modèle 2.

Ce découpage a été validé au cours du projet par la détection d'un épisode de data leakage dans le notebook de base (voir Section 6), ce qui souligne l'importance du contrôle rigoureux de cette frontière.

Description des données

Source et dimensions

Le dataset provient du programme *Seattle Building Energy Benchmarking* (année de déclaration : 2016). Dans sa forme initiale, il contient **3 510 observations** et **46 variables**. Un filtre est ensuite appliqué pour ne retenir que les bâtiments non-résidentiels (types : `NonResidential`, `SPS-District Energy Plant`, `Nonresidential WA`), réduisant la population à **1 666 observations**.

Observation clé

Une observation de type `Nonresidential WA` a été identifiée comme cas particulier correspondant à un musée. Elle est néanmoins conservée dans l'échantillon car elle satisfait les critères de

filtrage standard.

Catégorisation exhaustive des 46 variables

Le tableau 1 présente la répartition complète :

Table 1. Répartition des 46 variables par catégorie

Catégorie	Nombre	Rôle dans M1	Rôle dans M2
Variables autorisées	22	Features	Features
Variables consommation	12	Exclues	Exclues
Variables ENERGY STAR	2	Exclues	Features
Variables identifiant	9	Exclues	Exclues
Variable cible	1	Cible	Cible
Total	**46**		

Liste détaillée des variables autorisées

Table 2. Variables autorisées — détail complet

Variable	Type	Description	Manquance
PropertyGFABuilding(s)	Num.	Surface totale du bâtiment (sf)	0%
PropertyGFATotal	Num.	Surface totale de la proprieté	0%
PropertyGFAParking	Num.	Surface réservée au parking	0%
NumberOfBuildings	Num.	Nombre de bâtiments sur la parcelle	0%
NumberOfFloors	Num.	Nombre d'étages	0%
YearBuilt	Num.	Année de construction	0%
Latitude	Num.	Coordonnée latitude	0%
Longitude	Num.	Coordonnée longitude	0%
BuildingType	Cat.	Type de bâtiment	0%
CouncilDistrict	Cat.	Arrondissement municipal	0%
LargestPropertyUseType	Cat.	Usage principal	0,23%
LargestPropertyUseTypeGFA	Num.	Surface du usage principal	0,23%
SecondLargestPropertyUseType	Cat.	Usage secondaire	50,38%
SecondLargestPropertyUseTypeGFA	Num.	Surface du usage secondaire	50,38%
ThirdLargestPropertyUseType	Cat.	Usage tertiaire	79,58%
ThirdLargestPropertyUseTypeGFA	Num.	Surface du usage tertiaire	79,58%
ListOfAllPropertyUseTypes	Cat.	Liste de tous les usages	0%
*UseType (mode)	Cat.	Mode d'occupation	0%
*UseType (variance)	Cat.	Variabilité d'occupation	0%
*UseType (GFA)	Num.	Surface par type d'occupation	0%
Features ingéniérées (Section 4.7) :			
HasElectricity	Bin.	Présence raccordement électrique	–
HasNaturalGas	Bin.	Présence gaz naturel	–
HasSteam	Bin.	Présence vapeur	–

Num. = numérique, Cat. = catégorielle, Bin. = binaire

Variables de consommation (exclues)

Les 12 variables suivantes sont exclues de tout modèle car elles ne sont mesurables qu'après mise en service : `Electricity(kWh)`, `Electricity(kBtu)`, `NaturalGas(kBtu)`, `NaturalGas(therms)`, `Steam(therms)`, `SiteEUI(kBtu/sf)`, `SiteEUIWN(kBtu/sf)`, `SourceEUI(kBtu/sf)`, `SourceEUIWN(kBtu/sf)`, `GHGEmissionsIntensity` et leurs variants.

Variable cible

La variable cible `TotalGHGEmissions` représente les émissions totales en tonnes métriques de CO_2 par an. Son analyse descriptive est présentée en Section 3.

Analyse exploratoire des données (EDA)

Statistiques descriptives de la variable cible

L'analyse de `TotalGHGEmissions` produit les statistiques suivantes, calculées sur les 1 666 observations avant transformation :

Table 3. Statistiques descriptives — `TotalGHGEmissions`

Statistique	Valeur	Interprétation
Observations	1 666	
Valeurs manquantes	2 (0,1%)	Supprimées en prétraitement
Moyenne ($\bar{y}$)	184,97	Fortement tirée vers le haut
Médiane	49,58	Bâtiment « typique »
Écart-type (σ)	751,98	Très grande dispersion
Minimum	−0,80	Valeur négative (artefact)
Maximum	16 870,98	Grand complexe commercial
Q_1 **(25%)**	19,97	
Q_3 **(75%)**	142,38	
IQR	122,41	$Q_3 - Q_1$
Asymétrie (skewness)	14,21	$\gg 0$: forte asymétrie positive
Kurtosis	247,10	$\gg 0$: queues très lourdes
Rapport $\bar{y}$/méd.	3,73	Confirme l'asymétrie

Observation clé

Deux anomalies sont identifiées. La variable présente une valeur négative (−0,80), physiquement incohérente pour des émissions GHG. Le rapport moyenne/médiane de 3,73, combiné à une asymétrie de 14,21, confirme qu'une transformation est indispensable avant modélisation.

Test de normalité

Un test de Shapiro-Wilk et un test de Kolmogorov-Smirnov sont appliqués sur l'échantillon complet :

Table 4. Résultats des tests de normalité

Test	Distribution	Statistique	p-valeur	Conclusion
Shapiro-Wilk	Originale	0,177035	0,000000	Non-normale
Shapiro-Wilk	Log-transformée	0,996717	0,001287	Non-normale (strict)
Kolmogorov-Smirnov	Originale	–	0,000000	Non-normale
Kolmogorov-Smirnov	Log-transformée	–	0,287358	**Normale**

Le test KS ($p = 0{,}287 > 0{,}05$) valide que $\log_1 p$ rapproche la distribution de la loi normale. Shapiro-Wilk reste significatif ($p = 0{,}0013$) car il est connu pour être extrêmement sensible sur les échantillons supérieurs à 1 000 observations.

Détection des outliers sur l'échelle log

Après transformation, des z-scores sont calculés avec un seuil $|z| > 3$:

Table 5. Valeurs extrêmes détectées sur l'échelle logarithmique ($|z| > 3$)

TotalGHGEmissions	TotalGHGEmissions_log	Z-score
16 870,98	9,7334	3,914
12 307,16	9,4180	3,699
11 140,56	9,3184	3,631
10 734,57	9,2813	3,606
8 145,52	9,0053	3,418
6 330,91	8,7534	3,246
4 906,33	8,4985	3,073

7 outliers détectés, soit 0,42% de l'échantillon.

Observation clé

Décision : aucun traitement supplémentaire. Le pourcentage d'outliers est inférieur à 1%, la skewness après log est faible, et ces valeurs extrêmes correspondent à de grands bâtiments commerciaux contenant de l'information utile. La transformation $\log_1 p$ seule suffit à normaliser la distribution.

Analyse des valeurs manquantes

La catégorisation est réalisée sur le jeu d'entraînement (1 333 observations) :

Table 6. Catégorisation des variables selon leur taux de manquance

Variable	Manquance	Seuil	Stratégie
Groupe 1 — Suppression ($> 70\%$) : 3 variables			
`YearsENERGYSTARCertified`	94,59%	$> 70\%$	Supprimée
`ThirdLargestPropertyUseType`	79,58%	$> 70\%$	Supprimée
`ThirdLargestPropertyUseTypeGFA`	79,58%	$> 70\%$	Supprimée
Groupe 2 — Test MCAR puis imputation (10–70%) : 3 variables			
`SecondLargestPropertyUseType`	50,38%	10–70%	MCAR → imputation
`SecondLargestPropertyUseTypeGFA`	50,38%	10–70%	MCAR → imputation
`ENERGYSTARScore`	34,31%	10–70%	MCAR → imputation
Groupe 3 — Imputation simple ($< 10\%$) : 2 variables			
`LargestPropertyUseType`	0,23%	$< 10\%$	Mode
`LargestPropertyUseTypeGFA`	0,23%	$< 10\%$	Moyenne

Corrélations entre variables autorisées

L'analyse de corrélation est menée exclusivement sur les variables autorisées. `PropertyGFABuilding(s)` montre la corrélation la plus forte avec la cible ; `Latitude` et `Longitude` capturent les effets de localisation géographique ; les variables d'occupation (mode, variance, GFA) présentent des corrélations inter-variables élevées, d'où leur nettoyage decrit en Section 4.

Prétraitement des données

Principe fondamental : isolation temporelle du test

Attention — Data Leakage

La séparation train/test est effectuée **avant** toute transformation des données. Aucune statistique de centrage, de variance, ni d'imputation ne provient du jeu de test. Ce principe est respecté strictement à chaque étape du pipeline.

Traitement de la variable cible

Le traitement suit quatre étapes séquentielles, chacune validée par une vérification statistique.

Étape 1 — Suppression des valeurs manquantes. Les 2 observations avec `TotalGHGEmissions` manquant sont retirées, portant le dataset de 1 668 à 1 666 lignes.

Étape 2 — Conversion des valeurs négatives. La valeur −0,80 est remplacée par zéro pour garantir la validité de la transformation logarithmique.

Étape 3 — Transformation $\log_1 p$.

$$y^{\log} = \log_1 p(y) = \log(1 + y) \tag{1}$$

Le choix de $\log_1 p$ plutôt que log permet de gérer les zéros sans singularité. L'opération inverse, utilisée pour retourner à l'échelle originale, est :

$$y = \exp(y^{\log}) - 1 \tag{2}$$

Étape 4 — Vérification. Les résultats du test de Shapiro-Wilk et Kolmogorov-Smirnov sont présentés dans le Tableau 4.

Séparation Train / Test

```python
from sklearn.model_selection import train_test_split

TEST_SIZE = 0.20
RANDOM_STATE = 42

train_df, test_df = train_test_split(
    df,
    test_size=TEST_SIZE,
    random_state=RANDOM_STATE
)
```

Listing 1. Séparation train/test avec random_state fixé

Table 7. Répartition train/test

Jeu de données	Observations	Proportion
Entraînement (`train`)	1 333	80%
Test (`test`)	333	20%
Total	**1 666**	100%

Test de Little — Classification du mécanisme de manquance

Pour chaque variable entre 10% et 70% de manquance, un test de Little est appliqué. Le test repose sur une statistique χ^2 :

$$\chi^2 = n \cdot \bar{d}^\top \, \Sigma^{-1} \, \bar{d} \tag{3}$$

où $\bar{d}$ est le vecteur des moyennes des écarts et Σ la matrice de covariance estimée sur les données complètes. Si $p > 0{,}05$, le mécanisme est classifié MCAR.

Table 8. Résultats du test de Little (MCAR)

Variable	Manquance	Statistique	p-valeur	Conclusion
SecondLargestPropertyUseType	50,38%	0,0000	1,0000	MCAR
SecondLargestPropertyUseTypeGFA	50,38%	0,0000	1,0000	MCAR
ENERGYSTARScore	34,31%	0,0000	1,0000	MCAR

Les trois variables sont classifiées MCAR ($p > 0{,}05$) : imputation simple autorisée.

Détection des outliers — Approche bicritère

Méthode univariée (IQR) :

$$x_i < Q_1 - 1{,}5 \times \text{IQR} \quad \text{ou} \quad x_i > Q_3 + 1{,}5 \times \text{IQR} \tag{4}$$

avec $\text{IQR} = Q_3 - Q_1$.

Méthode multivariée : trois mesures sont calculées pour chaque observation :

$$\text{Leverage :} \quad h_{ii} > \frac{2p}{n} \tag{5}$$

$$\text{Distance de Cook :} \quad D_i > \frac{4}{n} \tag{6}$$

$$\text{DFFITS :} \quad |\text{DFFITS}_i| > 2\sqrt{\frac{p}{n}} \tag{7}$$

où p est le nombre de predicteurs et n le nombre d'observations.

Winsorisation

Les valeurs extrêmes sont atténuées aux percentiles 5e et 95e :

$$x_i^w = \begin{cases} P_5 & \text{si } x_i < P_5 \\ P_{95} & \text{si } x_i > P_{95} \\ x_i & \text{sinon} \end{cases} \tag{8}$$

Suppression des variables redondantes

Table 9. Variables supprimées et motifs

Variable	Motif	Groupe
NaturalGas(kBtu)	Consommation — leakage	Consommation
SiteEUI(kBtu/sf)	Consommation — leakage	Consommation
SiteEUIWN(kBtu/sf)	Consommation — leakage	Consommation
GHGEmissionsIntensity	Consommation — leakage	Consommation
Electricity(kBtu)	Consommation — leakage	Consommation
TaxParcelIdentificationNumber	Identifiant unique	ID
OSEBuildingID	Identifiant unique	ID
ZipCode	Identifiant géographique	ID
CouncilDistrictCode	Identifiant administratif	ID

Ingénierie de features

Trois features binaires sont crées pour capturer la *diversité* des systèmes énergétiques :

```
binary_features = {
    'HasElectricity' : 'Electricity(kWh)',
    'HasNaturalGas' : 'NaturalGas(therms)',
```

```
    'HasSteam' : 'Steam(therms)'
}

for new_col, source_col in binary_features.items():
    df[new_col] = df[source_col].notna().astype(int)
```

Listing 2. Création des features binaires

Module `preprocessing_advanced.py`

```
from src.preprocessing_advanced import (
    little_mcar_test,
    fit_missing_values_pipeline_advanced,
    transform_missing_values_pipeline_advanced,
    detect_outliers_univariate,
    detect_outliers_multivariate,
    fit_outliers_pipeline_advanced,
    transform_outliers_pipeline_advanced
)
```

Listing 3. API du module de prétraitement

Stratégie de modélisation

Algorithmes évalués

Table 10. Algorithmes, familles et paramètres clés

Algorithme	Famille	Type	Paramètre principal
Ridge Regression	Linéaire	Régularisé	α (L2)
Lasso Regression	Linéaire	Régularisé	α (L1)
Random Forest	Ensemble	Bagging	`n_estimators`, `max_depth`
Gradient Boosting	Ensemble	Boosting séquentiel	`learning_rate`, `n_estimators`
SVR	Noyau	Régression à vecteurs support	C, ϵ

Architecture du pipeline

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import (
    RandomForestRegressor, GradientBoostingRegressor
)
from sklearn.linear_model import Ridge, Lasso
from sklearn.svm import SVR

pipelines = {
    'Ridge' : Pipeline([
        ('scaler', StandardScaler()),
        ('model', Ridge())
    ]),
    'Lasso' : Pipeline([
        ('scaler', StandardScaler()),
        ('model', Lasso())
    ]),
    'Random Forest' : Pipeline([
        ('scaler', StandardScaler()),
        ('model', RandomForestRegressor(
            n_estimators=100, random_state=42, n_jobs=-1))
    ]),
    'Gradient Boosting' : Pipeline([
```

```
        ('scaler', StandardScaler()),
        ('model', GradientBoostingRegressor(
            n_estimators=100, random_state=42))
    ]),
    'SVR' : Pipeline([
        ('scaler', StandardScaler()),
        ('model', SVR(kernel='rbf'))
    ])
}
```

Listing 4. Structure du pipeline modèle

Le `StandardScaler` garantit que Ridge, Lasso et SVR opérent sur des données centrées ($\mu = 0$) et normalisées ($\sigma = 1$), calculées *uniquement* sur `train`.

Validation croisée 5-folds

$$\text{Score}_{CV} = \frac{1}{K}\sum_{k=1}^{K} \text{Score}\left(D_{\text{test}}^{(k)}\right), \quad K = 5 \tag{9}$$

Métriques d'évaluation

Coefficient de détermination :

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{10}$$

RMSE sur l'échelle log :

$$\text{RMSE}_{\log} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\log} - \hat{y}_i^{\log}\right)^2} \tag{11}$$

RMSE sur l'échelle originale (après inversion $\hat{y}_i = e^{\hat{y}_i^{\log}} - 1$) :

$$\text{RMSE}_{\text{orig}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad [\text{tonnes } CO_2] \tag{12}$$

MAPE :

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i|} \ \% \tag{13}$$

Taux d'overfitting :

$$\text{OF} = R^2_{\text{train}} - R^2_{\text{test}} \tag{14}$$

Un taux inférieur à 0,02 est considéré comme acceptable.

Résultats de base (Baseline)

Détection du data leakage

Attention — Data Leakage

Les résultats du notebook 04 montrent $R^2_{\text{test}} = 1{,}0000$ pour les deux modèles. L'analyse des

importances révèle que `TotalGHGEmissions` (la variable cible elle-même) domine à **99,998%**. Elle a été incluse par erreur dans les features.

Cette fuite invalide le notebook 04. Le notebook 05 utilise un jeu de features corrigé (`train_with_features.csv`) et les résultats de la **Section 7 sont les seuls résultats fiables**.

Résultats complets — Modèle 1 (sans ENERGY STAR)

Table 11. Baseline — Modèle 1 : tous les algorithmes

Algorithme	R^2 test	$\text{RMSE}_{\log}$	$\text{MAE}_{\log}$	$\text{RMSE}_{\text{orig}}$ (tonnes)	MAPE (%)	OF
Random Forest	1,0000	0,0043	0,0029	0,8451	0,3047	0,0000
Gradient Boosting	0,9999	0,0118	0,0087	1,8053	0,9077	0,0000
Ridge	0,6359	0,7769	0,6115	94,9668	94,9108	0,3640
SVR	0,1914	1,1578	0,9643	121,6794	70,1355	0,6966
Lasso	0,1390	1,1948	0,9879	155,1648	168,6580	−0,0033

Note : RF et GB sont biasés par la fuite. Temps : Ridge 8,08 s — Lasso 1,32 s — RF 14,53 s — GB 11,32 s — SVR 25,88 s.

Résultats complets — Modèle 2 (avec ENERGY STAR)

Table 12. Baseline — Modèle 2 : tous les algorithmes

Algorithme	R^2 test	$\text{RMSE}_{\log}$	$\text{MAE}_{\log}$	$\text{RMSE}_{\text{orig}}$ (tonnes)	MAPE (%)	OF
Random Forest	1,0000	0,0044	0,0029	0,8242	0,3072	0,0000
Gradient Boosting	0,9999	0,0118	0,0087	1,8046	0,9078	0,0000
Ridge	0,6344	0,7786	0,6144	94,2632	94,8289	0,3656
SVR	0,1946	1,1555	0,9657	122,9092	69,8791	0,6958
Lasso	0,1390	1,1948	0,9879	155,1648	168,6580	−0,0033

Temps : Ridge 2,91 s — Lasso 1,32 s — RF 16,65 s — GB 12,42 s — SVR 24,82 s.

Comparaison baseline M1 vs M2

Table 13. Écart M2 − M1 (algorithme Random Forest)

Métrique	M1	M2	Écart
R^2 **test**	1,0000	1,0000	0,0000
$\textbf{RMSE}_{\log}$	0,0043	0,0044	+0,0001 (+1,07%)
$\textbf{RMSE}_{\textbf{orig}}$ **(tonnes)**	0,8451	0,8242	−0,0209 (−2,5%)

ENERGY STAR n'améliore pas la prédiction au niveau baseline.

Dans le Modèle 2, `ENERGYSTARScore` se place au rang 8 avec une importance de $1{,}55 \times 10^{-6}$, confirmant sa contribution négligeable.

Optimisation des hyperparamètres

Cette section présente les **résultats fiables** du projet, obtenus avec les features corrigées.

Méthodologie

GridSearchCV effectue une recherche exhaustive sur Random Forest pour M1 : chaque combinaison est évalué en validation croisée 5-folds. **RandomizedSearchCV** réalise 100 itérations sur Random Forest et Gradient Boosting pour M1 et M2.

Espaces de recherche

Table 14. Espace de recherche — Random Forest

Hyperparametre	Valeurs testées
n_estimators	{50, 100, 150, 200}
max_depth	{5, 10, 15, 20, None}
min_samples_split	{2, 5, 8, 10, 15}
min_samples_leaf	{1, 2, 3, 4}
max_features	{'sqrt', None}
bootstrap	{True, False}

GridSearch (M1) : $4 \times 5 \times 5 \times 4 \times 2 \times 2 = 1\,600$ combinaisons.

Table 15. Espace de recherche — Gradient Boosting

Hyperparametre	Valeurs testées
n_estimators	{50, 100, 150, 200}
learning_rate	Uniforme $\mathcal{U}[0{,}001;\ 0{,}05]$
max_depth	{3, 4, 5, 6}
subsample	Uniforme $\mathcal{U}[0{,}6;\ 0{,}8]$
max_features	{'sqrt', 'log2', None, 0,3}

RandomizedSearch : 100 itérations par (algorithme, modèle).

Résultats — Top 5 modèles optimíses

Table 16. Classement des 5 meilleurs modèles après optimisation

#	Modèle	Méthode	R^2 test	RMSE (t)	MAPE (%)	OF
1	**M2—GB**	**Random**	**0,9881**	**43,93**	**7,28**	$< 0{,}015$
2	M1—GB	Random	0,9870	45,50	7,53	$< 0{,}015$
3	M2—RF	Random	0,9844	48,93	7,58	$< 0{,}015$
4	M1—RF	Grid	0,9840	49,94	7,70	$< 0{,}015$
5	M1—RF	Random	0,9839	50,42	7,49	$< 0{,}015$

GB = Gradient Boosting, RF = Random Forest, t = tonnes CO_2.

Analyse comparative

Gradient Boosting vs Random Forest

Apres optimisation, Gradient Boosting surpasse Random Forest de manière consistante. La nature séquentielle de GB explique ce résultat : chaque arbre corrige les erreurs résiduelles du précédent. L'écart est de l'ordre de 2–3 tonnes sur le RMSE.

Impact du score ENERGY STAR (optimísé)

Table 17. Gain M2 vs M1 — Gradient Boosting optimísé

Métrique	M1—GB	M2—GB	Gain
R^2 **test**	0,9870	0,9881	+0,0011 (+0,11%)
RMSE (tonnes)	45,50	43,93	−1,57 (−3,4%)
MAPE (%)	7,53	7,28	−0,25 (−3,3%)

Le gain est marginal et ne justifie pas la contrainte de certification ES.

Impact de l'optimisation

L'optimisation réduit le RMSE d'environ 52 tonnes (baseline corrigé, RF) à 43,93 tonnes (champion), soit une amélioration de ≈ 15%. Le taux d'overfitting reste inférieur à 0,015 pour tous les modèles optimíses.

Conclusions et recommandations

Conclusions principales

(1) Performances élevées. Le modèle champion (M2—Gradient Boosting optimísé) atteint $R^2 = 0{,}9881$ et RMSE = 43,93 tonnes CO_2, indiquant que les caractéristiques structurelles et géographiques expliquent plus de 98,8% de la variance des émissions.

(2) ENERGY STAR superflu. Le gain est de +0,11% en R^2. Le Modèle 1, utilisable dès le permis, sufflit pour une prédiction fiable.

(3) Déploiement au permis. Le Modèle 1 ne nécessite aucune donnée post-construction et est directement applicable à la phase de permis.

(4) Vigilance sur le leakage. Le notebook de base contenait une fuite détectable par l'analyse des importances (99,998% pour la cible elle-même).

Guide de sélection

Table 18. Recommandations de déploiement

Scénario	Modèle	R^2	RMSE (t)
Nouveau bâtiment (permis)	M1—GB optimísé	0,9870	45,50
Bâtiment avec certif. ES	M2—GB optimísé	0,9881	43,93
Comparaison impact ES	M1 vs M2	–	Gain < 4%

Perspectives

L'optimisation bayésienne (Optuna) pourrait affiner davantage les hyperparamètres. Les algorithmes XGBoost, LightGBM et CatBoost sont des candidats modernes particulièrement adaptés aux données tabulaires. Si des données sur plusieurs années sont disponibles, une validation temporelle renforcera la crédibilité. La sérialisation via `joblib` permet un déploiement en API REST.

Structure du projet

Appendice Régularisation Ridge et Lasso

Ridge (L_2) :

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \alpha \|\boldsymbol{\beta}\|_2^2 \tag{15}$$

La pénalisation L_2 réduit la magnitude de tous les coefficients sans en annuler aucun.

Lasso (L_1) :

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \alpha \|\boldsymbol{\beta}\|_1 \tag{16}$$

La pénalisation L_1 pousse certains coefficients à zéro. Le résultat de Lasso baseline ($R^2 = 0{,}1390$) suggère que cette sélection agressive supprime des features informatives.

Table 19. Arborescence du projet REGO3

Répertoire / Fichier	Description
`data/raw_data/base_raw.csv`	Dataset brut filtré
`data/interim_data/train_with_features.csv`	Features corrigées (optim.)
`data/interim_data/test_with_features.csv`	Features test corrigées
`data/processed_data/train_processed.csv`	Jeu entraînement final
`data/processed_data/test_processed.csv`	Jeu test final
`models/pipeline_modele1_best.pkl`	M1 baseline sérialisé
`models/pipeline_modele2_best.pkl`	M2 baseline sérialisé
`models/pipeline_m1_gb_optimized.pkl`	M1 GB optimísé
`models/pipeline_m2_gb_optimized.pkl`	**Champion** — M2 GB optimísé
`results/figures/`	Visualisations EDA & modélisation
`results/optimization/`	Résultats optim. (.json, .csv, .png)
`notebooks/`	Exports HTML des 4 notebooks
`src/`	Modules Python utilitaires

Table 20. Modules utilitaires

Module	Responsabilités
`preprocessing_advanced.py`	Test MCAR (Little), imputation, outliers univarié/-multivarié, Winsorisation
`modeling_utils.py`	Entraînement multi-algorithmes, sauvegarde joblib
`evaluation_utils.py`	R^2, RMSE, MAE, MAPE, overfitting, résumé
`visualization_utils.py`	EDA, comparaisons, résidus

Table 21. Workflow séquentiel des notebooks

Notebook	Titre	Livrables
01	Exploration & Catégorisation	Stats cible, 46 vars catégorisées
02	Prétraitement	Pipeline imputation, outliers, features
04	Modélisation	Baseline 5 algo, analyse features
05	Optimisation	Grid/RandomSearch, modèle champion

Appendice Formules de référence rapide

$$\text{Transformation :} \quad y^{\log} = \log(1+y), \qquad y = e^{y^{\log}} - 1 \tag{B.1}$$

$$\text{Z-score :} \quad z_i = \frac{x_i - \mu}{\sigma} \tag{B.2}$$

$$\text{IQR :} \quad \text{IQR} = Q_3 - Q_1 \tag{B.3}$$

$$\text{Bornes IQR :} \quad [Q_1 - 1{,}5\,\text{IQR};\ Q_3 + 1{,}5\,\text{IQR}] \tag{B.4}$$

$$\text{Winsorisation :} \quad x_i^w = \max(P_5,\ \min(P_{95},\ x_i)) \tag{B.5}$$

$$\text{Leverage :} \quad h_{ii} > \frac{2p}{n} \tag{B.6}$$

$$\text{Cook :} \quad D_i > \frac{4}{n} \tag{B.7}$$

Appendice Fichiers de sortie

Table 22. Ensemble des fichiers de modèles et résultats

Fichier	Description
`pipeline_modele1_best.pkl`	M1 baseline — Random Forest
`pipeline_modele2_best.pkl`	M2 baseline — Random Forest
`pipeline_m1_gb_optimized.pkl`	M1 — Gradient Boosting optimísé
`pipeline_m2_gb_optimized.pkl`	**Champion** — M2 GB optimísé
`pipeline_m1_rf_grid_optimized.pkl`	M1 — RF GridSearchCV
`pipeline_m1_rf_random_optimized.pkl`	M1 — RF RandomizedSearchCV
`pipeline_m2_rf_random_optimized.pkl`	M2 — RF RandomizedSearchCV
`metrics_comparison.json`	Métriques baseline détaillées
`predictions_finales.csv`	Prédictions sur le test set
`optimization/results_*.json`	Résultats par optimisation
`optimized_models_comparison.csv`	Comparaison tous modèles optimísés
`optimization_final_summary.json`	Synthèse finale