PROJET COMPAGNON IMMOBILIER MAR25

**Groupe:**

* Yasmine Peiffer
* Loick Dernoncourt
* Christophe Egea
* Maxime Hénon



# Table des matières

[Description du projet 4](#_Toc199240685)

[Partie 1 : Exploration - Preprocessing - Split 5](#_Toc199240686)

[Les données à disposition 5](#_Toc199240687)

[Premières intuitions sur les données 6](#_Toc199240688)

[Présélection (Variables explicatives à éliminer) 10](#_Toc199240689)

[Prétraitements : partie 1 12](#_Toc199240690)

[Prétraitements : partie 2 (Gestion des valeurs aberrantes AVANT le split train et test) 14](#_Toc199240691)

[Exploration des problématiques du projet 15](#_Toc199240692)

[Prétraitements : partie 3 (Gestion des valeurs extrêmes APRÈS le split train et test) 16](#_Toc199240693)

[Vérifications visuelles des distributions pour gestion des valeurs aberrantes et des valeurs extrêmes 16](#_Toc199240694)

[Partie 2 : Encodages - Feature Selection - Split X/y 18](#_Toc199240695)

[Sélection de Features 18](#_Toc199240696)

[Partie 3.1 : Machine Learning – séparation du groupe en 2 équipes 19](#_Toc199240697)

[Modélisation SARIMAX 20](#_Toc199240698)

[Contexte 20](#_Toc199240699)

[Classification du Problème 24](#_Toc199240700)

[Choix du Modèle et Optimisation 24](#_Toc199240701)

[Interprétation des Résultats et Diagnostics 25](#_Toc199240702)

[Visualisations 25](#_Toc199240703)

[Predictions SARIMAX 27](#_Toc199240704)

[Evaluation du modèle 28](#_Toc199240705)

[Partie 3.2 : Estimation du prix au m2 31](#_Toc199240706)

[**Préambule et objectif:** 31](#_Toc199240707)

[Objectif 31](#_Toc199240708)

[Métrique de performance 31](#_Toc199240709)

[Choix du modèle et optimisation 32](#_Toc199240710)

[Modèles retenus 32](#_Toc199240711)

[Optimisation des Hyperparamètres 32](#_Toc199240712)

[Gestion du Surapprentissage 33](#_Toc199240713)

[Approche par Clusters Territoriaux 34](#_Toc199240714)

[Performances des Modèles 34](#_Toc199240715)

[Partie 4 : Modélisation – Approfondissement et Interprétabilité 35](#_Toc199240716)

[4.1 Interprétation des modèles 35](#_Toc199240717)

[4.2 Valeurs SHAP et interprétabilité globale 36](#_Toc199240718)

[4.3 Limites de l’approche actuelle 37](#_Toc199240719)

# Description du projet

Développer une solution permettant aux acheteurs de logements d'explorer et de comparer différents territoires en termes de prix de l'immobilier, démographie, transports, services, éducation, criminalité, et économie.

L'application doit offrir une Data Visualization, permettant aux utilisateurs d'établir des classements et de visualiser les forces et faiblesses relatives des territoires.

Deux objectifs principaux sont à considérer :

1. La prédiction de l'évolution du prix des logements selon les territoires.
2. L’estimation du prix au m2 d'un logement donné. Une première prédiction peut être effectuée à l’aide des données tabulaires disponibles, et peut être approfondie avec des données relatives à l’annonce du bien, comme le texte descriptif ou les photos du logement.

L'objectif global est d'aider les acheteurs à prendre des décisions éclairées en traduisant des données complexes et nombreuses en informations utiles et accessibles.

# Partie 1 : Exploration - Preprocessing - Split

### Les données à disposition

Nous avons à disposition les sources suivantes :

* ech\_annonces\_ventes\_68.csv (département 68).
  + Période de 2019 à nos jours
  + 59 colonnes
* ech\_annonces\_locations\_68.csv (département 68).
  + Période de 2019 à nos jours
  + 52 colonnes
* Le fichier DVF (demandes des valeurs foncières)
  + Tous les départements sauf le 68
  + Période de 2019 à nos jours
  + 40 colonnes

### Premières intuitions sur les données

2 fichiers de structure similaire :

* ech\_annonces\_ventes\_68.csv (département 68).
* ech\_annonces\_locations\_68.csv (département 68).

2 fichiers faisant en rapport avec les ventes :

* ech\_annonces\_ventes\_68.csv (département 68).
* Le fichier DVF (demandes des valeurs foncières)

Les fichiers « ech\_annonces\_ventes\_68.csv » et « DVF » nous ont amené à déduire la variable cible comme étant le prix de vente au m².

La compréhension de cette nouvelle variable nous a contraint à réfléchir quant à l’exploitation de ces 2 fichiers.

**Fichier « DVF »**

Le fichier DVF présente des opérations de vente officielles et détaillées. C’est-à-dire, que plusieurs lignes à l’intérieur de celui-ci, peuvent correspondre au détail de l’achat d’un bien.

Par exemple, un corps de ferme avec la maison d'habitation, deux dépendances et deux parcelles, correspond à 5 lignes dans le fichier DVF. Par ailleurs, chacune des lignes portant sur l’opération d’achat de ce corps de ferme en question, sont enregistrées avec la valeur totale du bien.

Le fichier présente également la particularité de recenser l’ensemble des opérations de vente sur l’ensemble des départements français, DOM TOM inclus, mais département 68 exclu.

**Fichier « ech\_annonces\_ventes\_68 »**

Ce fichier présente les annonces de vente de biens immobiliers de manière plus commode que le fichier DVF. En effet, chaque ligne correspond au détail du bien, étage, balcon etc. ainsi que le prix de vente total et le prix de vente au m²

**Le fichier « merged\_sales\_data.csv »**

Malgré la précision et l’exactitude des informations que pouvait contenir le fichier « DVF », celui-ci présentait un travail substantiel qui nous était difficile d’évaluer en termes de temp et compte tenu des échéances de livraison du projet. Les traitements pour déduire un prix de vente au m² pour un bien immobilier.

Cette réflexion nous a donc dirigé vers le fichier d’annonce de vente. Le fichier de location n’ayant pas de rapport avec notre problématique de prix de vente au m².

En remontant à la source créatrice de ce fichier d’annonce de vente pour le département 68, nous nous sommes rendu compte qu’il était possible d’avoir ce même fichier mais pour l’ensemble des départements français.

Cette découverte nous a permis de développer le fichier « merged\_sales\_data.csv ».

Ce fichier est la fusion des fichiers concernant tous les départements sur les 5 dernières années

Ce fichier brut compte 9849326 lignes et 59 colonnes et pèse 2,07 Go.

Par la suite ces données seront enrichies notamment grâce :

* L’API des DPE (Diagnostics des performances Energétiques)
* Données INSEE

### Exploration

Cette partie reprend la structure du notebook, duquel ont été extraits quelques visualisations :

Déroulé :

1. Extrait et taille du dataset
2. Elimination des doublons
3. Gestion des NANs
   * Calcul des proportions
   * Visualisation des proportions par colonnes
   * Elimination des colonnes ayant plus de 75% de NANs:

Colonnes conservées :

['idannonce', 'type\_annonceur', 'typedebien', 'typedetransaction', 'etage', 'surface', 'surface\_terrain', 'nb\_pieces', 'prix\_bien', 'mensualiteFinance', 'balcon', 'eau', 'bain', 'dpeL', 'dpeC', 'mapCoordonneesLatitude', 'mapCoordonneesLongitude', 'annonce\_exclusive', 'nb\_etages', 'places\_parking', 'cave', 'exposition', 'ges\_class', 'annee\_construction', 'nb\_toilettes', 'porte\_digicode', 'ascenseur', 'nb\_logements\_copro', 'charges\_copro', 'chauffage\_energie', 'chauffage\_systeme', 'chauffage\_mode', 'categorie\_annonceur', 'logement\_neuf', 'duree\_int', 'typedebien\_lite', 'date', 'INSEE\_COM', 'IRIS', 'CODE\_IRIS', 'TYP\_IRIS\_x', 'TYP\_IRIS\_y', 'GRD\_QUART', 'UU2010', 'REG', 'DEP', 'loyer\_m2\_median\_n6', 'nb\_log\_n6', 'taux\_rendement\_n6', 'loyer\_m2\_median\_n7', 'nb\_log\_n7', 'taux\_rendement\_n7', 'prix\_m2\_vente']

**Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.Note :** le taux de 75% semblait ne pas compromettre la qualité d’information

### Dataviz

Pour la dataviz, nous avons choisi de suivre les préconisations de notre mentor :

1. Visualisation des modalités des variables catégorielles à moins de 10 modalités

Une image contenant texte, diagramme, capture d’écran, Rectangle

Le contenu généré par l’IA peut être incorrect.

1. Autres variables (plus de 10 modalités)

***Remarques*** : certaines variables (comme chauffage\_energie) ont un très grand nombre de modalités, malgré le fait qu’elles soient catégorielles. Nous avons cherché à les discrétiser.

### Présélection (Variables explicatives à éliminer)

La suppression de certaines n’engendrent pas de perte d’information.

Soit de manière évidente :

* La colonne 'idannonce' est un identifiant unique pour chaque annonce, elle n'est pas utile pour l'analyse
* La colonne 'annonce\_exclusive' est une variable qui n'est pas utile pour l'analyse
* La colonne 'typedebien' et 'typedebien\_lite' contiennent les mêmes informations; nous gardons la plus riche des deux : 'typedebien
* La colonne 'type\_annonceur' offre une distribution de valeurs trop déséquilibrée
* La colonne 'duree\_int' n'est pas interprétable (valeurs négatives, compréhension empirique)

Soit de par la redondance de l’information :

* Les colonnes 'REG', 'DEP', 'IRIS', 'CODE\_IRIS', 'TYP\_IRIS\_x', 'TYP\_IRIS\_y', 'GRD\_QUART', 'UU2010' sont des colonnes contenant de l'information redondante, de plus nous créerons une nouvelle colonne pour le code postal, générée à partir des coordonnées géographiques

**Note** : Nous avons gardé la colonne 'INSEE\_COM' pour l'utiliser lors de la gestion des outliers

## Prétraitements : partie 1

1. Variables binaires

Suite à la visualisation, il apparait que les variables porte\_digicode, cave et ascenseur sont des variables binaires, nous les transformons en type boolean.

Une image contenant texte, Police, capture d’écran, conception

Le contenu généré par l’IA peut être incorrect.

1. Discrétisations

Les variables

* annee\_construction,
* DpeL,
* ges\_class
* chauffage\_energie

sont discrétisées.

Variable Cible

1. Corrélations

La variable cible identifiée est : “prix\_m2\_vente”.

Certaines autres variables ont l’air d’être liées à celle-ci, nous les identifions avec une matrice de corrélation :Une image contenant texte, capture d’écran, nombre, diagramme

Le contenu généré par l’IA peut être incorrect.

=> Nous éliminons donc les variables “mensualitéFinance” et “prix\_bien” du dataset.

1. Première visualisation de la distribution

Une image contenant texte, capture d’écran, ligne, Rectangle

Le contenu généré par l’IA peut être incorrect.

Il est probable qu’il y ait eu un problème de collecte et la présence d’outliers, que nous avons traité ensuite.

## Prétraitements : partie 2 (Gestion des valeurs aberrantes AVANT le split train et test)

Ci-dessous un extrait de boxplots à ce stade.

Une image contenant texte, ligne, nombre, reçu

Le contenu généré par l’IA peut être incorrect.

Les boxplots montrent des distributions étonnantes.

Aussi certaines variables semblent montrer des problèmes d’unité d’échelle.

Il s’agit en particulier des variables (non exhaustif):

* charges\_copro, loyer\_m2\_median\_n6, loyer\_m2\_median\_n7, taux\_rendement\_n6, taux\_rendement\_n7, nb\_log\_n6,nb\_log\_n7

Suite à ces détections visuelles, nous avons décidé de procéder dans l’ordre à:

* L’élimination de valeurs aberrantes via sélection d’anomalies logiques

Exemple : nb\_toilettes ne peut être supérieur à nb\_pièces

(cf notebook Part-1 pour voir le détail)

* L’élimination de valeurs aberrantes via la définition des seuils\_max et seuils\_min issus de règles métiers

Exemple : taux\_rendement\_n7 ne peut être supérieur à 1

(cf notebook Part-1 pour voir le détail)

* Les séparations train et test

## Exploration des problématiques du projet

a. Afin de répondre aux **deux** problématiques posées, nous avons décidé d’opérer **deux** types de régressions (et donc deux types de séparation train / test):

* **Une régression par série temporelle**, répondant à la problématique de l’évolution du prix au M2 par territoire
* **Régression « classique** », répondant à la problématique d’une estimation fiable d’un logement à partir d’un nombre restreint de variables explicatives.

b. Étant donné **l’hétérogénéité des tendances immobilières** d’un département à l’autre — et parfois même d’une ville à l’autre — nous avons segmenté le territoire en zones homogènes via l’algorithme K-means. Le nombre optimal de clusters (k) a été déterminé grâce à la méthode du coude, garantissant un compromis entre complexité et qualité de regroupement.

Ces clusters alimentent notre analyse de séries temporelles et sont ensuite incorporés, en tant que variable explicative, dans notre modèle de régression “classique”.

c. Nous séparons à ce stade notre dataset en jeu d’entrainement et de test afin d’éviter le data leakage et le sur-apprentissage (constaté en premier lieu lorsque le split se faisait après preprocessing)

Enfin, une fois les splits faits nous procéderons à:

* L’identification des valeurs extrêmes et leur remplacement par la médiane du code INSEE (<=> Code Commune/Code postal).
  + Nous calculons la médiane sur train et appliquons le remplacement sur train et test

## Prétraitements : partie 3 (Gestion des valeurs extrêmes APRÈS le split train et test)

Paramètres gestion des outliers:

lower\_bound = 0.001

upper\_bound = 0.999

Colonne de regroupement : INSEE\_COM

Outlier\_tag = -999

Les outliers sont gérés sur les différents jeux de train puis nous appliquons la méthodologie du jeu de train au test (avec la médiane par code Insee de Train).

### Vérifications visuelles des distributions pour gestion des valeurs aberrantes et des valeurs extrêmes

Une image contenant texte, ligne, diagramme, Tracé

Le contenu généré par l’IA peut être incorrect.

Ci-dessus les boxplots des mêmes variables après les prétraitements : elles semblent correctes.

Une image contenant ligne, texte, Tracé, diagramme

Le contenu généré par l’IA peut être incorrect.

Ci-dessus une vérification de la distribution de la variable cible sur train : elle semble correcte

Une image contenant Tracé, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Ci-dessus une vérification d’une variable explicative : elle semble correcte.

# Partie 2 : Encodages - Feature Selection - Split X/y

**Encodages**

## Sélection de Features

Lors de la première itération, nous avions essayé à ce stade, plusieurs moyens de sélectionner les variables explicatives :

* SelectKBest (mutual\_info\_regression, f\_regression),
* RFE
* Lasso.
* BORUTA

Les temps de calculs pour obtenir une intuition sont parfois longs, par exemple avec la RFA.

Les résultats étaient peu concluants. Ils montraient que le dataset n’était pas suffisamment “propre” pour être interprété correctement (des top features, incohérentes au marché apparaissaient)

Nous avons donc décidé de reprendre le Pré-traitement en y incluant des règles métiers, en améliorant la gestion des valeurs extrêmes et en s’intéressant au sens même de chaque variable, même si nous n’avions pas de lexique des colonnes à disposition.

Après plusieurs ajustements et essais (notamment suite à l’enrichissement du dataset initial), nous avons ensuite directement entraîné les modèles sans sélection de features : les résultats étaient déjà très bons sans optimisation des hyper-paramètres et des paramètres, nous avons donc décidé d’optimiser les modèles dans recourir à la sélection de features.

# Partie 3.1 : Machine Learning – séparation du groupe en 2 équipes

L’équipe projet a été divisé en 2 groupes :

* Groupe 1 pour la régression classique
* Groupe 2 pour les séries temporelles

## Modélisation SARIMAX

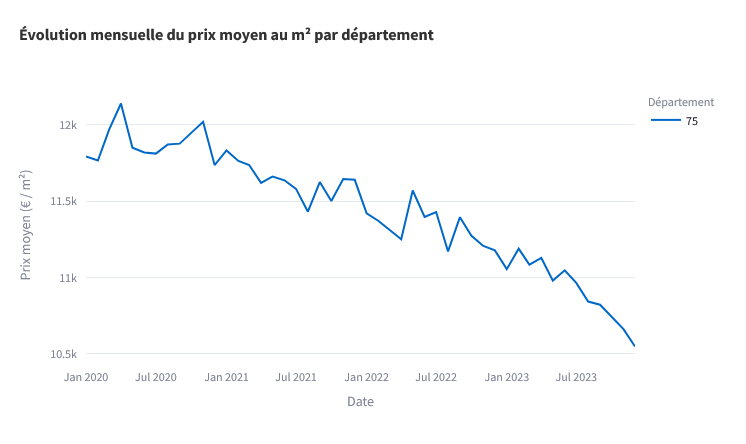
### Contexte

Ce rapport présente la modélisation SARIMAX appliquée aux séries temporelles immobilières, suivant la méthodologie détaillée dans le document “Projets\_méthodologie\_rapports”.

L'objectif principal de la modélisation par série temporelle est de prédire l'évolution du prix au m² sur une période plus longue.

A noter que les premières intentions ont été de faire un **SARIMA** par département.

Mais nous sommes rendus compte qu’il pouvait exister des ordres de prix très différents, et des tendances également différentes, au sein même d’un département donné.



*Figure 1 - Les ordres de prix à Paris*

Une image contenant texte, capture d’écran, Tracé, ligne

Le contenu généré par l’IA peut être incorrect.

*Figure 2 – Tendance en Région Parisienne (hors Paris)*

Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

*Figure 3 - Tendance et prix pour le département du Var*

Une image contenant texte, capture d’écran, Tracé, ligne

Le contenu généré par l’IA peut être incorrect.

*Figure 4 - Les ordres de prix à Saint-Tropez (en échelle logarithmique)*

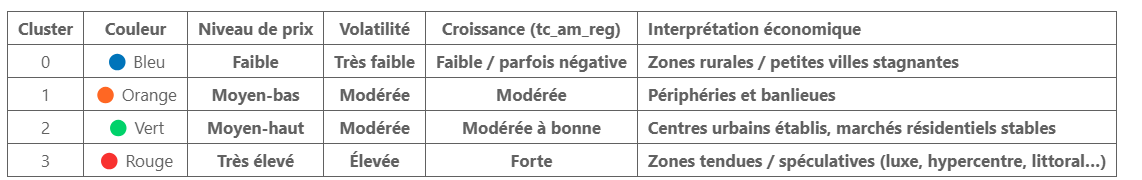
**Premiers constats :**

* Il n’est pas rare de voir des biens à plus de 32000 euros le m² à Saint Tropez.
* Malgré les prix exclusifs à Saint Tropez, le marché immobilier y reste stable.
* A Paris, les prix sont les plus élevés après Saint Tropez.
* Mais on observe un marché en baisse en termes de prix.

Le repérage de comportements atypiques pour chaque département devient un processus extrêmement chronophage.

Comment cerner les zones plus ou moins inflationnistes sans avoir recours à la granularité des villes et qui engendrerait un travail substantiel ?

C’est dans ce contexte que nous avons pensé à la clusterisation avec l’algorithme des K-means



Une image contenant carte, texte

Le contenu généré par l’IA peut être incorrect.

### Classification du Problème

* **Type de problème :** Régression sur séries temporelles.
* **Tâche :** Prédiction des prix immobiliers moyens **mensuels**.
* **Métrique principale :** **RMSE** (Root Mean Squared Error), choisie pour quantifier l'erreur de prédiction sur la même échelle que la variable cible.
* **Autres métriques :** **MAE** (Mean Absolute Error) pour robustesse aux valeurs aberrantes, **MAPE** pour la précision du rapport, Analyse du sur-apprentissage par comparaison du MSE des prédictions sur test vs train

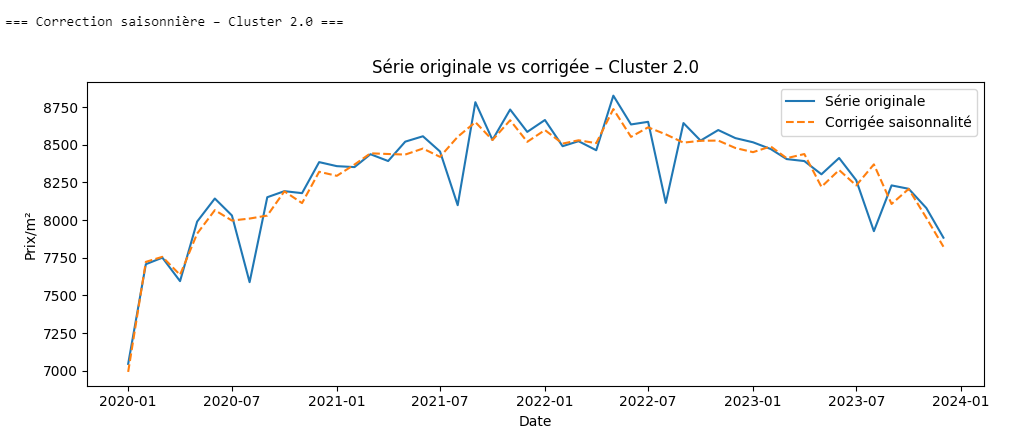
### Choix du Modèle et Optimisation

* **Modèles de Machine Learning testé :** **ARIMA**, **SARIMA**, **SARIMAX**, **Prophet** (pour un cluster, qui présentait de moins bons résultats avec SARIMAX)
* **Modèle de Machine learning retenu :** SARIMAX, avec intégration de « variables exogènes »
* **Les variables (X pour Exogène)** ont été analysées et pré-processées de telle sorte qu’on évite un data leakage et un sur-apprentissage : on a appliqué **un lag sur les variables très corrélées à la cible**
* **Sélection des ordres p,d,q et P,D,Q**: basé sur ACF/PACF, la significativité et les critère AIC et les résultats des test statistiques sur les résidus et l’autocorrélation
* **Optimisation :** Grid Search sur un espace restreint des hyperparamètres avec Validation Temporelle (rolling).

### Interprétation des Résultats et Diagnostics

* *Le modèle capte-il toute la dynamique ?* **Test de Ljung-Box** pour la détection d’une autocorrélation.
* *Les résidus suivent ils une loi normale ?* **Test de Jarque Bera** pour la normalité des résidus
* Affichage du diagnostic SARIMAX
* Visualisation des résidus (histogramme, QQ-plot) pour vérifier la normalité.
* Identification des éventuels patterns non modélisés (saisonnalité, hétéroscédasticité).

### Visualisations



*Figure 5 - Série corrigée des variations saisonnières ( CVS)*

## Décomposition et stationnarisation

Une image contenant texte, ligne, Tracé, diagramme

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, ligne, Tracé, nombre

Le contenu généré par l’IA peut être incorrect.

## Predictions SARIMAX

Une image contenant texte, diagramme, Tracé, capture d’écran

Le contenu généré par l’IA peut être incorrect.

* **Diagnostics du modèle**: ACF/PACF des résidus, histogramme et QQ-plot pour ajuster les premiers ordres (p, d, q)

Une image contenant texte, diagramme, Tracé, ligne

Le contenu généré par l’IA peut être incorrect.

## Evaluation du modèle

Une image contenant texte, capture d’écran, Police, nombre

Le contenu généré par l’IA peut être incorrect.

## Conclusions

**Cluster 0 – Zone rurales, petites villes stagnantes**

· Courbes très proches entre train réel et train prédit.

· Prédictions test légèrement sous-estimées, mais bien orientées.

· Très faible MSE train, bonne généralisation.

· Modèle SARIMAX(1,1,0)(0, 0, 0, 12) avec exogènes significatives (z\_geo, dpel), résidus propres.

=> Marché stable, bien modélisé. Rien à changer.

**Cluster 1 : Banlieue parisienne, zones frontalières et zones mixtes**

· Écart net entre train et test → MSE test x2.

· Sous-estimation persistante, tendance pas bien captée.

· SARIMAX(0,2,2) avec taux\_rendement\_n7, IPS\_primaire,(score relatif à la scolarité) tous significatifs.

· Résidus corrects, mais IC pas toujours aligné.

=> Suggestions d’amélioration:

* Ajouter des exogènes plus sensibles à la dynamique récente et au marché de l’immobilier
* Eventuellement re-cluster selon dynamique locale.

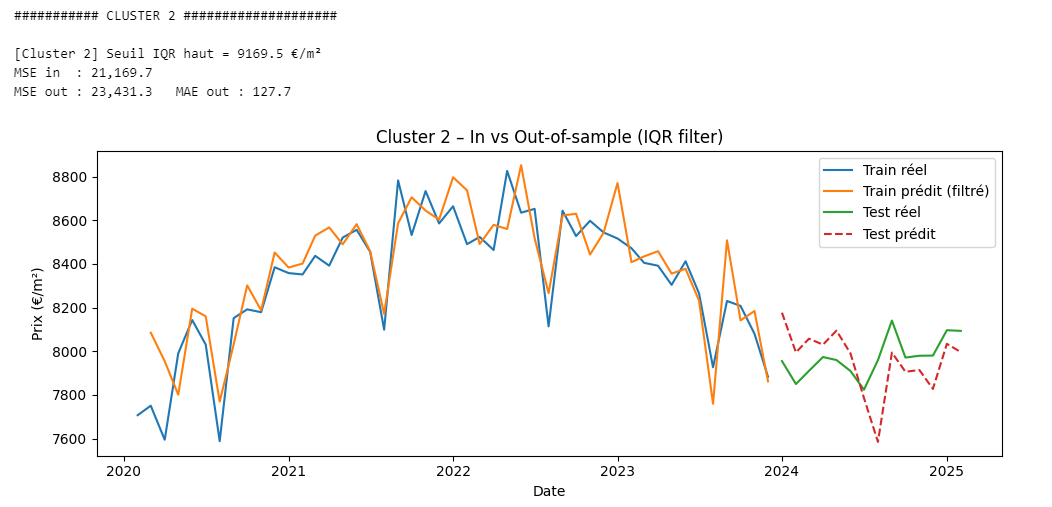
**Cluster 2 : centres urbains**

· Très mauvais scores en train et en test → instabilité du modèle.

· Courbes test en rupture avec l'historique → non prévisible par le modèle.

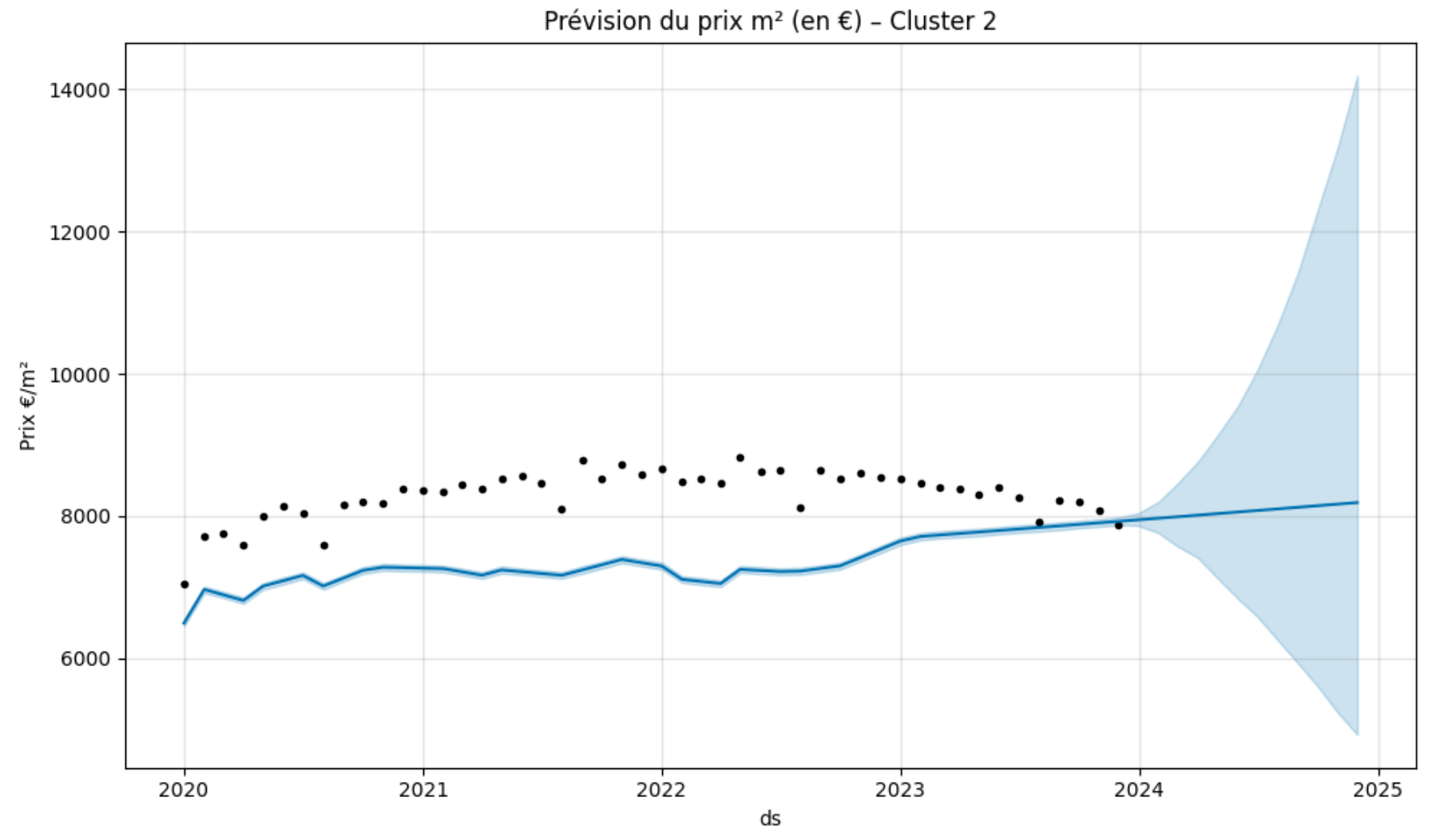
· SARIMAX(1,1,0), avec x\_geo, y\_geo et dpeL très significatifs.

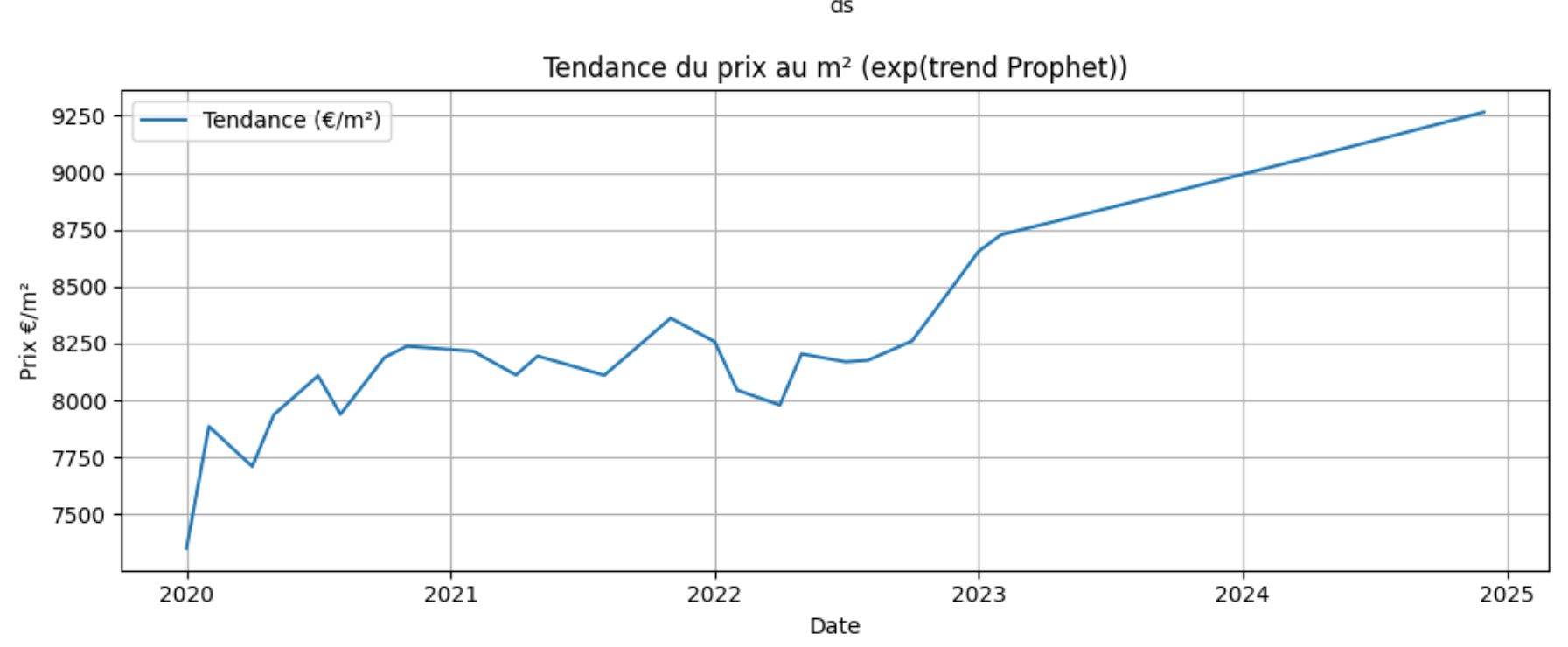
· Résidus statistiquement acceptables, mais mauvaise performance prédictive.

~~~~

=> Suggestions d’amélioration:

* Re-segmenter ce cluster (trop hétérogène ?) selon les particularités immobilières
* Ajouter des indicateurs liés à la rareté, rotation ou marché de niche.
* Tester des modèles plus flexibles : Prophet, XGBoost Time series

Sur ce cluster nous avons testé Prophet mais nous n’avons pas réussi à en tiré d’avantages significatives par rapport à SARIMAX



**Cluster 3 : Paris et villes de « luxe » - Zones spéculatives**

· Meilleure performance test que train → excellente généralisation.

· Suivi fidèle des dynamiques de prix.

· SARIMAX(0, 1, 0) avec y\_geo seul en exogène, résidus très propres.

=> Conclusion : Modèle très robuste

# Partie 3.2 : Estimation du prix au m2

## Préambule et objectif:

* nous ne retraçons pas ici la clusterisation, que nous avons utilisée pour améliorer la précision des estimations, car elle est déjà détaillée dans la partie “Série Temporelle”
* avant d’opérer les régressions, nous avons encodé les variables avec un pipeline rassemblant plusieurs techniques d’encodage, en fonction du type de variable:
  + ordinal encoding: ["ges\_class", "dpeL", "logement\_neuf", "nb\_pieces", "bain", "eau", "nb\_toilettes", "balcon"]
  + onehot encoding: ["typedebien", "typedetransaction", "chauffage\_mode", "chauffage\_energie\_principal"]
  + target encoding: ["etage", "nb\_etages", "exposition", "chauffage\_energie", "chauffage\_systeme"]
  + geo: ["x\_geo", "y\_geo", "z\_geo"]
  + dates: ["date"]

Que nous avons ensuite standardisé, sauf les dates

**Objectif**

La tâche consiste en une évaluation de prix (euros par mètre carré) par régression.

## Métrique de performance

La métrique principale utilisée pour comparer nos modèles est le RMSE (Root Mean Square Error). Cette métrique a été choisie pour plusieurs raisons :

* Elle est exprimée dans la même unité que notre variable cible, facilitant l'interprétation des résultats.
* Elle pénalise davantage les erreurs importantes, ce qui est crucial dans un contexte où les erreurs significatives peuvent avoir des conséquences financières importantes.
* C'est une métrique standard dans le domaine de la régression, facilitant la comparaison avec d'autres modèles ou études.

En complément du RMSE, nous avons utilisé :

* R² (coefficient de détermination) : Mesure la proportion de la variance de la variable cible expliquée par le modèle.
* MAE (Mean Absolute Error) : Mesure l'erreur absolue moyenne, offrant une vision complémentaire au RMSE.
* MAPE (Mean Absolute Percentage Error) : Exprime l'erreur en pourcentage de la valeur réelle, permettant une interprétation relative de la précision du modèle.

## Choix du modèle et optimisation

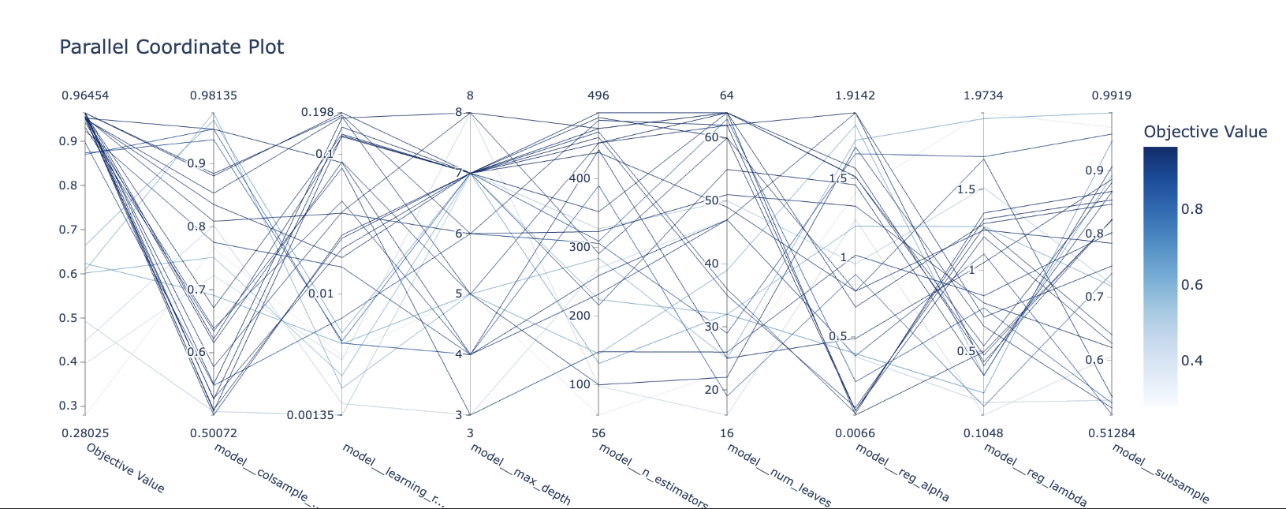
Nous avons testé plusieurs algorithmes :

* Modèles linéaires : Régression linéaire multiple, Ridge, Lasso.
* Arbres de décision : Arbres individuels, forêts aléatoires (Random Forest).
* Boosting : XGBoost, LightGBM.

## Modèles retenus

* LightGBM : Ce modèle s'est particulièrement distingué par sa rapidité d'exécution et sa précision, devenant notre modèle de référence pour la régression classique.
* Forêts aléatoires : ont montré une amélioration significative des performances par rapport aux arbres individuels grâce au principe du bagging.

## Optimisation des Hyperparamètres



* Validation croisée et recherche par grille
* Optimisation bayésienne via Optuna : pour les modèles complexes comme LightGBM.
* Hyper Paramètres optimisés pour LightGBM : nombre d’arbres, profondeur maximale, taux d’apprentissage, etc., via une recherche Optuna sur 30 essais avec validation croisée à 5 folds (scoring R²).
* résultats:

## Gestion du Surapprentissage

Techniques utilisées :

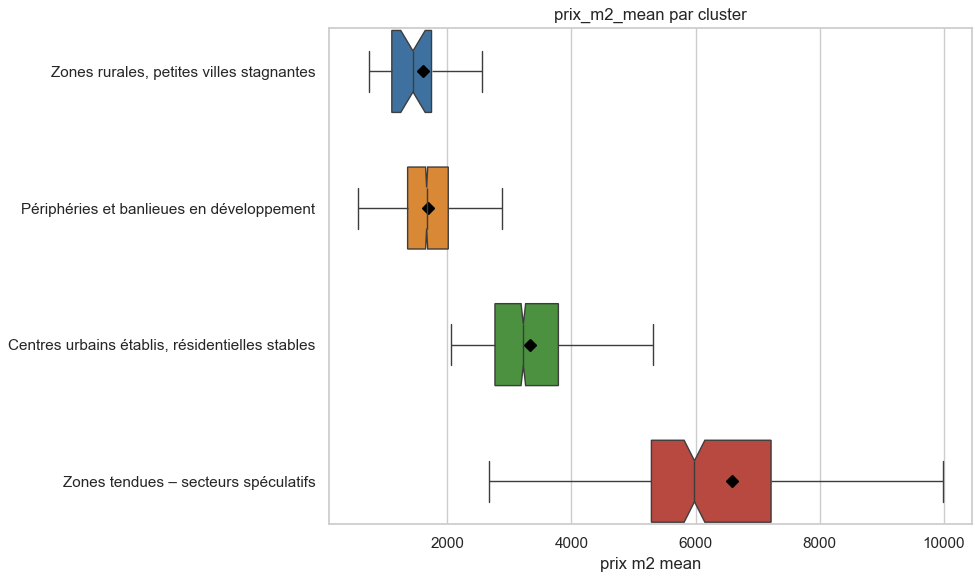
* Régularisation (L1, L2).
* Arrêt précoce (early stopping).
* Validation croisée.
* Sous-échantillonnage des variables et observations.

Résultats: 

### Approche par Clusters Territoriaux

Segmentation territoriale :

* Agrégation des codes postaux peu renseignés au niveau du département.
* Identification de cinq clusters distincts.
* Modèles spécifiques entraînés pour chaque cluster.



## Performances des Modèles

Régression classique :

* R² de 0,96 pour LightGBM.
* RMSE de 425 euros par mètre carré.

Nous estimons que ces métriques montrent une très bonne performance du modèle LightGBM.

# Partie 4 : Modélisation – Approfondissement et Interprétabilité

## 4.1 Interprétation des modèles

L’interprétation des modèles constitue un levier essentiel dans ce projet, afin de transformer les prédictions en connaissances actionnables. Plusieurs approches ont été mobilisées en fonction du type de modèle utilisé :

* Pour les modèles linéaires (régression multiple, Ridge, Lasso) :  
  L’analyse des coefficients permet une lecture directe de l’impact marginal de chaque variable sur la variable cible.
* Pour les modèles non linéaires et plus complexes (Random Forest, LightGBM) :  
  Des méthodes d’interprétation post-hoc ont été employées pour analyser leur fonctionnement interne.
* L’analyse d’importance des variables :  
  Elle offre une vue globale des facteurs influents en quantifiant la contribution moyenne de chaque variable à la performance prédictive.
* Les valeurs SHAP (SHapley Additive Explanations) :  
  Elles permettent d’attribuer, pour chaque prédiction, une part d’explication à chaque variable. Cette approche combine interprétabilité locale et vision globale, tout en tenant compte des effets non linéaires et des interactions.

## 4.2 Valeurs SHAP et interprétabilité globale

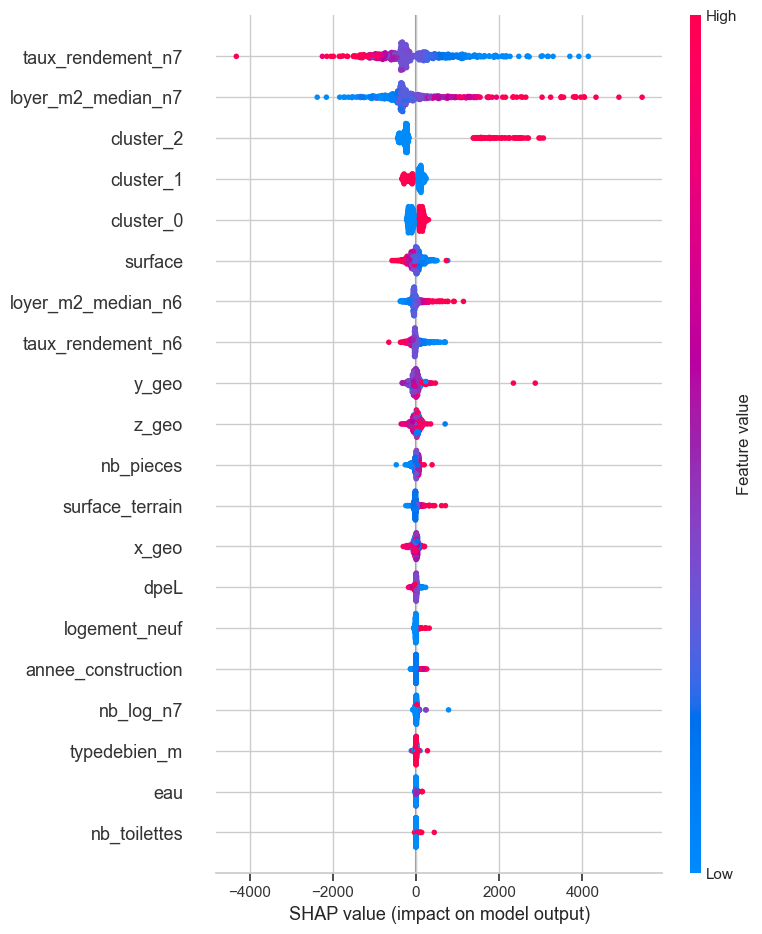
Les valeurs SHAP ont permis de quantifier l’influence de chaque variable sur les prédictions individuelles, tout en offrant une vision globale du modèle.

Le graphique ci-dessous illustre l’impact de chaque caractéristique sur la prédiction du prix au m², en montrant à la fois l’importance de la variable (position verticale), la direction de son effet (position horizontale), et la valeur associée (dégradé de couleurs).

On observe par exemple que :

* Un taux de rendement locatif élevé tend à faire baisser le prix au m².
* Les loyers médians et la localisation via les clusters jouent un rôle déterminant dans la prédiction.
* Les coordonnées géographiques ont un impact plus modéré, mais capturent des effets de micro-localisation.
* Des variables comme la surface, le DPE ou l’année de construction ont un effet plus limité mais non négligeable.

Ce type de visualisation permet d’identifier les leviers les plus influents du modèle, de valider les résultats métier et de repérer d’éventuelles redondances ou interactions entre variables.



## 4.3 Limites de l’approche actuelle

Malgré la richesse du modèle, certaines dimensions restent partiellement captées : la qualité du bien, les finitions, la vue, ou encore des nuisances locales ne sont pas prises en compte faute de données textuelles ou visuelles. L'intégration future d’images ou de descriptions d’annonces via du NLP et de la vision par ordinateur est une piste prometteuse.