Université Paris Nanterre

EURO INFORMATION

Crédit Mutuel

# Mémoire de M2

Master MIAGE (apprentissage)

# Machine learning et estimation immobilière

**Entreprise d'accueil : Euro Information**
**Mémoire réalisé du 4 septembre 2024 au 16 juin 2025**


*présenté et soutenu par*

**Niraiksan Ravimohan**


le 16 juin 2025

## Jury de la soutenance

| | | |
|---|---|---|
| **M. François Delbot,** | Maître de conférences | Responsable du master |
| **M. Valentin Bouquet,** | Maître de conférences | Tuteur enseignant |
| **M. Nourdine Ikhlef,** | Chef de projet | Maître d'apprentissage |

# Remerciements

Dans le cadre de ce mémoire, je tiens à exprimer ma profonde gratitude à plusieurs personnes qui ont joué un rôle important dans mon année et dans la réalisation de ce travail.

Tout d'abord, je remercie M. Valentin Bouquet, mon tuteur enseignant, pour m'avoir accompagné à chaque étape de ce travail. Son aide et nos échanges ont été extrêmement précieux.

Je suis également reconnaissant envers M. Delbot, responsable du master, qui a rendu ce mémoire possible. Grâce à lui, nous avons pu apprendre à faire de la recherche. Cet exercice, bien que difficile, m'a permis d'acquérir de nombreuses compétences précieuses.

Dans le cadre de mon alternance, je tiens à remercier chaleureusement mes maîtres d'apprentissage, Stéphane Camenen et Nourdine Ikhlef. Leur formation a été exemplaire, et leur pédagogie ainsi que leur gentillesse ont rendu cette expérience extrêmement enrichissante.

Enfin, je veux remercier ma famille et mes amis, qui m'ont accompagné et encouragé tout au long de l'année.

# Résumés

## Résumé

Ce mémoire explore la capacité des modèles de machine learning à prédire les prix immobiliers à partir des données publiques issues de la base DVF (Demandes de Valeurs Foncières). Deux espaces géographiques ont été étudiés : la commune de Colombes et le département des Hauts-de-Seine. L'objectif principal est d'évaluer la performance des modèles, tout en analysant l'impact de différentes variables explicatives sur la précision des prédictions.

Après un important travail de prétraitement (filtrage des appartements, suppression des valeurs aberrantes, création de nouvelles variables), plusieurs modèles ont été testés : régression linéaire, Random Forest, XGBoost et CatBoost. Les résultats montrent que les modèles ensemblistes obtiennent d'excellents scores pour la prédiction du prix absolu ($R^2 > 0{,}80$). Toutefois, ces performances reposent en grande partie sur la variable `surface`, qui explique à elle seule une part importante de la variance.

Afin de s'affranchir de cette dépendance, une seconde expérience a été menée en modélisant le prix au mètre carré. Les résultats, plus modestes, révèlent les limites du jeu de données initial. Pour améliorer la qualité prédictive, un enrichissement du dataset a été réalisé en intégrant des variables liées au contexte urbain (données géographiques issues d'OpenStreetMap) ainsi qu'une variable catégorisant les types d'appartement. Cet enrichissement a significativement amélioré les performances, en particulier à Colombes (+10 à +15 points de $R^2$).

L'étude souligne l'importance d'un enrichissement contextuel pour mieux estimer les prix dans les zones où les biens sont difficiles à distinguer sur la base des seules variables DVF. Elle ouvre aussi des perspectives pour de futures recherches, notamment l'intégration d'attributs immobiliers absents (état du bien, DPE, équipements, etc.) afin de mieux expliquer la variabilité des prix au mètre carré.

## Abstract

This study investigates the ability of machine learning models to predict real estate prices using public data from the French DVF (Demandes de Valeurs Foncières) database. Two geographical areas were analyzed : the city of Colombes and the broader Hauts-de-Seine department. The main objective is to assess how well machine learning performs in different spatial contexts (narrow vs. wide area) and to analyze the impact of various explanatory variables on prediction accuracy.

After a thorough data preprocessing phase (filtering for apartments, removing outliers, creating new variables), several models were trained and evaluated : linear regression, Random Forest, XGBoost, and CatBoost. Results show that ensemble models perform very well when predicting absolute prices ($R^2 > 0.80$), although this perfor-

mance is largely driven by the `surface` variable, which alone explains a significant portion of the variance.

To address this dependency, a second experiment was conducted to predict price per square meter instead. The results were more modest, revealing the limitations of the original dataset. To improve performance, the dataset was enriched with urban context variables (geographic proximity data from OpenStreetMap) and a categorical variable representing apartment type. This enrichment significantly improved model performance, especially for Colombes (up to +10 to +15 $R^2$ points).

The study highlights the importance of contextual enrichment to improve price estimation in areas where properties are difficult to distinguish based solely on DVF variables. It also opens new perspectives for future research, notably the integration of additional real estate attributes (property condition, energy ratings, amenities, etc.) to better explain variability in price per square meter.

# Table des matières

# Chapitre 1

# Introduction

L'immobilier est l'un des secteurs les plus importants de notre économie, car il concerne chaque individu, à différents niveaux. Pour l'État, l'immobilier est un levier majeur qui influence directement la conjoncture économique. Pour les investisseurs, il représente un terrain d'opportunités financières et patrimoniales. Et pour la population en général, il s'agit tout simplement du lieu de vie, d'un besoin fondamental.

Parallèlement, l'intelligence artificielle connaît aujourd'hui une croissance rapide et transforme de nombreux domaines, y compris celui de l'immobilier. Elle s'impose comme un outil capable d'automatiser diverses tâches réalisées par les professionnels du secteur, comme la rédaction d'annonces, les visites virtuelles ou encore l'estimation de biens. C'est précisément sur cette dernière tâche que se concentre ce mémoire : l'estimation immobilière.

Selon l'agence ORPI, estimer un bien immobilier consiste à évaluer sa valeur à un instant donné, qu'il s'agisse d'un appartement ou d'une maison. Il existe aujourd'hui plusieurs méthodes pour y parvenir, allant des approches traditionnelles, souvent manuelles et encore largement utilisées, aux méthodes plus récentes reposant sur des techniques avancées d'apprentissage automatique. Chaque pays développe ses propres pratiques en matière d'évaluation immobilière, influencées par sa culture, son marché et son historique.

Les méthodes traditionnelles d'estimation, comme celles fondées sur la comparaison ou le coût, sont souvent sujettes à des biais humains. Les méthodes issues du machine learning (ML), quant à elles, promettent d'apporter plus de rigueur, de rapidité et de précision.

L'objectif de ce mémoire est d'évaluer dans quelle mesure les méthodes de Machine Learning permettent d'estimer avec précision le prix d'un appartement dans les Hauts-de-Seine, en France, et d'identifier les modèles les plus performants. Pour ce faire, nous commencerons par présenter les principes généraux du Machine Learning, puis un état de l'art des approches existantes appliquées à l'estimation immobilière. Nous procéderons ensuite au nettoyage et à la préparation d'un jeu de données issu de la base Demandes de Valeurs Foncières (DVF). Enfin, une expérimentation empirique sera menée afin d'évaluer plusieurs modèles et d'en tirer des conclusions sur leur pertinence dans le contexte de la prédiction de prix immobiliers.

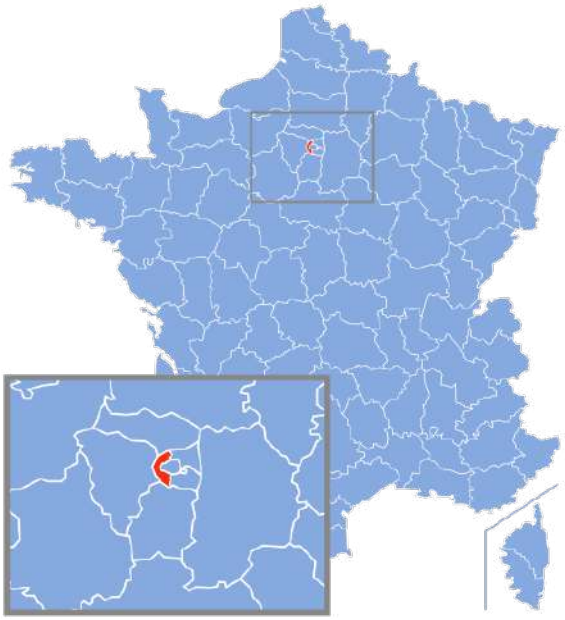


FIGURE 1.1 – Les Hauts-De-Seine [1]

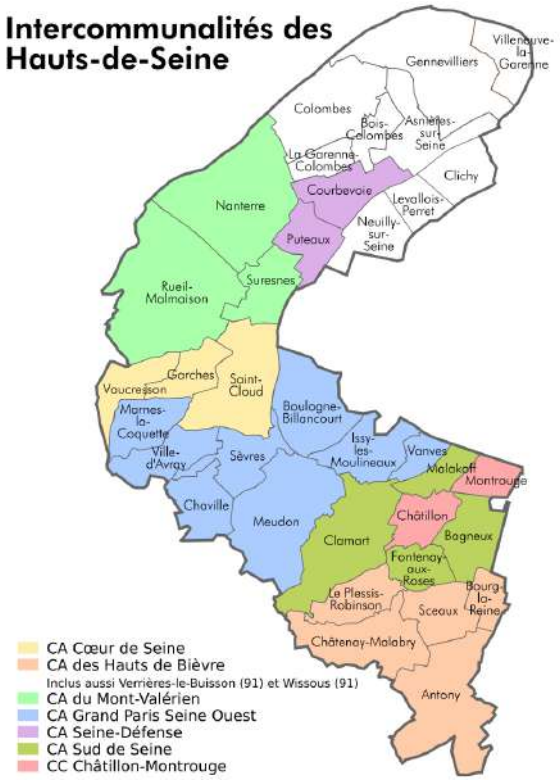



FIGURE 1.2 – Les Hauts-De-Seine [1]

# Chapitre 2

# Méthodologie

Dans ce chapitre, est présentée la méthode utilisée pour la recherche d'articles. Il s'agit de la méthode de la revue systématique de la littérature (Systematic Literature Review, SLR). En suivant cette méthode, nous avons défini les objectifs et les questions de recherche. Nous avons ensuite effectué le processus de recherche en deux étapes.

## 2.1 Objectifs et questions de la recherche

L'objectif principal de cette étude est d'identifier les différentes méthodes d'apprentissage automatique utilisées pour l'estimation de la valeur des biens immobiliers, d'évaluer leur performance respective, et de comprendre dans quelle mesure ces approches sont efficaces pour prédire le prix d'un appartement dans les Hauts-de-Seine à un instant donné.

Dans ce cadre, nous formulons les questions de recherche suivantes :

| |
|---|
| QR1 : Quelles sont les principales méthodes de machine learning appliquées à l'estimation immobilière ? |

| |
|---|
| QR2 : Quelles sont les méthodes les plus performantes selon les critères d'évaluation ? |

| |
|---|
| QR3 : Dans quelles conditions ces méthodes offrent-elles des résultats fiables et généralisables ? |

## 2.2 Processus de recherche

La recherche bibliographique a été menée à partir de plusieurs bases de données scientifiques reconnues, notamment *IEEE Xplore*, *ACM Digital Library*, *MDPI*, *CiteSeerX* et *ScienceDirect*. Elle s'est déroulée en deux étapes principales : d'abord l'identification des articles pertinents, puis leur sélection sur la base de critères définis.

### 2.2.1 Identification d'articles

À cette étape, les articles contenant les mots clés suivants sont recherchés sur Google Scholar : ”machine learning” ”real estate” ”prediction” ”valuation”.

Requête sur Google Scholar : intitle :”machine learning” intitle :”real estate” intitle :”prediction”|”valuation” -”review” -chemical

Résultats : 77 articles.

### 2.2.2 Sélection d'articles

À cette étape, les 77 articles sont triés et sélectionnés dans cet ordre :
— Sélection 1 : Exclusion des citations
— Sélection 2 : Une sélection est effectuée en lisant les titres des articles.
— Sélection 3 : Les articles accessibles et pertinents sont finalement choisis.

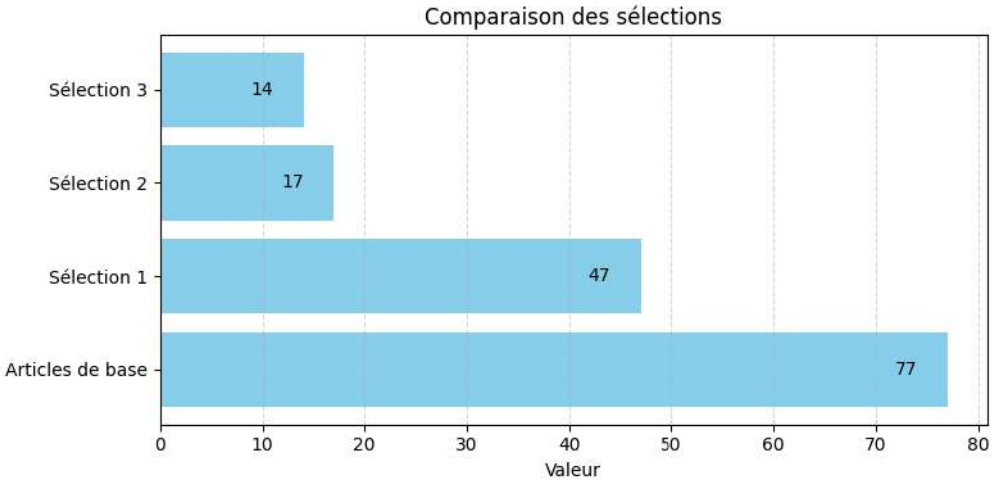



FIGURE 2.1 – Recherche 1 : nombre d'articles par sélection

Ce graphique 2.1 représente l'évolution du nombre d'articles en fonction des sélections.

En complément des sources scientifiques mobilisées, cette étude s'appuie également sur les connaissances acquises dans le cadre du Master 2 MIAGE de l'Université Paris Nanterre, et plus particulièrement sur les enseignements de Machine Learning dispensés par MADAME SANA BEN HAMIDA. Ces enseignements s'appuient sur l'ouvrage de référence *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* de AURÉLIEN GÉRON [15].

# Chapitre 3

# Généralités sur le Machine Learning

Dans ce chapitre, nous présentons les généralités sur le Machine Learning, en posant les bases nécessaires à la compréhension des méthodes utilisées dans le cadre de notre étude. L'ensemble des notions abordées s'appuie exclusivement sur le cours de Machine Learning dispensé par Madame Sana Ben Hamida, dans le cadre du Master 2 MIAGE de l'Université Paris Nanterre[15].

## 3.1 Les grandes catégories du Machine Learning

Selon Arthur Samuel (1959), le Machine Learning est un domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans avoir été explicitement programmés. Ainsi, le Machine Learning se divise en plusieurs grandes catégories, chacune répondant à des besoins spécifiques en matière d'analyse de données.

Tout d'abord, nous avons l'apprentissage supervisé (*Supervised Learning*) qui repose sur l'utilisation d'exemples d'apprentissage comprenant à la fois des données d'entrée et les sorties correspondantes, appelées *labels*. L'objectif est d'apprendre à prévoir ou à classifier en se basant sur ces données historiques et étiquetées. Les algorithmes supervisés sont particulièrement utilisés pour des tâches de classification (comme la reconnaissance de classes à partir de descripteurs) ou de régression (prédiction de valeurs numériques). Les données utilisées dans ce cadre sont généralement structurées sous forme de paires $(x_j, y_j)$, où $x_j$ représente un vecteur d'entrée et $y_j$ la valeur cible associée.

Ensuite, l'apprentissage non supervisé (*Unsupervised Learning*) diffère fondamentalement du précédent puisqu'il n'utilise que des données d'entrée, sans labels associés. Le but ici est d'identifier des structures ou des régularités dans les données, comme le regroupement d'exemples similaires en *clusters* ou la découverte de règles d'association. Ce type d'apprentissage est également utilisé pour des techniques de réduction de dimensionnalité.

Une autre approche est celle de l'apprentissage par renforcement (*Reinforcement Learning*), qui s'appuie sur le principe d'apprentissage par l'expérience. Il s'agit de permettre à un système de déterminer les actions à entreprendre pour maximiser une récompense. En observant les conséquences de ses décisions et en évaluant leur efficacité, le système ajuste ses comportements futurs. Ce paradigme repose donc sur une logique d'essais et erreurs, avec des mécanismes de récompenses et de punitions.

Enfin, l'apprentissage semi-supervisé (*Semi-Supervised Learning*) se positionne entre les approches précédentes. Il exploite un mélange de données étiquetées et non étiquetées, permettant ainsi de tirer parti d'un grand volume de données non annotées tout en s'appuyant sur un petit ensemble de données supervisées pour guider l'apprentissage.

Dans le cadre de ce mémoire, portant sur l'estimation des prix immobiliers — qui constitue un problème de régression, c'est-à-dire la prédiction d'une valeur quantitative continue — nous nous concentrerons principalement sur les approches d'apprentissage supervisé.

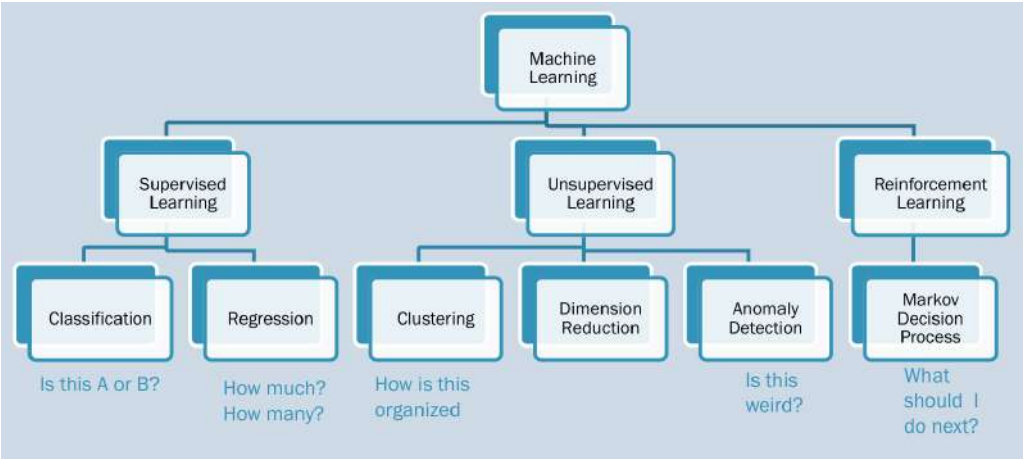



FIGURE 3.1 – Les grandes catégories du Machine Learning [15]

## 3.2 Introduction aux méthodes de régression

La régression est une technique d'*apprentissage supervisé* dont le but est, à partir d'une base de données d'exemples $(x_j, y_j)$, de découvrir une fonction $f$ capable de prédire la valeur cible réelle $Y$ pour un nouveau vecteur d'entrée $X$. Pour chaque vecteur $X_j$, cette fonction $f$ permet de calculer $f(x_j)$, et l'objectif est de minimiser l'écart entre $f(x_j)$ et $Y_j$ sur l'ensemble des exemples d'apprentissage.

Dans ce contexte, les données d'entrée $X = (x_1, x_2, ..., x_n)$ sont appelées *variables explicatives*, tandis que la cible $Y$ est désignée comme la *variable expliquée*. Il est important de noter que, dans le cadre de la régression, ces variables sont généralement quantitatives continues.

Il existe plusieurs méthodes pour effectuer une régression. La régression linéaire, tout d'abord, cherche à *modéliser la relation* entre la variable expliquée $Y$ et une ou plusieurs variables explicatives $X$. Elle le fait en ajustant une droite (dans le cas de la régression linéaire simple, avec la forme $Y = a + bx + e$) ou un *hyperplan* (pour la régression linéaire multiple) de manière à minimiser l'écart entre les valeurs prédites et les valeurs réelles. La méthode des moindres carrés est souvent utilisée pour estimer les coefficients.

Pour mieux capturer des relations non linéaires, on peut aussi se tourner vers la régression polynomiale. Celle-ci constitue un cas particulier de la régression linéaire dans lequel on enrichit les caractéristiques originales par multiplication afin d'introduire des termes non linéaires, comme des *termes quadratiques* dans une régression d'ordre 2.

D'autres techniques de régularisation, comme la régression Ridge et la régression Lasso, viennent compléter ce panorama. Ces deux méthodes étendent la régression linéaire classique en ajoutant une *pénalité* aux coefficients du modèle, afin de limiter le sur-apprentissage. Cette pénalité est régulée par un paramètre appelé $\lambda$, qui permet de *contrôler la force de la régularisation.*

Enfin, il existe encore d'autres approches, souvent plus avancées ou non linéaires, telles que les réseaux de neurones, la modélisation de variétés, ou encore la programmation génétique, qui peuvent être utilisées selon la *nature des données* et des *objectifs.*

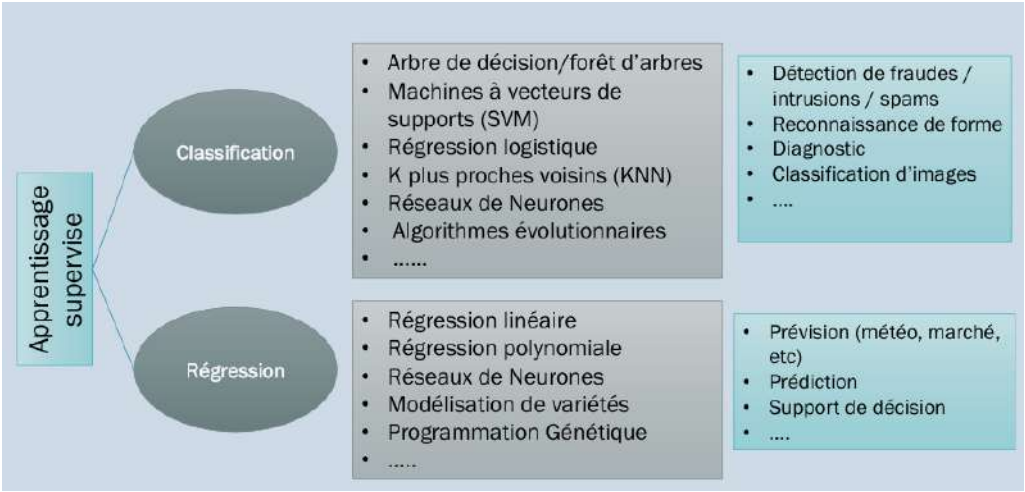



FIGURE 3.2 – Apprentissage supervisé : les techniques [15]

## 3.3 Le processus de développement d'un modèle de Machine Learning

Le développement d'un modèle en apprentissage automatique suit généralement une démarche séquentielle structurée en trois grandes phases : le prétraitement des données, l'apprentissage du modèle, puis le post-traitement ou évaluation. Dans cette section, nous nous concentrerons plus particulièrement sur les étapes de *prétraitement des données* et de *post-traitement.*

### 3.3.1 Prétraitement des données

La qualité des prédictions générées par un modèle d'apprentissage automatique dépend fortement de la qualité des données utilisées. Le principe bien connu ■ *Garbage In, Garbage Out* ■ (GIGO) rappelle qu'un modèle ne peut produire de bons résultats que si les données d'entrée sont fiables et pertinentes.

Le prétraitement des données regroupe ainsi un ensemble d'étapes essentielles visant à assurer leur cohérence, leur intégrité et leur compatibilité avec les algorithmes de machine learning :

- Collecte et intégration des données : cette étape initiale consiste à réunir les données issues de différentes sources (bases de données, fichiers CSV, API, etc.), tout en s'assurant de leur format, de leur validité et de leur compatibilité structurelle.
- Nettoyage des données : il s'agit de corriger ou d'éliminer les données erronées, incomplètes ou bruitées. Les techniques utilisées incluent l'imputation des valeurs manquantes (par moyenne, médiane ou via des méthodes comme `KNNImputer` ou `IterativeImputer`) et la gestion des anomalies ou doublons.
- Encodage des variables catégorielles : les données qualitatives doivent être converties en valeurs numériques via des techniques telles que le *label encoding* ou le *one-hot encoding*, afin de pouvoir être utilisées dans les modèles.
- Détection et traitement des valeurs aberrantes : les outliers peuvent fausser l'apprentissage. Ils sont souvent détectés via des méthodes statistiques ou visuelles (boxplots, z-scores) et peuvent être supprimés ou corrigés.
- Mise à l'échelle des variables : la normalisation (entre 0 et 1) ou la standardisation (centrée-réduite) permet de mettre toutes les variables sur une même échelle, ce qui est particulièrement important pour les modèles sensibles aux distances comme k-NN ou SVM.
- Réduction de la dimensionnalité : lorsque le nombre de variables devient trop élevé, il peut être pertinent de réduire la dimensionnalité via la sélection de caractéristiques (*feature selection*) ou leur extraction (*feature extraction*). L'Analyse en Composantes Principales (ACP ou PCA) est souvent utilisée à cet effet, après normalisation.
- Échantillonnage et partition des données : les données sont divisées en sous-ensembles distincts pour l'entraînement, la validation et le test, afin de pouvoir évaluer les performances du modèle de manière robuste.

### 3.3.2 Évaluation et post-traitement

Le post-traitement a pour but d'évaluer, valider, interpréter et éventuellement déployer le modèle. Plusieurs étapes sont impliquées :

- Validation croisée : au lieu de diviser les données en un unique ensemble d'apprentissage et de test, la méthode de validation croisée, notamment la *k-fold cross-validation*, permet d'obtenir des estimations plus stables et fiables de la performance du modèle.
- Métriques de performance : dans les tâches de régression, les mesures couramment utilisées sont :
  - l'erreur absolue moyenne (MAE : Mean Absolute Error),
  - l'erreur quadratique moyenne (MSE : Mean Squared Error),
  - la racine de l'erreur quadratique moyenne (RMSE : Root Mean Squared Error).

— Optimisation du modèle : sur la base des résultats d'évaluation, des ajustements peuvent être réalisés (tuning des hyperparamètres, choix de nouvelles variables, etc.).
— Déploiement : Une fois validé, le modèle peut être intégré dans une application ou un système décisionnel.

## 3.4 Points de vigilance

Il existe plusieurs point de vigilance lors de la mise en œuvre de projets de Machine Learning. Tout d'abord, il arrive fréquemment que les objectifs soient mal définis, ce qui rend l'orientation du projet *floue* et peut compromettre son succès.

Ensuite, les données disponibles peuvent s'avérer *inadaptées* ou de *mauvaise qualité*, limitant la performance des modèles. Même lorsque les données sont pertinentes, une *préparation insuffisante* ou *inadéquate* peut nuire à leur exploitation.

Par ailleurs, le choix de techniques inappropriées constitue un autre *écueil fréquent* pouvant affecter la qualité des résultats.

Enfin, des problèmes d'apprentissage peuvent survenir, notamment le sur-apprentissage (*overfitting*) et le sous-apprentissage (*underfitting*). Le sur-apprentissage se manifeste lorsque le modèle s'ajuste *excessivement* aux données d'entraînement, compromettant sa capacité à généraliser sur des données nouvelles. À l'inverse, le sous-apprentissage se produit lorsque le modèle ne parvient pas à *capturer les tendances* des données, ce qui le rend peu efficace aussi bien sur les données d'entraînement que sur des données inconnues.

Dans le chapitre suivant, nous aborderons les méthodes de régression spécifiques qui sont couramment utilisées dans la littérature pour l'estimation de la valeur des biens immobiliers.

# Chapitre 4

# Revue de littérature sur l'estimation immobilière par Machine Learning

Ce chapitre présente une synthèse des travaux de recherche récents portant sur l'application de techniques d'apprentissage automatique à l'estimation des prix immobiliers, en mettant en lumière les méthodologies utilisées ainsi que les performances comparées des principaux modèles étudiés.

## 4.1 Méthodologies et modèles d'apprentissage automatique appliqués

Plusieurs études s'intéressent à l'évaluation comparative de différentes techniques de *machine learning* pour l'estimation des prix immobiliers [7][6]. Divers modèles sont couramment utilisés dans la littérature. Parmi les approches classiques ou non paramétriques, les sources mentionnent l'évaluation et l'application des méthodes suivantes, que nous détaillerons dans le prochain chapitre :

- Modèles de régression classiques : La régression linéaire (*Linear Regression*) est souvent utilisée comme modèle de base [3][7][8][6][2]. La régression Ridge est également explorée [3]. La régression Lasso est mentionnée dans une étude [6].
- Arbres de décision (*Decision Trees*) : Ces modèles font partie des approches évaluées [3][1][6]. Dans une étude sur Bogotá, l'arbre de décision pour la régression a obtenu la meilleure performance parmi les modèles classiques avec un $R^2$ de 0,58 [3].
- Méthodes d'ensemble (*Ensemble Methods*) : Ces méthodes sont considérées comme une évolution des approches classiques, combinant plusieurs modèles pour améliorer la performance et la robustesse [3]. Elles sont fréquemment étudiées en raison de leur pertinence. Les modèles d'ensemble évalués comprennent le *Bagging* [3], le *Stacking* [3], la Forêt Aléatoire (*Random Forest*) [1][2][3][4][5][7], l'*AdaBoost* (*Adaptive Boosting*) [7] et le *Gradient Boosting* (incluant des variantes comme *XGBoost* et *CatBoost*) [7][5][2][4].
- Réseaux de Neurones Artificiels (*Artificial Neural Networks* - ANN) : Incluant le *Multilayer Perceptron* (MLP), ces modèles sont également couramment utilisés [1][7][8].
- Autres méthodes : La régression par Vecteurs de Support (*Support Vector Regression* - SVR) [7][8] et la méthode des Plus Proches Voisins (*k-Nearest Neighbors* - k-NN) [1][7] sont aussi appliquées.

La méthodologie *CRISP-DM* est parfois adaptée pour structurer le processus d'étude, comprenant la compréhension des données, la préparation des données, la modélisation, et l'évaluation [3][8]. La collecte de données repose fréquemment sur le *scraping* de portails immobiliers en ligne [3][5][2]. La préparation des données inclut le nettoyage et la sélection des caractéristiques (*features*) [7][8]. L'évaluation des modèles utilise souvent la validation croisée.

## 4.2 Évaluation de la performance et comparaison des modèles

L'évaluation de l'efficacité des modèles de *machine learning* pour l'estimation des prix est réalisée à l'aide de diverses métriques. Le coefficient de détermination ($R^2$) est couramment utilisé pour mesurer la proportion de la variance du prix expliquée par les variables d'entrée [3][6][5][7][2]. Des métriques d'erreur sont également employées, notamment l'Erreur Moyenne Absolue (MAE - *Mean Absolute Error*) [4][5][7][8], l'Erreur Quadratique Moyenne (MSE - *Mean Squared Error*) [4][5][7][8] et l'Erreur Quadratique Moyenne des Racines (RMSE - *Root Mean Square Error*). D'autres métriques comme le MSLE, Q1, la médiane des erreurs absolues (MedAE), COD et PRD sont aussi mentionnées [7].

Les résultats de la littérature montrent que :

- Les méthodes de *machine learning* surpassent généralement les régressions hédoniques traditionnelles en termes de capacité prédictive [7].
- Les méthodes d'ensemble, telles que la Forêt Aléatoire, le *Bagging* et les algorithmes de *Boosting* (*XGBoost*, *CatBoost*, *AdaBoost*, *Gradient Boosting*), montrent souvent les meilleures performances [2][3][5][7].
  - Dans l'étude sur Bogotá, *Bagging* ($R^2 = 0{,}63$) et *Random Forest* ($R^2 = 0{,}65$) ont obtenu les meilleurs résultats [3].
  - Dans l'étude sur la Turquie (Konya), *Random Forest* était le plus performant selon le $R^2$ [1].
  - Dans l'étude sur les villes françaises, les méthodes d'ensemble (*Random Forest*, *Gradient Boosting*, *AdaBoost*) ont surpassé les autres modèles lorsque les caractéristiques de géocodage étaient incluses [7].
  - Dans les études sur l'Arménie, *XGBoost* a obtenu de bons résultats et a surpassé *Random Forest* [2][5].
- Les réseaux de neurones artificiels peuvent également donner de très bons résultats, surpassant parfois d'autres méthodes, notamment lorsque les caractéristiques spatiales ne sont pas prises en compte [7].
- La régression linéaire peut avoir une performance plus faible dans certains contextes ($R^2 = 0{,}36$ dans l'étude sur Bogotá [3]) mais peut être très efficace dans d'autres ($R^2 = 0{,}81$ dans l'étude sur Bangalore [6]). *Lasso* a atteint un $R^2$ de 0,72 dans cette dernière étude [6].
- L'intégration des caractéristiques de localisation, en particulier via le géocodage, améliore de manière significative le pouvoir prédictif des modèles [7]. Des améliorations de plus de 50 % sur certaines métriques d'erreur ont été observées dans l'étude sur les villes françaises en ajoutant le géocodage [7]. Les pertes en termes de pouvoir prédictif pour un modèle qui n'intègre pas ces caractéristiques peuvent être importantes [7].

— Les performances des modèles peuvent varier considérablement d'une ville à l'autre, avec des différences de précision notables entre les villes à coût de vie élevé et celles à coût de vie moyen [7].

En conclusion, la littérature montre que les méthodes de machine learning sont très efficaces pour l'estimation des prix immobiliers, surpassant souvent les approches traditionnelles, notamment grâce à l'intégration de caractéristiques de localisation précises. Nous allons vérifier ces constats à travers nos propres expérimentations, en appliquant plusieurs de ces modèles à un jeu de données issu des transactions immobilières récentes.

# Chapitre 5

# Cadre de l'expérience

L'objectif de notre expérience est d'évaluer la capacité de modèles de Machine Learning à prédire le prix des appartements dans le département des Hauts-de-Seine. Plutôt que de traiter l'ensemble du département dès le départ, nous avons choisi de commencer par une ville spécifique : Colombes.

Ce choix méthodologique s'explique par deux raisons principales. D'une part, travailler directement à l'échelle du département aurait impliqué un volume de données important, rendant le traitement initial plus complexe. En concentrant d'abord notre analyse sur une seule commune, nous pouvons développer et affiner un pipeline de traitement efficace et reproductible à plus grande échelle. D'autre part, Colombes est une ville que nous connaissons bien, ce qui constitue un avantage lors du nettoyage des données. Cette connaissance du terrain nous permet de mieux repérer et gérer les valeurs aberrantes ou incohérentes dans le jeu de données.

Une fois un modèle performant obtenu pour Colombes, nous appliquerons la même démarche à l'ensemble des Hauts-de-Seine, en adaptant si nécessaire les étapes de préparation et de modélisation.

Ainsi dans ce chapitre, nous présentons le cadre technique et méthodologique que nous avons défini pour la réalisation de nos expérimentations.

## 5.1 Environnement Technique : La Librairie Scikit-learn

Dans le cadre de notre expérience, nous utilisons Scikit-learn [3], une bibliothèque open source d'apprentissage automatique pour le langage Python. Reconnue pour sa richesse fonctionnelle, sa simplicité d'utilisation et sa documentation exhaustive, Scikit-learn constitue un choix de référence pour la mise en œuvre de modèles de Machine Learning.

Scikit-learn couvre un large éventail de méthodes, incluant l'apprentissage supervisé, qui s'avère particulièrement pertinent dans le contexte de l'estimation immobilière. Elle propose une vaste collection d'algorithmes implémentés, notamment pour :

- la régression : `LinearRegression`, `Ridge`, `Lasso` etc.
- la classification : `DecisionTreeClassifier`, `RandomForestClassifier`, `LinearSVC`, `LogisticRegression` etc.
- le clustering : `KMeans`, `AgglomerativeClustering` etc.

Les données sont manipulées sous forme de tableaux à deux dimensions, généralement issus des bibliothèques `NumPy` ou `Pandas`, où chaque ligne correspond à un enregistrement et chaque colonne à un attribut. Les valeurs cibles sont quant à elles stockées dans un tableau unidimensionnel. La bibliothèque prend également en charge la lecture de données issues de fichiers texte ou CSV, facilitant ainsi l'intégration des jeux de données.

Scikit-learn met à disposition des méthodes standardisées pour l'entraînement des modèles (`fit`) et la prédiction (`predict`). Elle propose en complément un module `metrics` dédié à l'évaluation des performances, avec des indicateurs adaptés aux différentes tâches. Dans le cadre de la régression, nous utilisons notamment le coefficient de détermination ($R^2$ score). Par ailleurs, la validation croisée est prise en charge via la fonction `cross_val_score` du module `model_selection`, permettant d'obtenir des évaluations robustes des modèles.

Un des atouts majeurs de Scikit-learn réside dans ses outils de prétraitement des données. Le module `preprocessing` fournit des fonctionnalités essentielles telles que :

- le traitement des valeurs manquantes ;
- la transformation des variables qualitatives en variables numériques (ex. `LabelEncoder`, `OneHotEncoder`) ;
- la normalisation ou le redimensionnement des données (ex. `StandardScaler`, `MinMaxScaler`), étape cruciale pour certains algorithmes sensibles à l'échelle des variables.

Enfin, Scikit-learn permet la sauvegarde des modèles entraînés en vue d'une réutilisation ultérieure, notamment grâce à des bibliothèques comme `joblib` ou `pickle`, ce qui facilite le déploiement ou la réplication des résultats.

## 5.2 Choix des modèles de Machine Learning

Pour notre expérience, nous avons sélectionné cinq modèles de régression : la régression linéaire, l'arbre de décision, la forêt aléatoire (Random Forest), XGBoost et CatBoost. La régression linéaire est utilisée comme modèle de référence (baseline) en raison de sa simplicité, de sa rapidité d'exécution et de son interprétabilité. Elle permet de poser un point de comparaison pour évaluer les performances des modèles plus avancés.

Les autres modèles ont été choisis pour leur efficacité démontrée dans la littérature. Ils seraient capables de gérer des relations non linéaires, des interactions complexes entre variables, et sont bien adaptés à la nature hétérogène des données du secteur immobilier.

Nous présentons ci-dessous les principales caractéristiques techniques de chacun de ces modèles.

### 5.2.1 Régression Linéaire

La régression linéaire constitue un modèle statistique de base[11], fréquemment utilisée comme point de comparaison ("baseline") dans les études empiriques[7]. Elle vise à modéliser une relation linéaire entre une variable dépendante (le prix immobilier, dans notre cas) et une ou plusieurs variables indépendantes représentant les caractéristiques du bien[11]. Ce modèle d'apprentissage supervisé cherche à estimer la meilleure approximation linéaire à partir des données d'entraînement[10].

Dans le cas d'une seule variable explicative (régression linéaire simple), le modèle estime les coefficients $a$ (interception) et $b$ (pente) dans l'équation

$$Y = a + bX + \varepsilon,$$

où $Y$ désigne la variable cible, $X$ la variable explicative, et $\varepsilon$ l'erreur résiduelle[11].

En présence de plusieurs variables explicatives (régression linéaire multiple), l'équation devient

$$Y = a + bX_1 + cX_2 + dX_3 + \ldots + \varepsilon,$$

avec $X_1, X_2, X_3, \ldots$ comme variables indépendantes et $b, c, d, \ldots$ leurs coefficients associés[11]. L'objectif est alors de minimiser l'erreur résiduelle $\varepsilon$ entre les prédictions et les valeurs réelles[11].

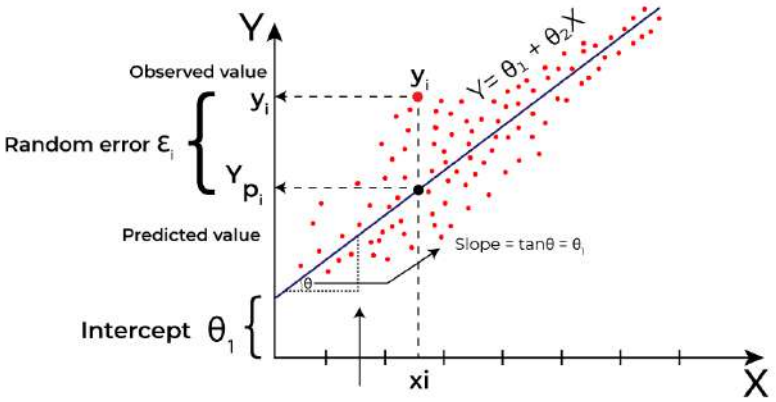



FIGURE 5.1 – Régression Linéaire Simple [4]

### 5.2.2 Arbre de Décision

L'arbre de décision appliqué à la régression est un modèle non paramétrique d'apprentissage supervisé[11], basé sur une partition récursive de l'espace des données[7].

La construction de l'arbre commence par un nœud racine regroupant l'ensemble des données. Le modèle identifie ensuite la variable explicative et le seuil de division qui permettent de séparer les données en deux sous-ensembles les plus homogènes possibles vis-à-vis de la variable cible[7]. Ce processus est répété récursivement sur chaque nœud enfant, formant ainsi une structure arborescente[7]. Le critère de division repose généralement sur la réduction de la variance ou d'une mesure d'impureté[7]. Le processus s'arrête lorsqu'un critère prédéfini est atteint (profondeur maximale, taille minimale des nœuds, ou faible gain d'impureté)[7]. Pour prédire la valeur cible d'une nouvelle observation, celle-ci est acheminée jusqu'à une feuille, dont la prédiction est généralement la moyenne des valeurs cibles associées

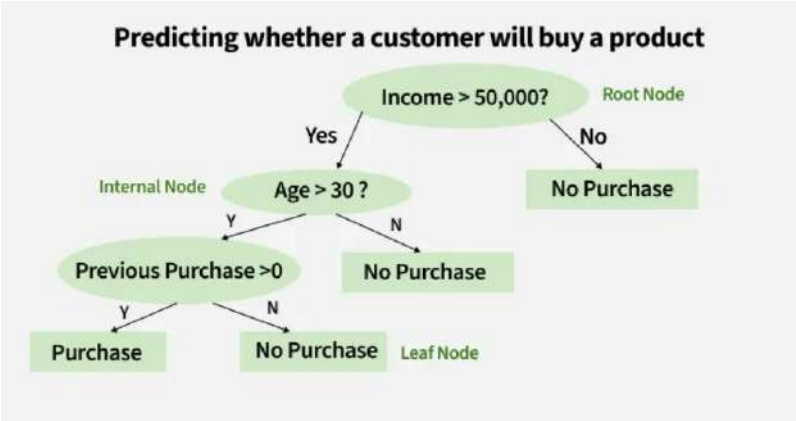



FIGURE 5.2 – Exemple d'Arbre de Décision [5]

### 5.2.3 Random Forest

Le Random Forest (ou forêt aléatoire) est une méthode d'ensemble reposant sur l'agrégation de plusieurs arbres de décision[10, 5]. Il s'appuie sur la technique du "bagging" (bootstrap aggregating)[4, 7, 10] afin de renforcer la robustesse et la stabilité du modèle.

Concrètement, un Random Forest construit un grand nombre ($n\_estimators$) d'arbres indépendants[10], en introduisant deux types d'aléa pour accroître la diversité :

1. Échantillonnage bootstrap : chaque arbre est entraîné sur un échantillon aléatoire (avec remplacement) issu des données d'origine[10].
2. Échantillonnage des variables : à chaque nœud, seule une sous-partie aléatoire des variables est sélectionnée pour déterminer la meilleure division[2, 10].

En régression, la prédiction finale est obtenue par la moyenne des prédictions des arbres individuels[10]. Cette approche réduit la variance et améliore la capacité de généralisation par rapport à un arbre isolé[10]. Les hyperparamètres clés sont notamment le nombre d'arbres ($n\_estimators$) et la profondeur maximale de chaque arbre ($max\_depth$)[7].

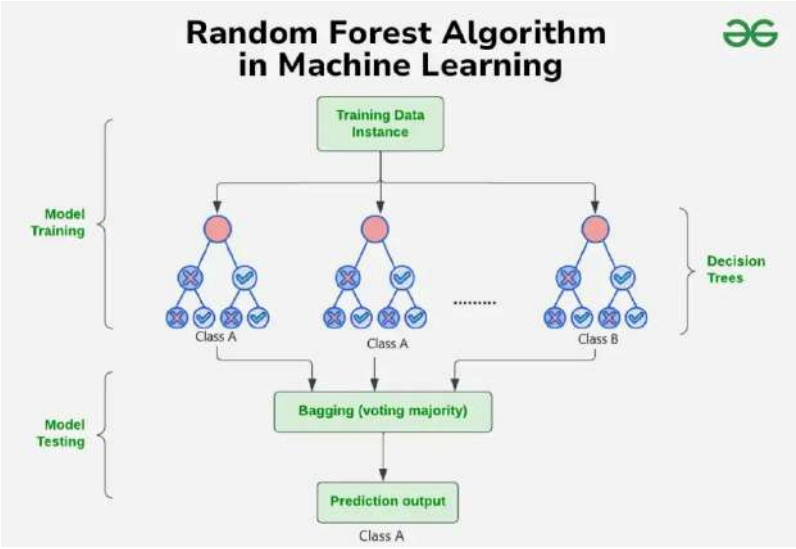



FIGURE 5.3 – Random Forest [6]

### 5.2.4 XGBoost

XGBoost (Extreme Gradient Boosting) est une version optimisée et performante de l'algorithme de Gradient Boosting[2, 10], une technique d'ensemble construite de manière séquentielle pour corriger les erreurs des prédictions précédentes[7].

Le modèle est une somme d'estimateurs faibles, souvent des arbres de décision peu profonds[7]. Le processus commence avec un modèle initial simple (souvent la moyenne des cibles), puis ajoute itérativement de nouveaux arbres ajustés sur les résidus[7]. Chaque nouvel arbre vise à prédire les erreurs restantes, et ses prédictions sont pondérées par un taux d'apprentissage (*learning_rate*)[7].

XGBoost se distingue par plusieurs améliorations techniques[2] :

— Une régularisation (L1/L2) pour limiter le surapprentissage.
— Un traitement efficace des valeurs manquantes.
— Une optimisation matérielle pour le calcul parallèle et distribué.
— Des stratégies d'élagage avancées.

Ces améliorations rendent XGBoost particulièrement rapide et performant[5, 10]. Les hyperparamètres cruciaux incluent le nombre d'estimateurs (*n_estimators*), la profondeur des arbres (*max_depth*), et le taux d'apprentissage (*learning_rate*)[7].

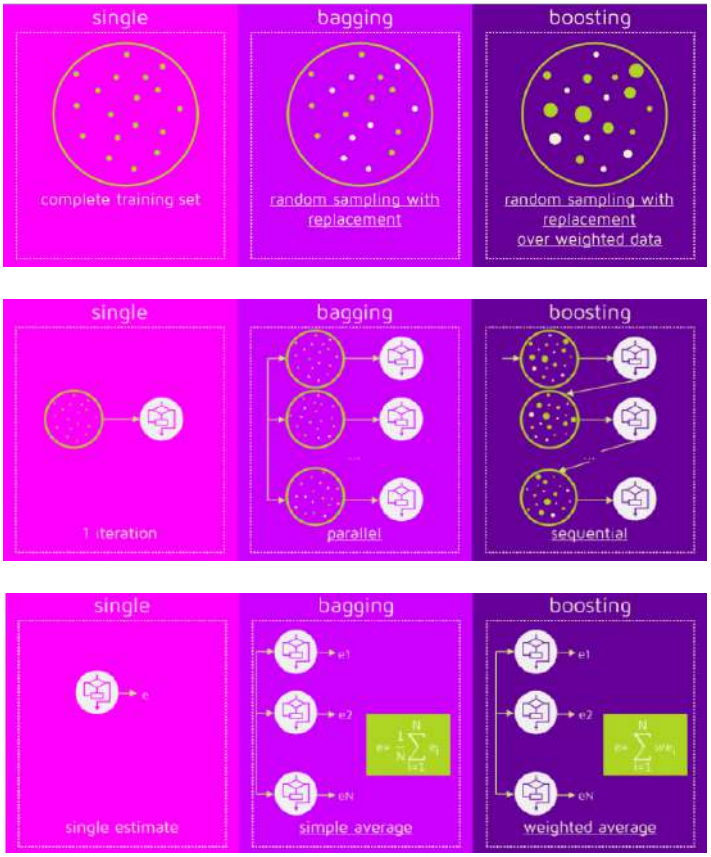



FIGURE 5.4 – Single Methods VS Bagging Methods VS Boosting Methods [7]

### 5.2.5 CatBoost

CatBoost est une autre implémentation du Gradient Boosting, spécialisée dans le traitement efficace des variables catégorielles[10], d'où son nom ("Categorical Boosting").

Comme les autres algorithmes de boosting, CatBoost construit ses prédictions de manière séquentielle en corrigeant les erreurs des modèles précédents[7]. Il se distingue notamment par les innovations suivantes[10] :

- Traitement optimisé des variables catégorielles : utilisation de l'"Ordered Target Encoding" et de l'"Ordered Boosting".
- Gestion des valeurs manquantes : mécanismes robustes intégrés pour leur traitement automatique[14].
- Optimisation interne : calcul efficace, estimation non biaisée des gradients, et parallélisation[10].

CatBoost se caractérise par sa simplicité d'utilisation et ses performances élevées, sans nécessiter de prétraitement manuel des variables catégorielles[10, 14]. Il est particulièrement adapté aux jeux de données contenant un grand nombre de variables qualitatives[10].

## 5.3 Dataset

En ce qui concerne notre dataset, nous avons choisi d'utiliser le jeu de données intitulé ■ Demandes de valeurs foncières géolocalisées (DVF géolocalisées) ■. Ce jeu de données, disponible sur data.gouv.fr, constitue une ressource publique essentielle pour l'analyse des transactions immobilières en France.

### 5.3.1 Présentation du dataset

Le jeu de données DVF géolocalisées est un dérivé enrichi du jeu de données original ■ Demandes de valeurs foncières ■, diffusé par la Direction générale des Finances publiques (DGFiP). Il propose un format alternatif, normalisé et plus aisément exploitable par des outils informatiques, en comparaison aux fichiers bruts initialement publiés.

La mise à disposition de ces données est encadrée par l'article 13 de la loi n° 2018-727 du 10 août 2018 (dite loi ESSOC) et le décret n° 2018-1350 du 28 décembre 2018, qui imposent à la DGFiP la publication électronique des données relatives aux mutations immobilières.

Les informations fournies dans ce jeu de données résultent du traitement informatisé ■ Demande de valeurs foncières ■, lui-même alimenté par la Base nationale des données patrimoniales (BNDP). Cette dernière regroupe les informations issues des documents transmis par les redevables ou leurs représentants aux services en charge de la publicité foncière et de l'enregistrement. La BNDP est notamment alimentée par les traitements informatisés de l'administration fiscale relatifs à la documentation cadastrale (traitement MAJIC) et à la publicité foncière (traitement FIDJI).

Ainsi, chaque fichier disponible est issu du système d'information de la DGFiP, après publication des actes par le service de la publicité foncière et enrichissement par les données cadastrales (référence cadastrale, nature et descriptif des biens). Il convient toutefois de noter que le descriptif complet des biens issus des actes notariés n'est pas intégré, à l'exception de la surface Carrez lorsqu'elle est mentionnée.

Le périmètre géographique couvert par ce jeu de données correspond à la France métropolitaine, à l'exception des départements du Bas-Rhin, du Haut-Rhin et de la Moselle, où le régime du livre foncier prévaut et dont les données ne sont pas intégrées à la BNDP. Les départements et régions d'outre-mer ne sont également pas couverts, à l'exception de Mayotte.

Sur le plan structurel, les fichiers sont organisés par mutation, chacune identifiée par un identifiant unique (`id_mutation`). Une mutation peut concerner plusieurs biens (locaux ou terrains) ; ainsi, chaque ligne du fichier représente un bien concerné par une même mutation. En cas de pluralité de biens ou de natures de culture, le fichier dupliquera les données communes (par exemple, le prix ou la valeur foncière) pour chaque combinaison possible. Par exemple, si $n$ locaux sont construits sur un terrain comprenant $p$ types de cultures, le fichier peut contenir jusqu'à $n \times p$ lignes.

Les données sont fournies à l'échelle du département ou de la commune, avec une mise à jour semestrielle prévue par le décret du 28 décembre 2018. Deux diffusions sont organisées chaque année :

— La diffusion d'avril, couvrant les mutations des cinq dernières années (soit 10 semestres) publiées jusqu'au 31 décembre de l'année précédente.
— La diffusion d'octobre, incluant les mutations des cinq dernières années (11 semestres) publiées jusqu'au 30 juin de l'année en cours.

À la date de la dernière mise à jour (14 avril 2025), le jeu de données couvre une période allant de juillet 2019 à décembre 2024.

Enfin, ce dataset est publié sous Licence Ouverte / Open Licence version 2.0, avec une qualité de métadonnées évaluée à 88,88/100, ce qui en garantit la fiabilité et la réutilisabilité dans un cadre de recherche.

### 5.3.2 Description des colonnes

La liste ci-dessous présente le mapping des colonnes présentes dans le jeu de données "Demandes de valeurs foncières géolocalisées", en décrivant leur signification et leur origine lorsque celle-ci est précisée dans les sources.

Table 5.1: Mapping des colonnes du jeu de données "Demandes de valeurs foncières géolocalisées"

| Nom colonne | Description | Origine |
|---|---|---|
| id_mutation | Identifiant de mutation. Il est indiqué comme non stable et sert à grouper les lignes. Sa création date du 26/04/2019. | Document (acte) |
| date_mutation | Date de la mutation. Il s'agit de la date de signature de l'acte, restituée au format ISO-8601 (AAAA-MM-JJ). | Document (acte) |
| numero_disposition | Numéro de disposition. Dans les actes comprenant plusieurs mutations (appelées "dispositions"), chaque disposition est identifiée par un numéro. Seules les dispositions concernant les mutations à titre onéreux sont restituées. Une disposition est une unité d'analyse juridique. | Document (acte) |

Continued on next page

**Suite du tableau**

| Nom colonne | Description | Origine |
|---|---|---|
| nature_mutation | Nature de la mutation. Les natures possibles sont Vente, Vente en l'état futur d'achèvement (VEFA), Vente de terrain à bâtir, Adjudication, Expropriation ou Échange. Une VEFA est une vente "sur plan" où l'acquéreur devient propriétaire des sols et constructions existantes et à venir. | Document (acte) |
| valeur_fonciere | Valeur foncière. Il s'agit du montant ou de l'évaluation déclaré(e) dans le cadre d'une mutation à titre onéreux. La valeur foncière inclut les frais d'agence si à la charge du vendeur et l'éventuelle TVA. Elle exclut les frais d'agence si à la charge de l'acquéreur, les frais de notaires, et la valeur des biens meubles stipulée dans l'acte. À chaque disposition correspond une valeur foncière déclarée. Le séparateur décimal est le point. | Document (acte) |
| adresse_numero | Numéro dans la voie de l'adresse. | Données cadastrales |
| adresse_suffixe | Suffixe du numéro de l'adresse (B, T, Q). C'est l'indice de répétition. | Données cadastrales |
| adresse_code_voie | Code FANTOIR de la voie (4 caractères). C'est le code Rivoli, un répertoire informatisé codifiant les voies, lieux-dits et ensembles immobiliers par commune. | Données cadastrales |
| adresse_nom_voie | Nom de la voie de l'adresse. Libellé de la voie. | Données cadastrales |
| code_postal | Code postal (5 caractères). Normalisé à 5 caractères. | Données cadastrales |
| code_commune | Code commune INSEE (5 caractères). Normalisé à 5 caractères. Le code INSEE de la commune est généralement sur 3 chiffres, mais le jeu de données le normalise à 5. | Données cadastrales (partie de la référence cadastrale) |
| nom_commune | Nom de la commune (accentué). Libellé de la commune. Utilise des libellés riches et accentués. | Document (acte) |
| ancien_code_commune | Ancien code commune INSEE (si différent lors de la mutation). | Document (acte) |
| ancien_nom_commune | Ancien nom de la commune (si différent lors de la mutation). | Document (acte) |

Continued on next page

**Suite du tableau**

| Nom colonne | Description | Origine |
|---|---|---|
| code_departement | Code département INSEE (2 ou 3 caractères). | Données cadastrales (partie de la référence cadastrale) |
| id_parcelle | Identifiant de parcelle (14 caractères). Compatible avec les fichiers cadastraux proposés par Etalab. Correspond à la référence cadastrale nationale dans la notice descriptive, identifiant l'immeuble par département, commune, préfixe de section, section, numéro de plan, éventuellement volume et lot. Une parcelle est une portion de terrain d'un seul tenant constituant une unité foncière indépendante. | Données cadastrales |
| ancien_id_parcelle | Ancien identifiant de parcelle (si différent lors de la mutation). | Document (acte) |
| numero_volume | Numéro de volume. Un volume est une division de l'espace au-dessus ou en dessous d'un terrain. | Document (acte). Fait partie de la référence cadastrale. |
| lot_1_numero à lot_5_numero | Numéros des 5 premiers lots. Un lot de copropriété est constitué d'une partie privative et d'une quote-part de parties communes. Seuls les 5 premiers lots sont mentionnés, les autres ne sont pas restitués si le nombre total est supérieur à 5. | Document (acte) |
| lot_1_surface_carrez à lot_5_surface_carrez | Surface Carrez des 5 premiers lots. La surface Carrez est la superficie des planchers des locaux clos et couverts après déduction de certaines surfaces, hors parties de locaux < 1,80m et lots < 8m$^2$. Restituée lorsqu'indiquée dans l'acte notarié. | Document (acte) |
| nombre_lots | Nombre total de lots par disposition. | Document (acte) |
| code_type_local | Code de type de local : 1 (maison), 2 (appartement), 3 (dépendance isolée), 4 (local industriel et commercial ou assimilés). La notion de local est une notion fiscale regroupant potentiellement plusieurs lots. | Données cadastrales |
| type_local | Libellé du type de local. | Données cadastrales |

Continued on next page

**Suite du tableau**

| Nom colonne | Description | Origine |
|---|---|---|
| surface_reelle_bati | Surface réelle du bâti, mesurée au sol entre les murs ou séparations, arrondie au mètre carré inférieur, sans prendre en compte les surfaces des dépendances. | Données cadastrales |
| nombre_pieces_principales | Nombre de pièces principales. Cuisines, salles d'eau et dépendances ne sont pas prises en compte. Une "pièce" est un espace cloisonné destiné à séjourner, dormir ou prendre des repas. | Document (acte) |
| code_nature_culture | Code de nature de culture, s'applique aux biens non bâtis. Correspond à une table de références. | Données cadastrales |
| nature_culture | Libellé de nature de culture. Correspond à la table de références. | Données cadastrales |
| code_nature_culture_speciale | Code de nature de culture spéciale. Correspond à une table de références. | Données cadastrales |
| nature_culture_speciale | Libellé de nature de culture spéciale. Correspond à la table de références. | Données cadastrales |
| surface_terrain | Surface du terrain, contenance cadastrale du terrain. | Données cadastrales |
| longitude | Longitude (WGS-84) du centre de la parcelle concernée. Issue du géocodage réalisé par jointure avec le fichier des parcelles cadastrales. | Document (acte) |
| latitude | Latitude (WGS-84) du centre de la parcelle concernée. Issue du géocodage réalisé par jointure avec le fichier des parcelles cadastrales. | Document (acte) |

Il est précisé que les colonnes "Code service" et "Identifiant local" ne sont pas restituées en application du décret n° 2018-1350 du 28 décembre 2018.

Ce dataset fournit donc une richesse d'informations sur les transactions immobilières (date, prix, nature), la localisation géographique précise (coordonnées WGS-84 au niveau de la parcelle), et les caractéristiques des biens (type de local, surface bâtie/-terrain, nombre de pièces principales, nature de culture), le rendant particulièrement pertinent pour des travaux d'estimation immobilière par Machine Learning.

### 5.3.3 Observations des données

Notre dataset comprend 354 751 lignes, représentant autant de mutations immobilières enregistrées dans les Hauts-de-Seine au cours des cinq dernières années. Nous présentons ci-dessous quelques observations afin de mieux comprendre la composition et la structure de ces données.

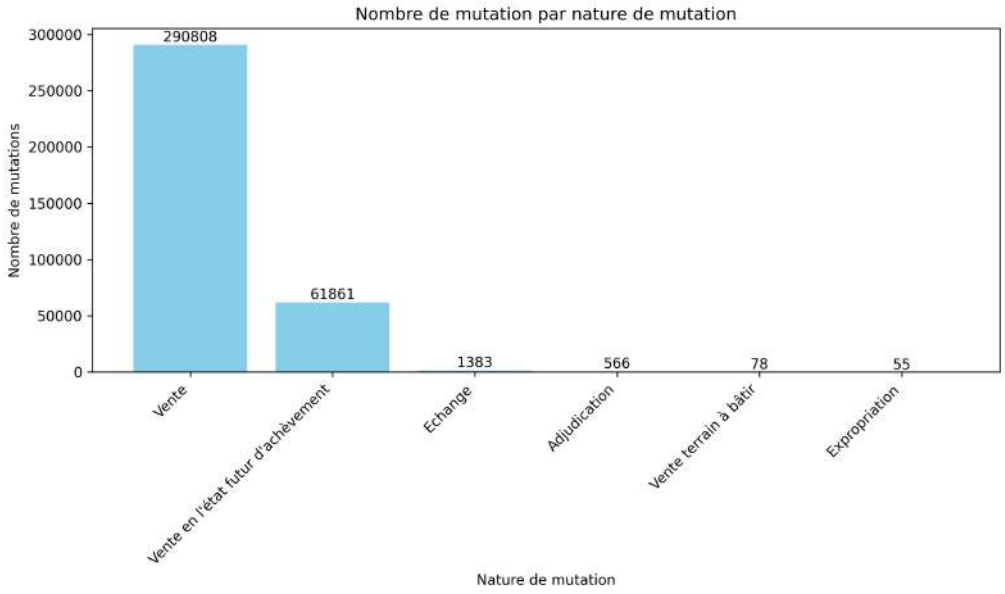



FIGURE 5.5 – Nombre de mutation par nature de mutation

Dans le graphique 5.5, on observe que les mutations les plus fréquentes sont les ventes et les ventes en l'état futur d'achèvement (correspondant à une vente d'un bien immobilier avant sa construction ou pendant sa construction, l'acquéreur devenant propriétaire au fur et à mesure de l'avancement des travaux). Cela est particulièrement pertinent pour notre analyse, puisque notre objectif est de prédire la valeur des appartements vendus, et ces types de mutations correspondent précisément à ce que nous cherchons à modéliser.

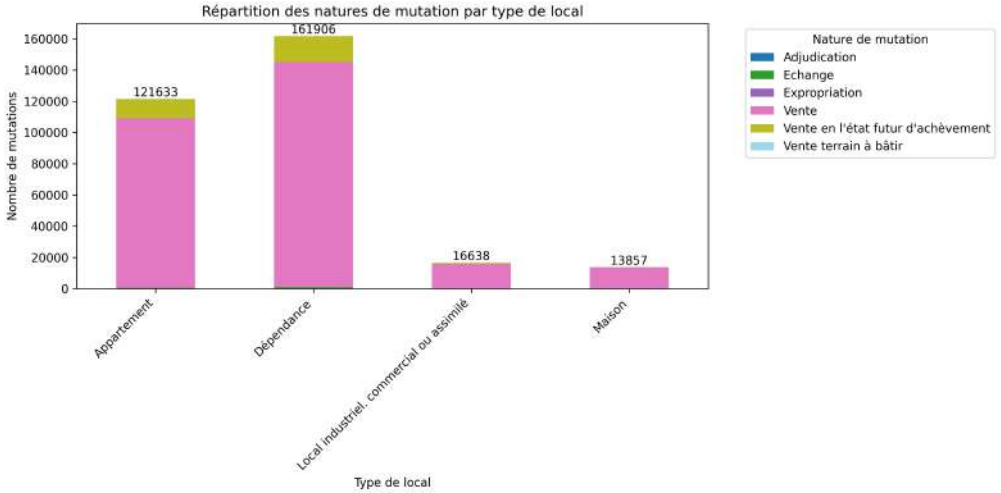



FIGURE 5.6 – Répartition des natures de mutation par type de local

Dans le graphique 5.6, on constate que notre dataset contient un grand nombre de mutations de dépendances, suivies par celles concernant des appartements, tandis que les maisons et locaux sont peu représentés. Avec 121 633 mutations d'appartements, nous disposons d'un volume de données suffisant pour entraîner des modèles robustes.

Par ailleurs, ces mutations d'appartements concernent majoritairement des ventes et des VEFA, ce qui correspond exactement au type de transactions que nous souhaitons analyser. Les autres types de mutations sont présents mais en très faible proportion, ce qui ne nuit pas à la cohérence de notre jeu de données.

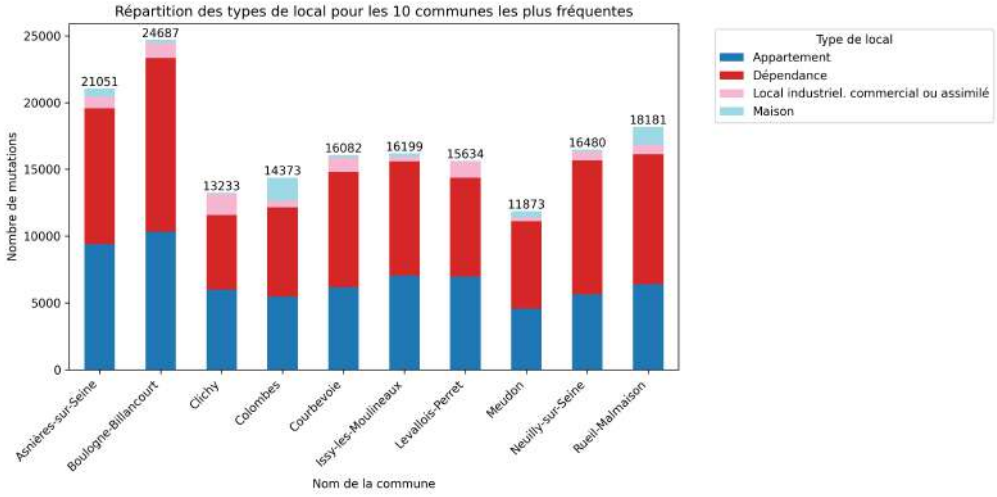



FIGURE 5.7 – Répartition des types de local pour les 10 communes les plus fréquentes

Le graphique 5.7 illustre la répartition des types de locaux pour les dix communes présentant le plus de mutations. On y observe que Colombes fait partie de ce top 10, et que les appartements représentent près de la moitié des mutations enregistrées dans la commune.

Cela confirme la pertinence de notre choix de Colombes comme point de départ pour l'expérimentation, avec un volume de données suffisant et bien ciblé pour l'analyse des ventes d'appartements.

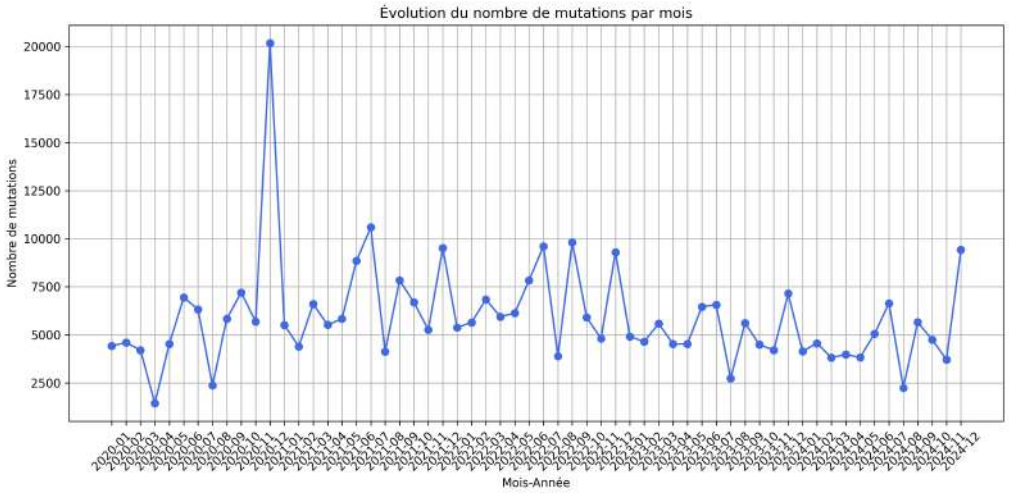



FIGURE 5.8 – Évolution du nombre de mutations par mois

Le graphique 5.8 montre l'évolution du nombre de mutations par mois au cours des cinq dernières années dans les Hauts-de-Seine. On observe une répartition relativement régulière des transactions dans le temps, ce qui est favorable pour notre étude.

Une exception notable concerne la période du confinement lié à la COVID-19, marquée par une forte baisse du nombre de ventes, suivie d'une hausse significative post-confinement, traduisant un rattrapage des transactions. Cette dynamique reste néanmoins cohérente et n'impacte pas négativement la qualité globale du dataset pour notre expérience.

En résumé, notre dataset se révèle riche en informations concernant les ventes d'appartements dans les Hauts-de-Seine. La qualité, la quantité et la pertinence des données disponibles en font une base solide pour mener à bien notre expérience de modélisation prédictive.

# Chapitre 6

# Estimation immobilière

Après avoir défini le cadre méthodologique de notre étude, ce chapitre décrit en détail la mise en œuvre concrète de notre démarche. Nous y présentons les différentes étapes du traitement des données, ainsi que les expérimentations menées pour estimer le prix des appartements, d'abord dans la commune de Colombes, puis à l'échelle plus large du département des Hauts-de-Seine (lien github **??**github]).

## 6.1 Préparation des données

Pour rappel, l'objectif de notre expérience est d'évaluer la capacité de modèles de *Machine Learning* à prédire le prix des appartements dans le département des Hauts-de-Seine (92). Dans une optique de montée en généralité progressive, nous avons choisi de commencer par l'étude d'une commune spécifique du département, à savoir Colombes, avant d'étendre notre analyse à l'ensemble du territoire des Hauts-de-Seine.

Dans la pratique, nous avons commencé par prétraiter le jeu de données correspondant à la commune de Colombes, sur lequel nous avons entraîné nos premiers modèles de *Machine Learning* afin d'observer les performances et affiner notre méthodologie. Ce n'est qu'ensuite que nous avons étendu notre travail à l'ensemble du département des Hauts-de-Seine (92), en appliquant les mêmes étapes de préparation aux données.

Cependant, dans cette section, nous présentons de manière synthétique et unifiée l'ensemble des étapes de prétraitement que nous avons appliquées, sans suivre strictement l'ordre chronologique. Les opérations décrites ci-après ont été réalisées de façon identique pour les deux jeux de données, par script Python Panda.

### Constitution des jeux de données

Nous avons ainsi construit deux jeux de données distincts :

— Le premier contient exclusivement les appartements situés à Colombes, avec 5473 données.

— Le second regroupe l'ensemble des appartements du département des Hauts-de-Seine, avec 121633 données.

## Sélection des données

Nous avons tout d'abord filtré les mutations pour ne conserver que celles correspondant à des ventes effectives :

— `Vente`

— `Vente en l'état futur d'achèvement (VEFA)`

Les autres types de mutations (donations, échanges, etc.) ne sont pas pertinents dans le cadre de notre étude sur la valorisation immobilière.

Par ailleurs, nous avons exclu les lignes pour lesquelles le champ `nature_culture` est renseigné, car celui-ci est utilisé pour décrire des terrains non bâtis, ce qui sort du périmètre de notre analyse, centrée uniquement sur les logements.

Nous avons également supprimé les lignes pour lesquelles les valeurs essentielles (`valeur_fonciere`, `latitude`, `longitude`) sont manquantes.

Enfin, nous avons supprimé les cas de mutations multiples, où un même identifiant (`id_mutation`) peut correspondre à plusieurs biens. Conformément à la documentation officielle, ces cas résultent de la duplication de données communes pour chaque bien concerné par une même mutation.

## Enrichissement des données

Le champ `surface_reelle_bati` s'est révélé souvent imprécis, conformément à ce que soulignent la documentation officielle [8] : cette valeur est cadastrale et peut ne pas refléter fidèlement la réalité du bien au moment de la vente. Pour pallier cette incertitude, nous avons utilisé les informations de surface des **lots**, lorsqu'elles étaient disponibles, pour estimer plus précisément la surface habitable :

— `nb_lots_renseignes` : nombre de lots ayant une surface renseignée.

— `lots_surface_total` : somme des surfaces des lots renseignés.

— `lots_surface_logement` : plus grande surface parmi les lots, supposée être celle du logement principal.

— `lots_surface_autre_lots` : différence entre la surface totale des lots et celle du logement principal.

Dans les cas où aucun lot n'est renseigné, nous remplaçons les surfaces `lots_surface_total` et `lots_surface_logement` par la `surface_reelle_bati`, et fixons `lots_surface_autre_lots` à zéro.

Nous avons ensuite enrichi les données avec les colonnes suivantes :

— `prix_m2` : ratio entre la `valeur_fonciere` et la `lots_surface_logement`.

— `code_secteur_ville` : dérivé de `id_parcelle` (en supprimant les quatre derniers caractères), afin d'identifier des secteurs géographiques au sein de la commune.

— `construction_recente` : variable binaire valant 1 lorsque la mutation correspond à une vente en l'état futur d'achèvement (VEFA), 0 sinon.

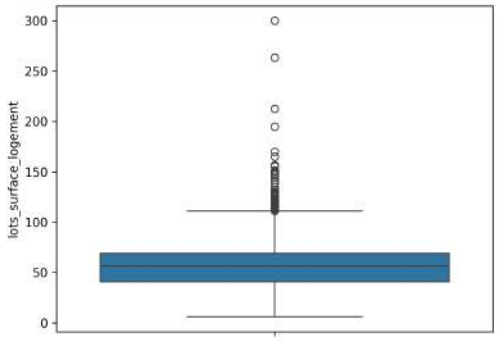
## Traitement des valeurs aberrantes

Afin d'éliminer les valeurs aberrantes susceptibles de fausser l'apprentissage des modèles, nous avons procédé à cinq itérations successives de filtrage sur les variables suivantes :

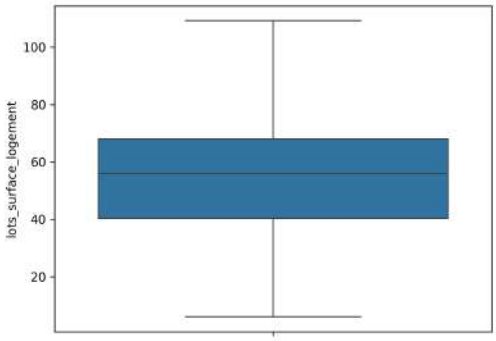
— `lots_surface_logement`
— `prix_m2`
— `valeur_fonciere`

À chaque itération, nous avons supprimé les valeurs situées en dehors de l'intervalle interquartile (inférieures au premier quartile ou supérieures au quatrième quartile).

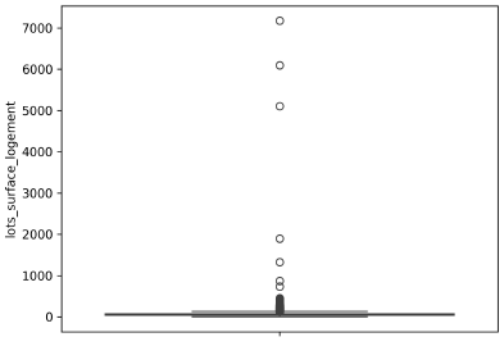
**Colombes - Avant traitement valeurs aberrantes**




**Colombes - Après traitement valeurs aberrantes**




**92 - Avant traitement valeurs aberrantes**




**92 - Après traitement valeurs aberrantes**

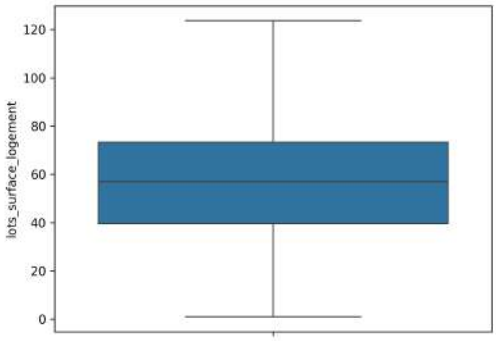
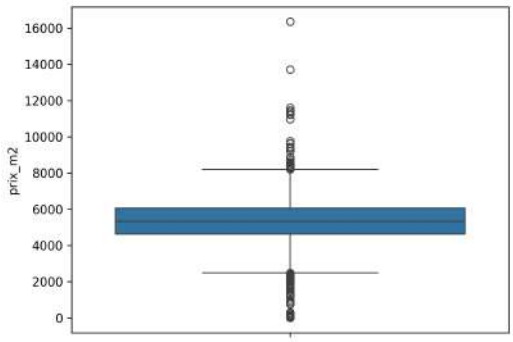
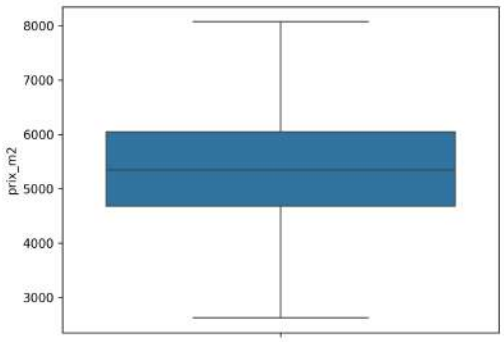



FIGURE 6.1 – Lots surface logement

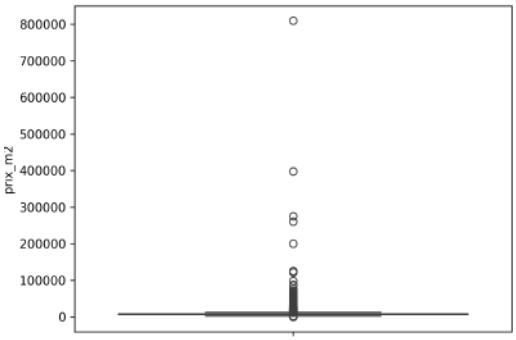
**Colombes - Avant traitement valeurs aberrantes**




**Colombes - Après traitement valeurs aberrantes**




**92 - Avant traitement valeurs aberrantes**




**92 - Après traitement valeurs aberrantes**

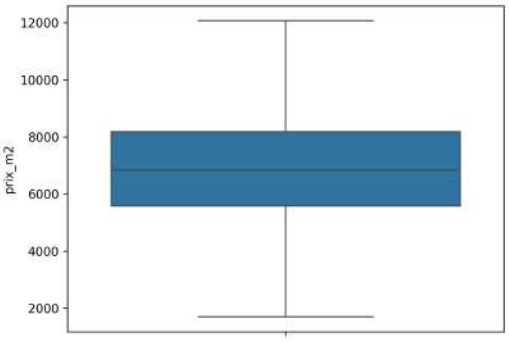
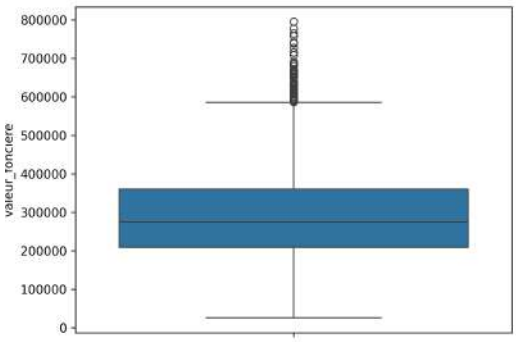
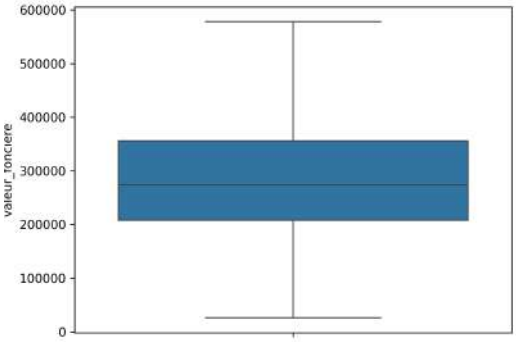



FIGURE 6.2 – Prix m2

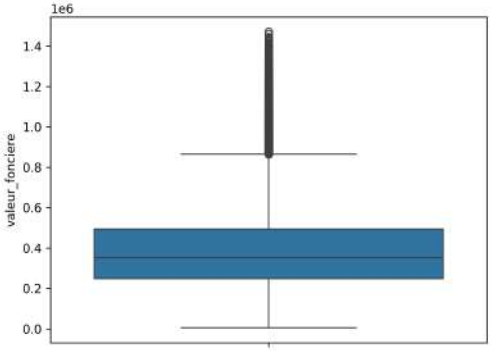
**Colombes - Avant traitement valeurs aberrantes**




**Colombes - Après traitement valeurs aberrantes**




**92 - Avant traitement valeurs aberrantes**




**92 - Après traitement valeurs aberrantes**

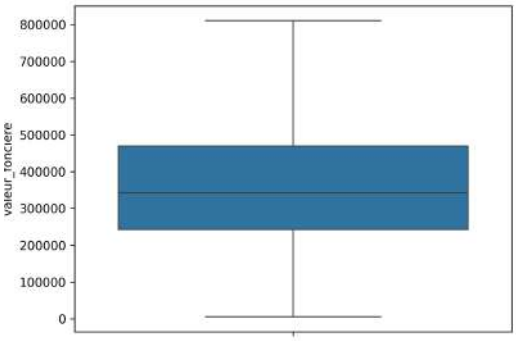



FIGURE 6.3 – Valeur foncière

### Encodage des variables catégorielles

Enfin, nous avons appliqué un `LabelEncoder` sur les variables catégorielles non numériques afin de rendre les données compatibles avec les algorithmes de Machine Learning :

- Pour le jeu de données de Colombes : `id_parcelle`, `code_secteur_ville`, `adresse_code_voie`.
- Pour le jeu de données des Hauts-de-Seine : `id_parcelle`, `code_secteur_ville`, `adresse_code_voie`, `nom_commune`.

Ces étapes nous permettent désormais d'entraîner nos modèles de Machine Learning sur des données préparées de manière rigoureuse, en vue d'estimer au mieux les prix des appartements dans le département des Hauts-de-Seine.

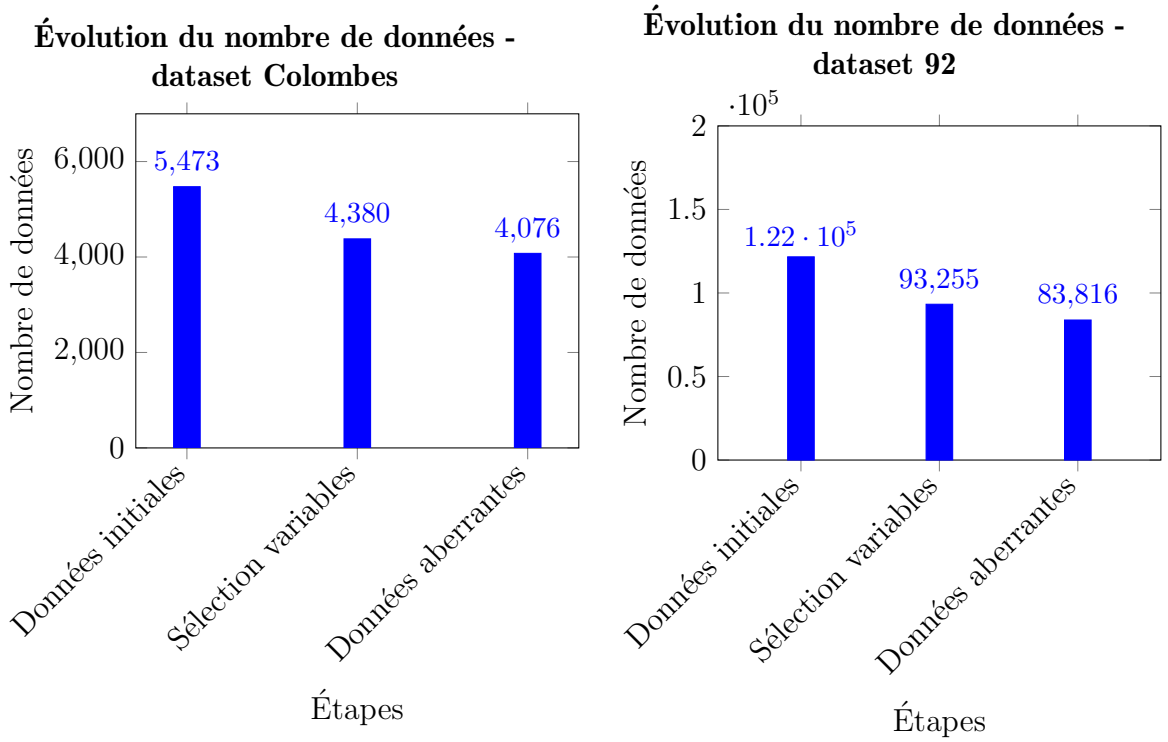



FIGURE 6.4 – Évolution du nombre de données

On peut noter 6.4 que le prétraiement a éliminé 25% des données du dataset de Colombes et 30% des données du dataset des Hauts-de-Seine.

Voici un extrait des jeux de données finaux, obtenus après l'ensemble des étapes de traitement, de nettoyage et d'enrichissement.

| valeur_fonciere | longitude | latitude | code_secteur_ville | id_parcelle | adresse_code_voie | lots_surface_logement | nombre_pieces_principales | construction_recente |
|---|---|---|---|---|---|---|---|---|
| 218000.0 | 2.238498 | 48.925608 | 67 | 681 | 67 | 45.16 | 1.0 | 0 |
| 335402.0 | 2.252587 | 48.932724 | 1 | 5 | 1 | 68.0 | 3.0 | 1 |
| 237000.0 | 2.243444 | 48.915559 | 43 | 480 | 43 | 47.33 | 2.0 | 0 |
| 224340.0 | 2.246255 | 48.926389 | 65 | 676 | 65 | 58.91 | 5.0 | 0 |

FIGURE 6.5 – Extrait dataset encodé - Colombes

| valeur_fonciere | nom_commune | longitude | latitude | code_secteur_ville | id_parcelle | adresse_code_voie | lots_surface_logement | nombre_pieces_principales | construction_recente |
|---|---|---|---|---|---|---|---|---|---|
| 385000.0 | 27 | 2.176749 | 48.877211 | 927 | 13087 | 927 | 72.31 | 2.0 | 0 |
| 351000.0 | 27 | 2.198613 | 48.874248 | 917 | 12833 | 917 | 51.04 | 2.0 | 1 |
| 245000.0 | 26 | 2.235872 | 48.880499 | 904 | 12526 | 904 | 37.71 | 1.0 | 0 |
| 450000.0 | 26 | 2.244781 | 48.883831 | 909 | 12682 | 909 | 63.94 | 3.0 | 0 |

FIGURE 6.6 – Extrait dataset encodé - Hauts-de-Seine

## 6.2 Choix des hyperparamètres et protocole d'évaluation

Une fois le prétraitement des données achevé, nous avons procédé à la sélection des hyperparamètres pour les algorithmes retenus, ainsi qu'à la définition du protocole d'évaluation des performances.

### 6.2.1 Hyperparamètres

Pour rappel, nous avons sélectionné cinq modèles de *Machine Learning* :
— Régression linéaire
— Arbre de décision
— Forêt aléatoire (*Random Forest*)
— XGBoost (*Extreme Gradient Boosting*)
— CatBoost

Le modèle de régression linéaire est sensible aux différences d'échelle entre les variables. Par conséquent, nous avons appliqué une standardisation (scaling) des variables numériques à l'aide de `StandardScaler` afin d'assurer une convergence stable et de meilleurs coefficients.

Pour les autres modèles basés sur des arbres, qui sont robustes aux variations d'échelle, le scaling n'était pas nécessaire. Toutefois, il a été nécessaire de spécifier certains hyperparamètres :

— Arbre de décision : `max_depth=5`, `random_state=42` Ces valeurs permettent de limiter la profondeur de l'arbre afin d'éviter le surapprentissage (overfitting), tout en assurant la reproductibilité des résultats via le paramètre `random_state`.
— Random Forest : `n_estimators=100`, `random_state=42` Nous avons choisi 100 estimateurs comme compromis entre performance et temps de calcul. Le paramètre `random_state` est maintenu constant pour des raisons de reproductibilité.

— XGBRegressor : `n_estimators=100`, `learning_rate=0.1`, `max_depth=6`, `random_state=42`, `verbosity=0` Les hyperparamètres ont été choisis de manière empirique selon les recommandations usuelles : un taux d'apprentissage modéré, une profondeur raisonnable et une sortie silencieuse pour éviter l'affichage en console.
— CatBoostRegressor : `verbose=0`, `random_state=42` Nous avons désactivé les sorties de logs pour simplifier la lecture, et fixé le `random_state` pour garantir la reproductibilité.

### 6.2.2 Protocole d'évaluation

Pour évaluer les performances des différents modèles, nous avons recours à la validation croisée à 5 plis (5-fold cross-validation). Ce choix permet de réduire le risque de biais lié à un découpage arbitraire du jeu de données, tout en assurant un bon équilibre entre stabilité de l'évaluation et temps de calcul.

Nous avons également mis de côté 5% du jeu de données en tant que jeu de test final. L'objectif est double :
— D'une part, évaluer la performance des modèles via une *validation croisée à 5 plis* réalisée sur les 95% restants du dataset
— D'autre part, confronter ces modèles à des données jamais vues pendant l'apprentissage (les 5%) pour mesurer leur capacité de généralisation sur des cas réels.

Cette méthodologie permet de vérifier si les modèles sont *bien généralisables* : en comparant les scores $R^2$ moyens issus de la validation croisée à la performance obtenue sur le jeu de test final.

Pour évaluer les performances de nos modèles de prédiction, nous utilisons les métriques suivantes, complémentaires et classiques en régression :

| Métrique | Description | Objectif |
|---|---|---|
| **$R^2$** (coefficient de détermination) | Mesure la part de la variance de la variable cible expliquée par le modèle.<br>$R^2 \approx 1$ : prédictions proches des vraies valeurs.<br>$R^2 < 0$ : modèle moins performant qu'une prédiction naïve (par la moyenne). | Le plus proche possible de 1 |
| **MAE** (Mean Absolute Error) | Moyenne des erreurs absolues. Facilement interprétable : "en moyenne, le modèle se trompe de X unités".<br>Moins sensible aux valeurs aberrantes. | Le plus petit possible |
| **MSE** (Mean Squared Error) | Moyenne des carrés des erreurs.<br>Pénalise fortement les erreurs importantes (effet quadratique). | Le plus petit possible (attention à l'unité au carré) |
| **RMSE** (Root Mean Squared Error) | Racine carrée du MSE. Interprétation directe dans les unités de la cible.<br>Met en valeur les grosses erreurs tout en étant lisible. | Le plus petit possible |

TABLE 6.1 – Résumé des métriques utilisées pour l'évaluation des modèles

## 6.3 Résultats d'expériences - Prédiction prix

Nous présentons à présent les résultats obtenus au cours de nos différentes expérimentations, ainsi que leur interprétation.

### 6.3.1 Modélisation du prix (Colombes)

Le tableau 6.7 présente les résultats obtenus lors de l'entraînement des différents modèles de machine learning sur Colombes.

Nous constatons que les modèles ensemblistes, à savoir CatBoost, XGBoost et Random Forest, surpassent nettement les modèles plus simples tels que la régression linéaire et l'arbre de décision. En particulier, CatBoost obtient les meilleurs résultats avec un $R^2$ moyen de 0,858 ± 0,009.

La régression linéaire et l'arbre de décision, bien que simples à implémenter, atteignent un $R^2$ moyen plus faible (respectivement 0,769 et 0,762).

Indépendamment du modèle utilisé, la variable `lots_surface_logement` est systématiquement identifiée comme la plus importante, avec une importance variant de 0,51 (régression linéaire) à 0,92 (arbre de décision). Cela confirme que la surface habitable est un facteur central dans l'évaluation du prix d'un bien immobilier.

Les variables de localisation géographique (`latitude`, `longitude`) apparaissent également parmi les trois premières variables les plus influentes dans tous les modèles, soulignant l'impact fort de l'emplacement à l'intérieur de la commune de Colombes.

La variable `construction_recente`, est également significative, en particulier dans les modèles XGBoost, CatBoost. Cela suggère que l'ancienneté du bâtiment joue un rôle dans la valorisation du bien.

Enfin, certaines variables comme `code_secteur_ville`, `id_parcelle` ou `adresse_code_voie` sont présentes dans quelques modèles avec une importance relativement plus faible. Cela peut s'expliquer par leur redondance partielle avec les données de géolocalisation, ou par une faible variabilité discriminante.

Concernant les données de test (les 5 % du dataset mis de côté), les résultats sont résumés à travers les indicateurs suivants : *$R^2$ test*, *RMSE test*, *MSE test* et *MAE test*.

Nous observons que les scores de $R^2$ obtenus sur ce jeu de test sont cohérents avec ceux observés en validation croisée, ce qui est un bon indicateur de *généralisation* des modèles.

Le modèle le plus performant sur les données de test est, une fois de plus, CatBoost, avec un excellent score de *$R^2$ = 0,870*. Ses performances sur les autres métriques sont également très satisfaisantes :

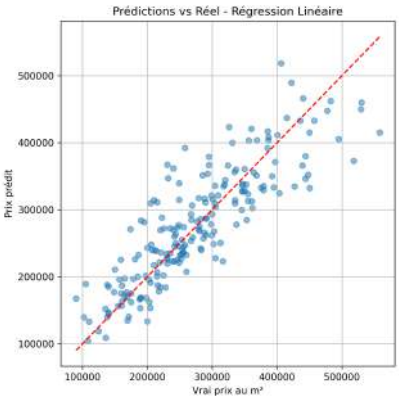
— RMSE test = 34 000,
— MSE test = 1 156 629 957,
— MAE test = 24 154.

Par ailleurs, les graphiques 6.8 associés permettent d'observer la correspondance visuelle entre les valeurs réelles et les valeurs prédites pour chacun des modèles.

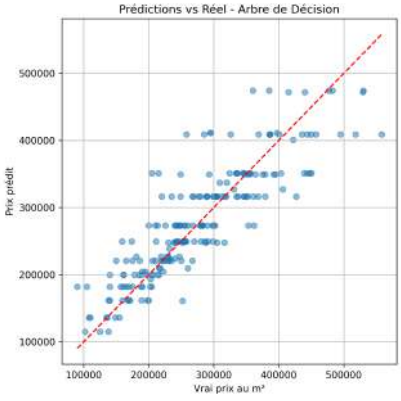
| Modèle | $R^2$ CV train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.769 ± 0.011 | 0.736 | 48470.13 | 2349353865.23 | 36644.40 | lots_surface_logement<br>Importance : 0.51 | latitude<br>Importance : 0.16 | construction_recente<br>Importance : 0.09 | id_parcelle<br>Importance : 0.06 | adresse_code_voie<br>Importance : 0.06 |
| Arbre de Décision | 0.762 ± 0.013 | 0.750 | 47120.58 | 2220349302.21 | 33806.41 | lots_surface_logement<br>Importance : 0.92 | latitude<br>Importance : 0.05 | longitude<br>Importance : 0.02 | id_parcelle<br>Importance : 0.01 | construction_recente<br>Importance : 0.01 |
| Random Forest | 0.836 ± 0.012 | 0.840 | 37762.63 | 1426015937.70 | 26923.84 | lots_surface_logement<br>Importance : 0.79 | latitude<br>Importance : 0.08 | longitude<br>Importance : 0.06 | id_parcelle<br>Importance : 0.03 | construction_recente<br>Importance : 0.02 |
| XGBoost | 0.850 ± 0.012 | 0.862 | 35004.58 | 1225320540.98 | 24367.60 | lots_surface_logement<br>Importance : 0.60 | construction_recente<br>Importance : 0.12 | latitude<br>Importance : 0.09 | longitude<br>Importance : 0.06 | id_parcelle<br>Importance : 0.05 |
| CatBoost | 0.858 ± 0.009 | 0.870 | 34009.26 | 1156629957.00 | 24154.47 | lots_surface_logement<br>Importance : 0.55 | latitude<br>Importance : 0.12 | longitude<br>Importance : 0.11 | construction_recente<br>Importance : 0.07 | id_parcelle<br>Importance : 0.06 |

FIGURE 6.7 – Résumé des performances et variables importantes par modèle (Colombes) 3

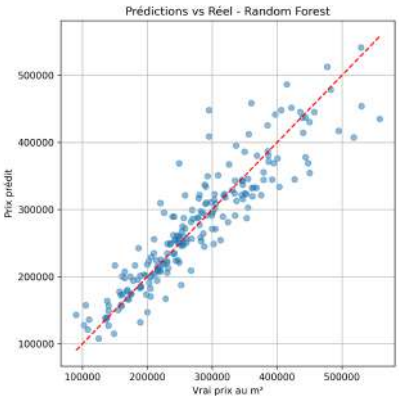
**Régression Linéaire**



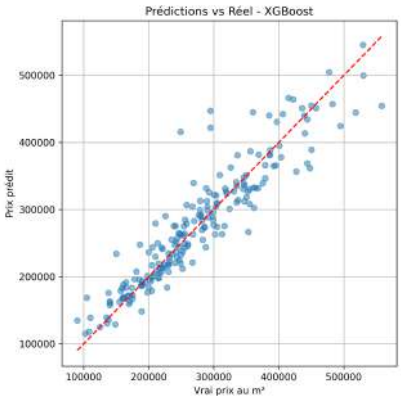

**Arbre de Décision**




**Random Forest**




**XGBoost**




**CatBoost**

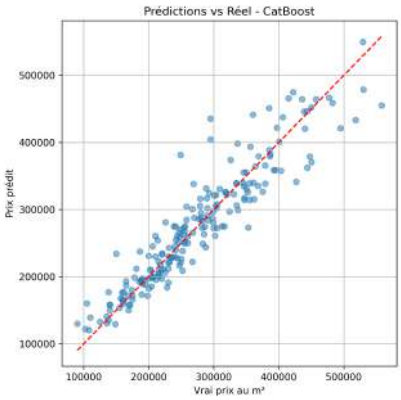



FIGURE 6.8 – Colombes - Prix - Prédictions vs Réel

### 6.3.2 Modélisation du prix (Hauts-de-Seine)

Le tableau 6.9 présente les résultats obtenus lors de l'entraînement des différents modèles de machine learning sur l'ensemble du département des Hauts-de-Seine.

Les modèles ensemblistes obtiennent, cette fois aussi, les meilleurs résultats. CatBoost atteint un $R^2$ moyen de 0,843 $\pm$ 0,004, suivi de près par Random Forest (0,839) et XGBoost (0,833).

À l'inverse, les modèles plus simples comme la régression linéaire ($R^2$ de 0,604) et l'arbre de décision ($R^2$ de 0,689) sous-performent. Cela suggère que les relations entre les variables explicatives et le prix sont non linéaires et nécessitent des approches plus complexes pour être modélisées efficacement.

La variable `lots_surface_logement` est, enocre une fois, la plus prédictive dans tous les modèles.

Les variables de localisation géographique, à savoir `latitude` et `longitude`, sont présentes dans tous les modèles dans le top 3, ce qui montre que le positionnement précis du bien dans le département influe fortement sur sa valeur.

La variable `id_parcelle` joue également un rôle important dans les modèles complexes.

Des variables comme `construction_recente`, `nom_commune`, et `nombre_pieces_principales` apportent une contribution secondaire mais peuvent améliorer la précision de certains modèles.

Concernant les données de test nous observons que les scores de $R^2$ obtenus sur ce jeu de test sont cohérents avec ceux observés en validation croisée, ce qui est un bon indicateur de *généralisation* des modèles.

Le modèle le plus performant sur les données de test est CatBoost, avec un excellent score de $R^2 = 0{,}845$. Ses performances sur les autres métriques sont également très satisfaisantes :

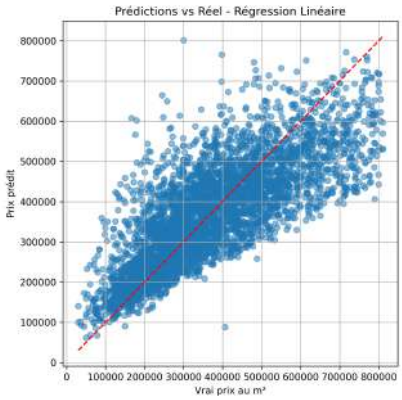
— RMSE test = 64 000,
— MSE test = 4 096 857 750,
— MAE test = 44 581.

Par ailleurs, les graphiques 6.10 associés permettent d'observer la correspondance visuelle entre les valeurs réelles et les valeurs prédites pour chacun des modèles.

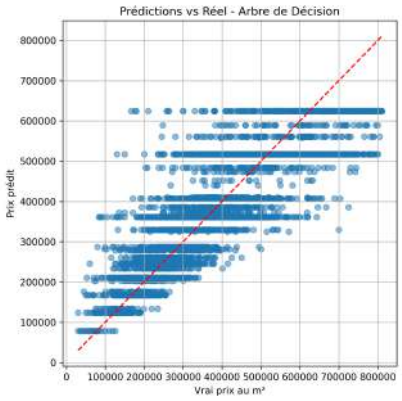
| Modèle | $R^2$ CV Train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.604 ± 0.003 | 0.600 | 102889.05 | 10586157387.61 | 77339.08 | lots_surface_logement<br>Importance : 0.49 | id_parcelle<br>Importance : 0.19 | adresse_code_voie<br>Importance : 0.06 | code_secteur_ville<br>Importance : 0.06 | nombre_pieces_principales<br>Importance : 0.06 |
| Arbre de Décision | 0.689 ± 0.001 | 0.690 | 90531.46 | 8195944672.31 | 67936.10 | lots_surface_logement<br>Importance : 0.82 | latitude<br>Importance : 0.12 | longitude<br>Importance : 0.04 | construction_recente<br>Importance : 0.01 | nom_commune<br>Importance : 0.00 |
| Random Forest | 0.839 ± 0.003 | 0.841 | 64915.26 | 4213990543.41 | 44201.99 | lots_surface_logement<br>Importance : 0.66 | latitude<br>Importance : 0.16 | longitude<br>Importance : 0.09 | id_parcelle<br>Importance : 0.04 | construction_recente<br>Importance : 0.02 |
| XGBoost | 0.833 ± 0.003 | 0.835 | 66074.50 | 4365838966.13 | 46278.01 | lots_surface_logement<br>Importance : 0.67 | latitude<br>Importance : 0.10 | construction_recente<br>Importance : 0.06 | longitude<br>Importance : 0.05 | nom_commune<br>Importance : 0.05 |
| CatBoost | 0.843 ± 0.004 | 0.845 | 64006.70 | 4096857750.36 | 44581.55 | lots_surface_logement<br>Importance : 0.44 | latitude<br>Importance : 0.29 | longitude<br>Importance : 0.13 | code_secteur_ville<br>Importance : 0.03 | construction_recente<br>Importance : 0.03 |

FIGURE 6.9 – Résumé des performances et variables importantes par modèle (Hauts-de-Seine) 4

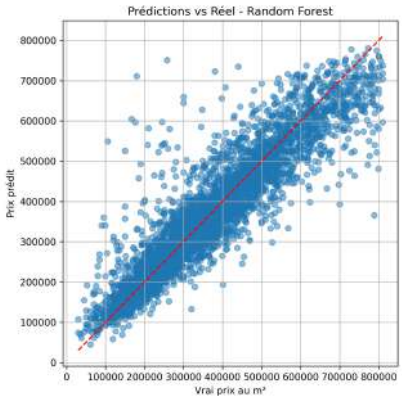
**Régression Linéaire**



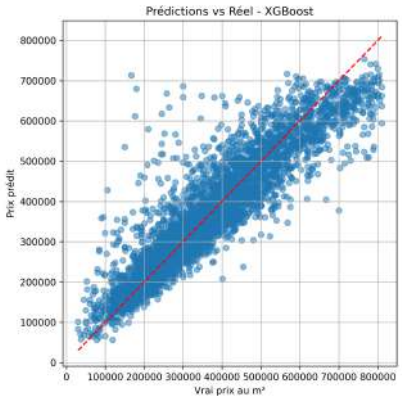

**Arbre de Décision**




**Random Forest**




**XGBoost**




**CatBoost**

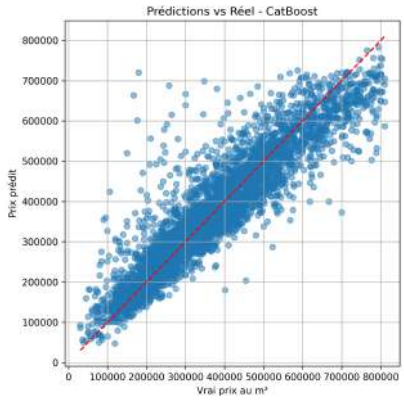



FIGURE 6.10 – 92 - Prix - Prédictions vs Réel

### 6.3.3 Analyse des résultats

Les résultats obtenus montrent que les modèles ensemblistes, notamment *Random Forest*, *XGBoost* et *CatBoost*, offrent d'excellentes performances pour la prédiction des prix immobiliers. Cette observation est en cohérence avec les conclusions récurrentes dans la littérature scientifique, où ces modèles sont fréquemment identifiés comme particulièrement adaptés aux données hétérogènes et non linéaires du marché immobilier.

Fait notable, nos modèles surpassent même certains résultats publiés dans des études antérieures. Par exemple, dans une analyse menée sur la ville de Bogota, le modèle *Random Forest* atteignait un $R^2$ d'environ 0,65[3], alors que dans notre cas, nous dépassons les 0,82. Cela confirme la pertinence de notre approche, tant sur le plan méthodologique que sur la qualité des données mobilisées. À ce stade, il serait tentant de considérer la tâche comme accomplie.

Cependant, un point particulier soulève une interrogation importante. Dans l'ensemble des modèles testés, la variable `lots_surface_logement` émerge systématiquement comme la plus déterminante dans la prédiction des prix. En d'autres termes, la surface du logement semble à elle seule capturer une grande partie de l'information nécessaire pour estimer le prix. Ce constat, bien que logique — la surface étant un facteur fondamental en immobilier — soulève une problématique : dans quelle mesure les modèles sont-ils réellement capables d'exploiter d'autres variables explicatives si la surface est absente ?

C'est précisément à ce niveau que notre travail propose une contribution originale par rapport aux travaux antérieurs. Nous avons souhaité aller plus loin que la simple prédiction de prix, en testant la robustesse et la valeur ajoutée des autres variables du jeu de données. Pour cela, nous avons formulé l'hypothèse suivante : *si l'on retire la variable surface du modèle, peut-on toujours prédire de manière fiable le prix au mètre carré, c'est-à-dire la valeur unitaire de l'appartement, en se basant uniquement sur les autres variables explicatives ?*

Concrètement, cela nous a amené à créer une nouvelle variable : le prix au mètre carré. Nous avons volontairement retiré la variable `prix`, puis entraîné de nouveaux modèles afin de tester leur capacité à prédire cette nouvelle cible. Cette approche vise à mieux comprendre l'influence réelle des autres variables, notamment celles liées à la localisation, à la construction ou encore à l'environnement urbain.

Ce travail ouvre ainsi une nouvelle perspective : au lieu de modéliser uniquement le prix absolu d'un logement, nous explorons, dans la prochaine partie, la possibilité de prédire son prix au mètre carré, dans le but de s'affranchir de la dépendance directe à la variable `surface`.

## 6.4 Nouvelle perspective - Prédiction prix m2

Nous abordons à présent l'estimation du prix au mètre carré.

### 6.4.1 Préparation des données

La préparation des données pour cette nouvelle expérience suit globalement le même processus que celui décrit précédemment pour la prédiction du prix. Toutefois, une modification majeure est apportée : nous introduisons une nouvelle variable cible, le prix au mètre carré, calculé comme le ratio entre la `valeur_fonciere` et la `surface du logement`. Une fois cette variable créée, les variables `valeur_fonciere` et `lots_surfac_logement` sont retirées du jeu de données afin d'éviter toute fuite de données (*data leakage*) et de garantir que les modèles n'utilisent pas ces valeurs pour reconstituer indirectement le prix au mètre carré.

Les figures 6.11 6.12 suivantes permettent de visualiser la distribution des prix au mètre carré pour les communes de Colombes et du département des Hauts-de-Seine.

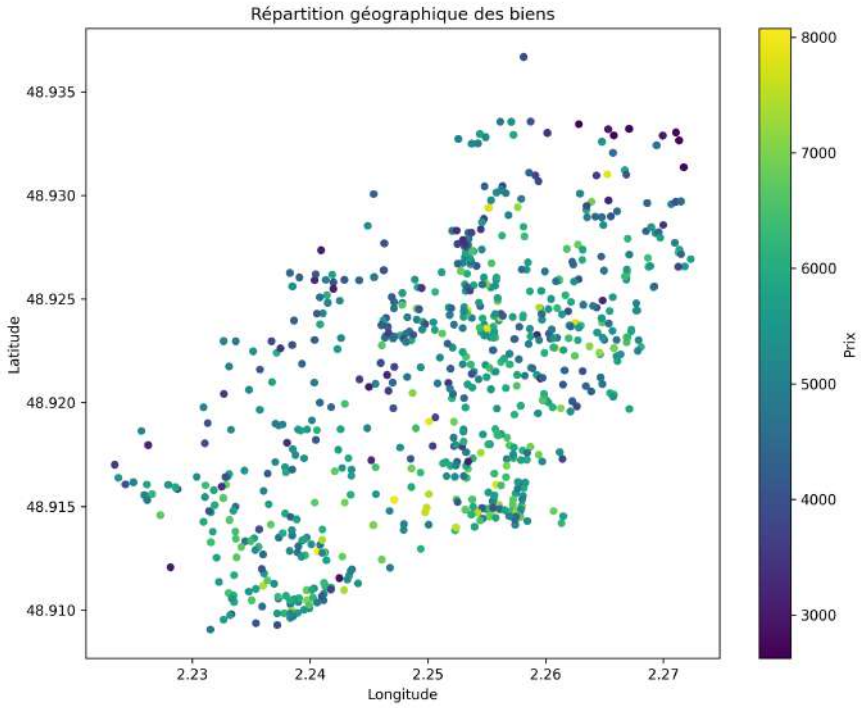



FIGURE 6.11 – Distribution du prix au mètre carré — Colombes

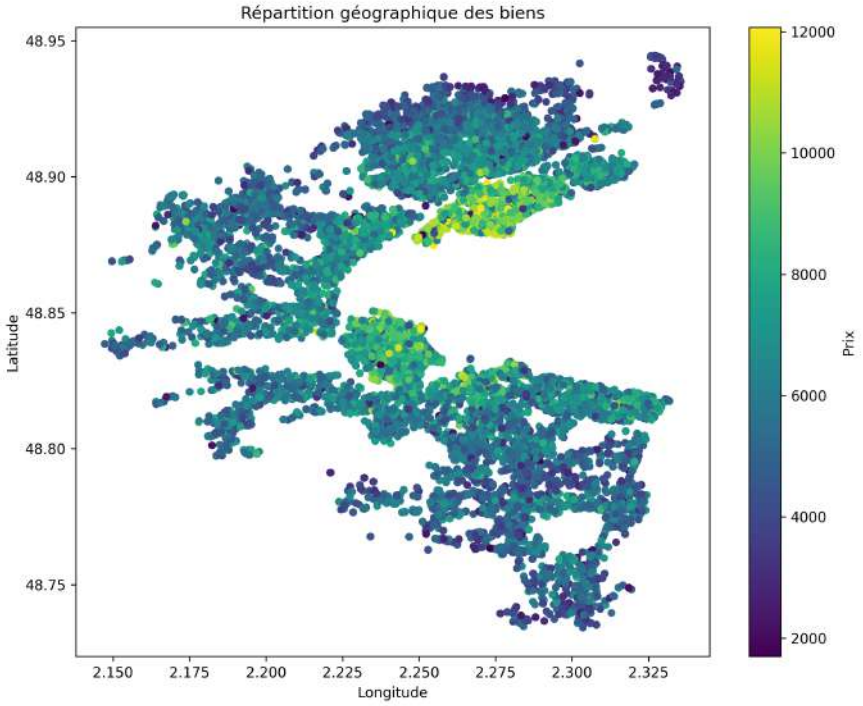



FIGURE 6.12 – Distribution du prix au mètre carré — Hauts-de-Seine

### 6.4.2 Résultats — Modélisation du prix au mètre carré (Colombes)

Le tableau 6.13 présente les résultats obtenus lors de l'entraînement des différents modèles de machine learning sur Colombes.

Au cours de la phase d'entraînement avec validation croisée, les scores de $R^2$ étaient globalement modestes. Le meilleur modèle, CatBoost, n'atteint qu'un score de $R^2$ de 0,383, ce qui reste relativement faible.

Cependant, lors de l'évaluation sur les données de test (les 5 % mis de côté), les performances des modèles se révèlent légèrement meilleures :

- Random Forest : $R^2 = 0{,}430$
- XGBoost : $R^2 = 0{,}484$
- CatBoost : $R^2 = 0{,}504$

Parmi les variables les plus importantes dans cette nouvelle configuration, on observe :

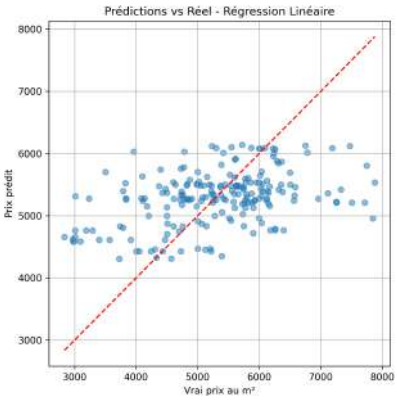
- Latitude et longitude, qui apparaissent systématiquement en tête, ce qui montre l'importance cruciale de la localisation dans la formation du prix au mètre carré.
- `id_parcelle`, qui capte probablement des effets micro-locatifs précis (proximité d'une rue commerçante, d'un parc, etc.).
- D'autres variables comme `construction_recente` ou `code_secteur_ville` émergent également comme significatives.

Les graphiques 6.16 associés permettent d'observer la correspondance visuelle entre les valeurs réelles et les valeurs prédites pour chacun des modèles.

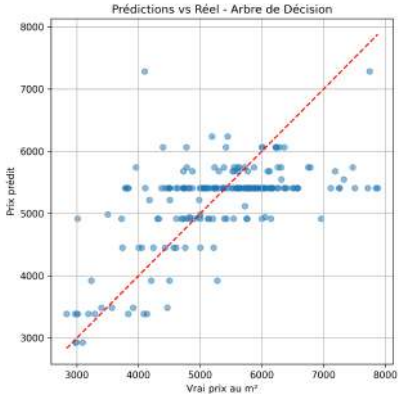
| Modèle | $R^2$ CV train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.175 ± 0.034 | 0.178 | 938.85 | 881445.01 | 734.30 | latitude<br>Importance : 0.34 | id_parcelle<br>Importance : 0.16 | construction_recente<br>Importance : 0.15 | adresse_code_voie<br>Importance : 0.14 | code_secteur_ville<br>Importance : 0.14 |
| Arbre de Décision | 0.288 ± 0.041 | 0.358 | 829.58 | 688202.28 | 625.50 | latitude<br>Importance : 0.48 | longitude<br>Importance : 0.43 | id_parcelle<br>Importance : 0.08 | code_secteur_ville<br>Importance : 0.01 | construction_recente<br>Importance : 0.00 |
| Random Forest | 0.358 ± 0.064 | 0.430 | 781.47 | 610696.86 | 577.46 | latitude<br>Importance : 0.37 | longitude<br>Importance : 0.36 | id_parcelle<br>Importance : 0.20 | construction_recente<br>Importance : 0.05 | code_secteur_ville<br>Importance : 0.02 |
| XGBoost | 0.380 ± 0.051 | 0.484 | 743.72 | 553114.15 | 548.93 | latitude<br>Importance : 0.26 | longitude<br>Importance : 0.20 | id_parcelle<br>Importance : 0.19 | construction_recente<br>Importance : 0.18 | code_secteur_ville<br>Importance : 0.17 |
| CatBoost | 0.383 ± 0.054 | 0.504 | 729.10 | 531592.96 | 544.59 | latitude<br>Importance : 0.39 | longitude<br>Importance : 0.27 | id_parcelle<br>Importance : 0.14 | construction_recente<br>Importance : 0.11 | adresse_code_voie<br>Importance : 0.05 |

FIGURE 6.13 – Résumé des performances et variables importantes par modèle (Colombes) 5

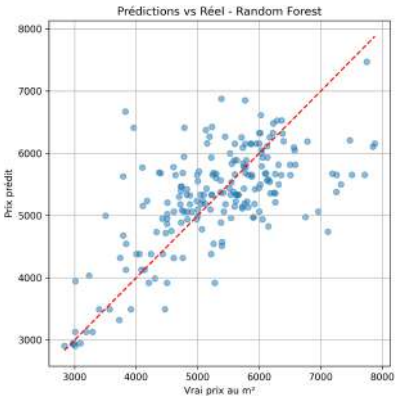
**Régression Linéaire**



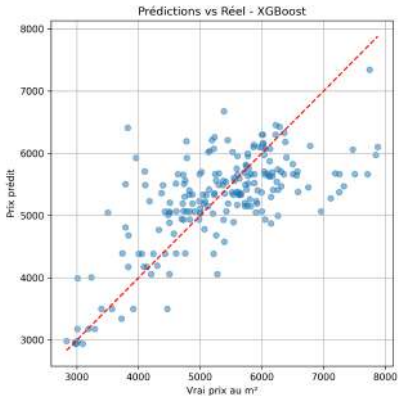

**Arbre de Décision**




**Random Forest**




**XGBoost**




**CatBoost**

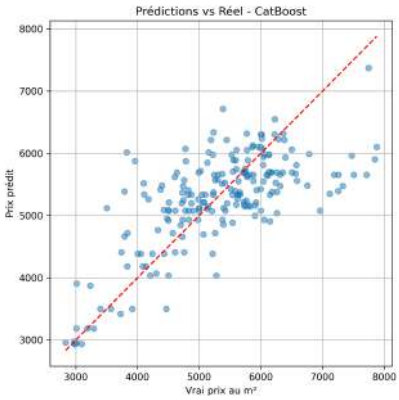



FIGURE 6.14 – Colombes - Prix m2 - Prédictions vs Réel

### 6.4.3 Résultats — Modélisation du prix au mètre carré (Hauts-de-Seine)

Le tableau 6.15 présente les résultats obtenus lors de l'entraînement des différents modèles de machine learning sur l'ensemble du département des Hauts-de-Seine.

Les modèles d'ensemble, à savoir Random Forest, XGBoost et CatBoost, affichent des performances nettement supérieures pour la prédiction du prix au mètre carré dans le département des Hauts-de-Seine par rapport à la ville de Colombes. Ces modèles atteignent de bons scores de $R^2$ :

— Random Forest : $R^2 = 0{,}602$
— XGBoost : $R^2 = 0{,}580$
— CatBoost : $R^2 = 0{,}591$

En comparaison, la régression linéaire montre une inadéquation totale à ce type de tâche dans ce contexte, avec un score de $R^2 = 0{,}020$. L'arbre de décision atteint un $R^2 = 0{,}412$, ce qui le positionne entre les deux groupes de performance.

Concernant l'importance des variables dans les modèles les plus performants, les variables suivantes ressortent de manière récurrente :

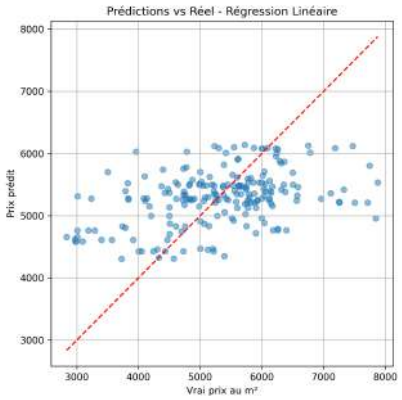
— Latitude et Longitude, qui confirment à nouveau le rôle central de la localisation géographique dans la formation du prix au mètre carré.
— Nom de la commune (`nom_commune`), qui reflète les différences structurelles entre les marchés immobiliers des différentes villes du département.
— `Construction_recente`, `id_parcelle` et `adresse_code_voie`, qui apportent des informations plus fines sur la nature ou la localisation précise des biens.

Les graphiques 6.16 associés permettent d'observer la correspondance visuelle entre les valeurs réelles et les valeurs prédites pour chacun des modèles.

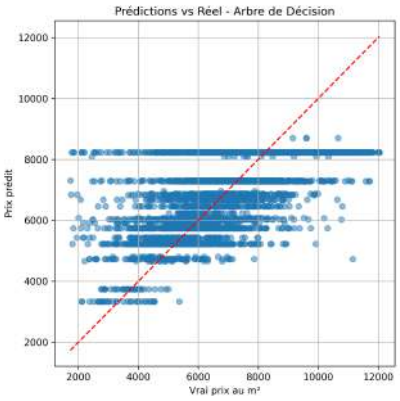
| Modèle | $R^2$ CV train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.020 ± 0.001 | 0.016 | 1842.85 | 3396082.47 | 1460.05 | id_parcelle<br>importance : 0.43 | code_secteur_ville<br>importance : 0.14 | adresse_code_voie<br>importance : 0.14 | nom_commune<br>importance : 0.11 | latitude<br>importance : 0.08 |
| Arbre de Décision | 0.412 ± 0.005 | 0.404 | 1434.68 | 2058310.04 | 1082.02 | latitude<br>importance : 0.64 | longitude<br>importance : 0.28 | construction_recente<br>importance : 0.05 | code_secteur_ville<br>importance : 0.01 | id_parcelle<br>importance : 0.01 |
| Random Forest | 0.602 ± 0.004 | 0.604 | 1168.66 | 1365760.14 | 833.49 | latitude<br>importance : 0.48 | longitude<br>importance : 0.28 | id_parcelle<br>importance : 0.15 | construction_recente<br>importance : 0.05 | code_secteur_ville<br>importance : 0.02 |
| XGBoost | 0.580 ± 0.004 | 0.577 | 1208.65 | 1460830.90 | 890.86 | construction_recente<br>importance : 0.26 | latitude<br>importance : 0.26 | longitude<br>importance : 0.22 | id_parcelle<br>importance : 0.12 | adresse_code_voie<br>importance : 0.07 |
| CatBoost | 0.591 ± 0.004 | 0.591 | 1188.95 | 1413608.96 | 870.96 | latitude<br>importance : 0.49 | longitude<br>importance : 0.26 | id_parcelle<br>importance : 0.08 | construction_recente<br>importance : 0.05 | code_secteur_ville<br>importance : 0.05 |

FIGURE 6.15 – Résumé des performances et variables importantes par modèle (Hauts-de-Seine) 6

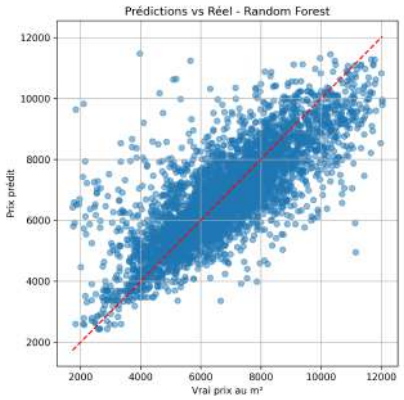
**Régression Linéaire**



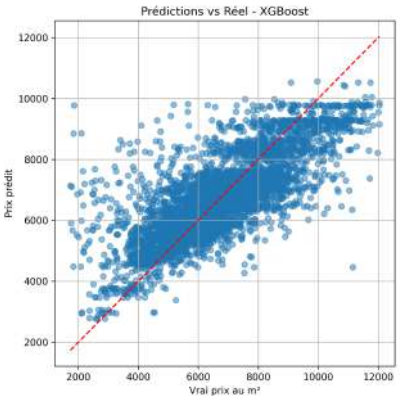

**Arbre de Décision**




**Random Forest**




**XGBoost**




**CatBoost**

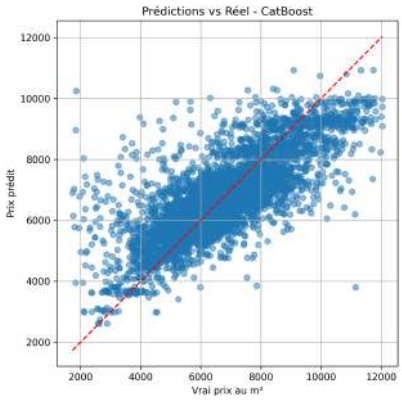



FIGURE 6.16 – 92 - Prix m2 - Prédictions vs Réel

### 6.4.4 Analyse des résultats

Les résultats confirment une observation attendue : sans la variable "surface du logement", les modèles rencontrent plus de difficultés à prédire correctement le prix. La performance globale diminue, ce qui souligne le poids considérable de la surface dans la formation du prix absolu d'un bien immobilier. Malgré cela, les modèles ensemblistes (Random Forest, XGBoost, CatBoost) continuent de surperformer les approches plus simples comme la régression linéaire ou les arbres de décision isolés.

Par ailleurs, on note que les variables les plus importantes pour la prédiction du prix au mètre carré sont principalement liées à la localisation : `latitude`, `longitude`, ou encore `nom_commune`. Cela illustre le fait que la valeur immobilière au mètre carré est fortement déterminée par la position géographique du bien.

Un autre enseignement clé concerne la comparaison entre deux périmètres géographiques :

— Les performances obtenues sur l'ensemble du département des Hauts-de-Seine sont significativement meilleures que celles obtenues sur la seule ville de Colombes.

Cela suggère que l'élargissement du périmètre géographique permet aux modèles d'apprendre sur une diversité plus riche de contextes urbains et donc d'extraire des relations plus robustes et pertinentes.

Concernant les résultats obtenus pour Colombes, bien que les scores obtenus lors de la validation croisée soient restés relativement faibles, on observe que les modèles ont su *mieux généraliser sur les données de test*. Cette tendance est encourageante et laisse penser que les performances ont pu être sous-estimées pendant la validation croisée, potentiellement à cause d'une variabilité interne élevée ou d'une répartition non homogène des observations au sein du jeu de données.

Malgré cette légère amélioration sur les données de test (avec un score maximal de 0,504 en $R^2$ pour le modèle CatBoost), les performances globales restent modestes. Cela suggère que les variables actuellement disponibles ne suffisent pas à capturer l'ensemble des facteurs qui influencent le prix au mètre carré dans cette commune.

Face à cette limite, une question se pose : peut-on améliorer la capacité prédictive de nos modèles en intégrant de nouvelles variables explicatives ? Parmi les variables susceptibles d'être ajoutées, celles relatives au contexte urbain et à l'environnement immédiat du bien immobilier apparaissent comme particulièrement pertinentes. Elles sont non seulement accessibles via des sources ouvertes (notamment OpenStreetMap), mais également susceptibles de refléter des dimensions essentielles de l'attractivité d'un logement.

C'est précisément dans cette perspective que s'inscrit la prochaine section. Nous y explorons l'effet de l'enrichissement du jeu de données par des variables de proximité géographique (présence de transports, d'établissements scolaires, de commerces, etc.), dans le but d'évaluer leur contribution à la performance des modèles de prédiction.

## 6.5 Prédiction prix m2 - données enrichies

Afin d'améliorer les performances des modèles sur des zones géographiques restreintes (comme la ville de Colombes), nous avons entrepris un enrichissement du jeu de données en intégrant des informations contextuelles issues de l'environnement immédiat des logements.

Nous avons choisi d'ajouter des Points of Interest (POI) — c'est-à-dire des infrastructures et services à proximité pouvant influer sur la valeur immobilière — à partir de données en open source issues du projet *OpenStreetMap* (répertoire `FR:Points d'intérêt`).

Un script Python a été développé pour extraire automatiquement, pour chaque logement :

— La distance (en kilomètres) entre le bien et le centre-ville le plus proche ;
— Le nombre d'écoles situées dans un rayon de 500 mètres ;
— Le nombre de gares accessibles dans un rayon de 500 mètres ;
— Le nombre de supermarchés dans un rayon de 500 mètres ;
— Le nombre de centres commerciaux dans un rayon de 500 mètres ;
— Le nombre d'hôpitaux présents à moins de 500 mètres.

Le seuil de 500 mètres a été retenu comme une distance significative à l'échelle piétonne, représentant environ trois minutes de marche. Cette proximité immédiate est particulièrement pertinente en milieu urbain dense, où l'accessibilité rapide à certains services ou équipements peut avoir un impact notable sur la valorisation d'un bien.

D'autre part, dans un souci de neutralité vis-à-vis de la variable "surface_logement", nous avions initialement choisi de la retirer du jeu de données afin d'éviter que le prix ne soit fortement corrélé à cette seule information. Cependant, au cours de l'analyse, une observation importante a émergé : le prix au mètre carré dépend également du type d'appartement.

En effet, selon la typologie du logement (studio, T2, T3, etc.), les dynamiques de marché varient :

— Les petits logements (T1/T2) affichent généralement un prix au mètre carré plus élevé, en raison d'une forte demande locative.
— Les logements intermédiaires (T3/T4) se situent dans une gamme de prix au mètre carré plus moyenne.
— Les grands appartements (T5 et plus) ont souvent un prix au mètre carré plus faible, car leur demande est moins soutenue et ils s'adressent à un public plus restreint.

Cette double relation entre surface et prix (valeur absolue et prix au mètre carré) révèle l'existence d'un effet de structure qui n'était pas initialement pris en compte. Pour remédier à cela, nous avons décidé d'introduire une nouvelle variable catégorielle : `type_appart`.

Cette variable a été déduite automatiquement à partir de la surface du logement, selon un barème simple :

— Surface $< 30$ m$^2$ : T1
— $30$ m$^2 <$ Surface $< 45$ m$^2$ : T2
— $45$ m$^2 <$ Surface $< 65$ m$^2$ : T3
— $65$ m$^2 <$ Surface $< 85$ m$^2$ : T4
— $85$ m$^2 <$ Surface $< 100$ m$^2$ : T5
— Surface $> 100$ m$^2$ : T6 et plus

Une fois cette typologie extraite, nous avons supprimé la variable `surface_logement`.

Un extrait du nouveau jeu de données enrichi est présenté ci-dessous :

| prix_m2 | longitude | latitude | code_secteur_ville | id_parcelle | adresse_code_voie | construction_recente | type_appart | centre_ville_distance | nb_ecole_proximite_500m | nb_gare_proximite_500m | nb_supermarket_500m | nb_centrecommercial_proximite_500m | nb_hospital_500m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4827.280779450842 | 2.238498 | 48.925608 | 92025000BX | 92025000BX0193 | 4628 | 0 | T3 | 1.217337274095387 | 6 | 0 | 1 | 0 | 1 |
| 4932.382352941177 | 2.252587 | 48.932724 | 92025000B | 92025000B0252 | 10 | 1 | T4 | 1.1427448468794288 | 3 | 0 | 0 | 0 | 0 |
| 5007.394886963871 | 2.243444 | 48.915559 | 92025000AT | 92025000AT0227 | 5220 | 0 | T3 | 1.1172640747290452 | 6 | 0 | 0 | 0 | 0 |
| 3808.1819725004248 | 2.246255 | 48.926389 | 92025000BS | 92025000BS0012 | 7165 | 0 | T3 | 0.7378413689384381 | 11 | 0 | 2 | 0 | 0 |
| 5578.598092077976 | 2.233084 | 48.910817 | 92025000CF | 92025000CF0207 | 135 | 0 | T3 | 2.035422423177509 | 3 | 0 | 3 | 1 | 0 |
| 5620.042262678803 | 2.233542 | 48.91039 | 92025000CF | 92025000CF0198 | 1564 | 0 | T3 | 2.040806979061709 | 5 | 1 | 4 | 1 | 0 |
| 4386.509414442376 | 2.246603 | 48.923661 | 92025000BK | 92025000BK0158 | 65 | 0 | T3 | 0.5878122929267999 | 8 | 0 | 3 | 0 | 0 |

FIGURE 6.17 – Extrait dataset enrichie - Colombes 9

| prix_m2 | nom_commune | longitude | latitude | code_secteur_ville | id_parcelle | adresse_code_voie | construction_recente | type_appart | centre_ville_distance | nb_ecole_proximite_500m | nb_gare_proximite_500m | nb_supermarket_500m | nb_centrecommercial_proximite_500m | nb_hospital_500m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4210.836277974087 | Clichy | 2.305108 | 48.903617 | 920240000K | 920240000K0192 | 6140 | 0 | T1 | 0.0908222871889472 | 13 | 1 | 10 | 0 | 1 |
| 8250.0 | Clichy | 2.311397 | 48.901329 | 920240000Z | 920240000Z0088 | 3560 | 1 | T4 | 0.5026264447770217 | 9 | 1 | 11 | 0 | 2 |
| 7077.435470441299 | Clichy | 2.30897 | 48.906286 | 920240000J | 920240000J0031 | 4122 | 0 | T2 | 0.487457018458497 | 10 | 1 | 8 | 0 | 2 |
| 4438.307525396982 | Clichy | 2.306151 | 48.896005 | 92024000AF | 92024000AF0022 | 1411 | 0 | T6+ | 0.7621126732340674 | 1 | 0 | 5 | 0 | 0 |
| 6956.8674219837 | Clichy | 2.306581 | 48.900427 | 920240000R | 920240000R0096 | 4420 | 0 | T3 | 0.2916379934660225 | 10 | 1 | 11 | 0 | 1 |
| 8068.181818181818 | Clichy | 2.300614 | 48.90294 | 920240000L | 920240000L0019 | 3525 | 1 | T2 | 0.3159998290322829 | 8 | 1 | 9 | 0 | 0 |
| 7377.777777777777 | Clichy | 2.312447 | 48.899447 | 920240000Y | 920240000Y0017 | 1490 | 0 | T2 | 0.6666481774461893 | 4 | 0 | 8 | 0 | 2 |
| 2589.834898025251 | Clichy | 2.301019 | 48.902312 | 920240000L | 920240000L0088 | 3535 | 0 | T2 | 0.2912864874895379 | 8 | 1 | 9 | 0 | 0 |

FIGURE 6.18 – Extrait dataset enrichie - Hauts-de-Seine 10

### 6.5.1 Résultats — Modélisation du prix au mètre carré - données enrichies

L'analyse des figures 6.19 et 6.20 met en évidence une amélioration significative des performances des modèles de machine learning suite à l'enrichissement du jeu de données, notamment par l'ajout de variables de typologie (`type_appart`) et de proximité géographique (`centre_ville_distance`, `nb_centre_commercial_proximité`, etc.).

**Pour Colombes** : Lors de la validation croisée, l'ajout de ces variables a permis aux modèles ensemblistes (`Random Forest`, `XGBoost`, `CatBoost`) d'améliorer leur score $R^2$ de plus de 10 points de pourcentage. C'est un gain non négligeable, bien que le meilleur score obtenu (environ $R^2 = 0{,}474$ pour `CatBoost`) reste modeste. Cela suggère une forte variabilité interne au sein des données, difficile à capturer entièrement avec les variables explicatives disponibles, même après enrichissement.

Cependant, les résultats obtenus sur les données de test sont plus encourageants. Le score $R^2$ du meilleur modèle (CatBoost) atteint 0,625, contre seulement 0,504 sans enrichissement. Cela indique que, malgré une validation croisée moyennement concluante, les modèles sont capables de généraliser efficacement sur des données encore jamais vues, ce qui constitue un résultat très satisfaisant.

**Pour les Hauts-de-Seine** : Les modèles étaient déjà performants avant l'ajout des nouvelles variables (par exemple, $R^2 = 0{,}602$ pour `Random Forest`). L'enrichissement a apporté une amélioration marginale, de l'ordre de 1 à 2 points de pourcentage. En revanche, on note une nette progression pour la `régression linéaire`, initialement inefficace ($R^2 \approx 0$), qui atteint désormais un score de $R^2 = 0{,}241$. Cela montre que les nouvelles variables sont informatives, mais leur apport est davantage exploité par les modèles simples que par les modèles ensemblistes, déjà bien calibrés sur les données initiales.

| Modèle | R² CV train | R² test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.289 ± 0.043 | 0.363 | 826.57 | 683210.14 | 649.81 | id_parcelle<br>Importance : 0.22 | latitude<br>Importance : 0.21 | construction_recente<br>Importance : 0.14 | type_appart<br>Importance : 0.11 | centre_ville_distance<br>Importance : 0.10 |
| Arbre de Décision | 0.334 ± 0.046 | 0.467 | 756.20 | 571834.10 | 572.91 | latitude<br>Importance : 0.31 | type_appart<br>Importance : 0.28 | centre_ville_distance<br>Importance : 0.15 | construction_recente<br>Importance : 0.10 | id_parcelle<br>Importance : 0.07 |
| Random Forest | 0.428 ± 0.055 | 0.545 | 698.73 | 488219.42 | 506.41 | latitude<br>Importance : 0.23 | type_appart<br>Importance : 0.17 | centre_ville_distance<br>Importance : 0.16 | longitude<br>Importance : 0.14 | id_parcelle<br>Importance : 0.13 |
| XGBoost | 0.467 ± 0.055 | 0.615 | 642.92 | 413344.61 | 476.25 | code_secteur_ville<br>Importance : 0.15 | latitude<br>Importance : 0.11 | construction_recente<br>Importance : 0.11 | commercial_proximite<br>Importance : 0.11 | centre_ville_distance<br>Importance : 0.10 |
| CatBoost | 0.474 ± 0.056 | 0.625 | 633.71 | 401588.08 | 471.58 | latitude<br>Importance : 0.20 | type_appart<br>Importance : 0.17 | centre_ville_distance<br>Importance : 0.15 | longitude<br>Importance : 0.12 | construction_recente<br>Importance : 0.12 |

FIGURE 6.19 – Résumé des performances et variables importantes par modèle (Colombes) 7

| Modèle | $R^2$ CV train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.261 ± 0.008 | 0.276 | 1571.10 | 2468364.37 | 1231.07 | id_parcelle<br>importance : 0.18 | nb_supermarket_500m<br>importance : 0.16 | type_appart<br>importance : 0.09 | construction_recente<br>importance : 0.09 | code_secteur_ville<br>importance : 0.08 |
| Arbre de Décision | 0.402 ± 0.005 | 0.413 | 1414.93 | 2002036.85 | 1089.92 | latitude<br>importance : 0.42 | nb_supermarket_500m<br>importance : 0.29 | longitude<br>importance : 0.11 | centre_ville_distance<br>importance : 0.07 | construction_recente<br>importance : 0.06 |
| Random Forest | 0.613 ± 0.008 | 0.648 | 1095.26 | 1199595.95 | 775.58 | latitude<br>importance : 0.30 | nb_supermarket_500m<br>importance : 0.15 | longitude<br>importance : 0.14 | id_parcelle<br>importance : 0.11 | centre_ville_distance<br>importance : 0.10 |
| XGBoost | 0.601 ± 0.006 | 0.618 | 1141.02 | 1301929.61 | 832.89 | nb_supermarket_500m<br>importance : 0.30 | latitude<br>importance : 0.15 | construction_recente<br>importance : 0.13 | non_commune<br>importance : 0.08 | code_secteur_ville<br>importance : 0.06 |
| CatBoost | 0.622 ± 0.007 | 0.647 | 1096.41 | 1202110.72 | 796.82 | latitude<br>importance : 0.38 | longitude<br>importance : 0.16 | nb_supermarket_500m<br>importance : 0.07 | centre_ville_distance<br>importance : 0.07 | construction_recente<br>importance : 0.05 |

FIGURE 6.20 – Résumé des performances et variables importantes par modèle (Hauts-de-Seine) 8

### 6.5.2 Analyse des résultats

Ces résultats confirment plusieurs tendances :

— Dans le cas des Hauts-de-Seine, les performances sont bonnes avec ou sans enrichissement. Cela s'explique probablement par une diversité géographique et socio-économique plus riche dans le département, qui permet aux modèles de mieux apprendre les dynamiques locales, même sans variables de contexte supplémentaires.

— À Colombes, l'enrichissement a un effet plus marqué : on observe une nette amélioration de la qualité des prédictions, notamment sur les données de test. Cela montre que, dans des territoires plus restreints, où les données sont plus homogènes mais aussi plus sensibles à des facteurs contextuels locaux, l'ajout d'informations environnementales devient crucial. Toutefois, le niveau de performance reste moyen en validation croisée ($R^2 \approx 0{,}5$), ce qui laisse supposer que certaines sources de variabilité ne sont pas encore bien modélisées.

# Chapitre 7

# Conlusion

## Synthèse et analyse des résultats

L'objectif principal de notre étude était d'évaluer dans quelle mesure les modèles de machine learning peuvent prédire efficacement le prix des appartements dans le département des Hauts-de-Seine.

### Premiers constats — Prédiction du prix brut

Les premières expériences, portant sur la prédiction du prix des logements, ont donné lieu à plusieurs observations importantes :

- Les modèles entraînés (notamment les modèles ensemblistes comme `Random Forest`, `XGBoost` et `CatBoost`) ont atteint de très bons scores de prédiction, avec des $R^2$ dépassant les 0,80, aussi bien sur la commune de Colombes (zone restreinte) que sur l'ensemble des Hauts-de-Seine. Cela montre que les modèles sont globalement capables d'estimer avec précision le prix des logements sur ces territoires.
- Cependant, une analyse plus fine des variables explicatives a révélé que la variable `surface` expliquait à elle seule une grande part de la variance du prix (souvent plus de 50 %). Cela pose la question suivante : *les modèles peuvent-ils rester performants si cette variable dominante est retirée ?*

### Deuxième étape — Vers la prédiction du prix au mètre carré

Afin de répondre à cette interrogation, nous avons construit une nouvelle variable : le prix au mètre carré, obtenu en divisant la `valeur foncière` par la `surface` du logement. Ensuite, pour éviter toute redondance ou fuite d'information, nous avons retiré les variables `surface` et `valeur foncière` du jeu de données.

Les résultats ont mis en lumière les éléments suivants :

- Sur les Hauts-de-Seine, les modèles ont conservé de bons résultats, avec un $R^2$ avoisinant les 0,60 pour les meilleurs modèles. Cela démontre une capacité des algorithmes à estimer le prix au mètre carré, même sans l'information explicite de la surface.
- À l'inverse, sur Colombes, les performances ont été plus limitées. Les scores en validation croisée restaient faibles ($R^2$ entre 0,35 et 0,40), bien qu'une amélioration ait été observée lors de la prédiction sur les données de test ($R^2$ autour

de 0,50). Cela suggère une certaine hétérogénéité du marché local et un manque de variables explicatives suffisamment discriminantes dans notre dataset.

## Troisième étape — Enrichissement des données

Pour tenter d'améliorer ces résultats, nous avons enrichi notre base de données avec des données géographiques de proximité, issues d'OpenStreetMap (POI), ainsi qu'une nouvelle variable de `typologie` du logement (`type_appart`), estimée à partir de la surface.

Les impacts de cet enrichissement ont été les suivants :

— Dans les Hauts-de-Seine, les performances des modèles sont restées stables, avec des scores comparables à ceux obtenus sans enrichissement ($R^2 > 0{,}60$). Cela peut s'expliquer par la richesse déjà présente dans le jeu de données initial et par la diversité géographique plus importante à l'échelle du département.

— Sur Colombes, l'enrichissement a eu un effet positif marqué. Les scores en validation croisée ont progressé de 10 à 15 points de pourcentage, atteignant jusqu'à 0,474, et les performances sur les données de test se sont nettement améliorées (jusqu'à 0,625 en $R^2$ pour le modèle `Random Forest`). Cela montre que les variables de contexte urbain ont une réelle influence sur la capacité à prédire le prix au mètre carré à Colombes.

## Limites et perspectives

Malgré ces améliorations, les performances des modèles à Colombes restent en deçà de celles observées pour les Hauts-de-Seine. Cela laisse penser qu'il manque encore des variables explicatives clés dans notre jeu de données. En effet, nous ne disposons actuellement d'aucune information sur :

— L'état général du bien (neuf, à rénover, travaux à prévoir, etc.) ;

— L'année de construction (ancien vs. récent) ;

— Les performances énergétiques (DPE, GES) ;

— La présence d'aménagements spécifiques (ascenseur, balcon, parking, etc.).

Ces éléments, s'ils étaient disponibles, pourraient fortement améliorer la prédiction du prix au mètre carré, en particulier dans des zones comme Colombes où les variations de prix sont difficiles à expliquer avec les seules variables disponibles.

Ainsi, même si l'objectif principal de ce mémoire — démontrer la capacité des modèles de machine learning à prédire efficacement les prix immobiliers — a bien été atteint, cette étude ouvre également des perspectives intéressantes. En particulier, elle met en lumière l'importance de l'enrichissement du jeu de données, non seulement avec des données géographiques (POI), mais aussi avec des informations immobilières plus qualitatives.

Il est aussi intéressant de noter que le choix initial d'analyser séparément les prix à Colombes, puis dans l'ensemble des Hauts-de-Seine, s'est révélé pertinent. Ce découpage, pensé à l'origine pour des raisons de faisabilité technique et de montée en complexité progressive, a finalement permis de tirer des enseignements plus profonds sur la performance des modèles selon l'étendue géographique et la diversité des données. Cela montre qu'une analyse localisée peut révéler des limites que l'on ne perçoit pas à plus grande échelle — et offre ainsi des pistes concrètes d'amélioration pour de futures recherches ou applications professionnelles.

# Retour d'expérience

Cette expérience a été très formatrice, tant sur le plan technique que méthodologique. Plusieurs erreurs commises au cours du projet m'ont permis de mieux comprendre les bonnes pratiques à adopter dans le cadre d'une démarche d'analyse de données et de modélisation prédictive. Voici un retour structuré sur les principales difficultés rencontrées et les enseignements que j'en ai tirés.

## Les erreurs initiales et leurs corrections

— Mélange de types de biens : Dans un premier temps, j'avais inclus à la fois les maisons et les appartements dans un seul et même jeu de données. Cela a fortement perturbé les modèles, car les dynamiques de prix entre ces deux types de biens sont très différentes. J'ai donc corrigé cela en me concentrant uniquement sur les appartements, ce qui a immédiatement amélioré la cohérence des prédictions.

— Espace géographique trop large dès le départ : J'avais initialement tenté de modéliser les prix sur l'ensemble de l'Île-de-France. Cependant, la taille et l'hétérogénéité de cette zone rendaient l'analyse très complexe, et les performances étaient médiocres. J'ai alors opté pour une approche plus progressive en me focalisant d'abord sur une ville restreinte (Colombes), puis en élargissant le périmètre à l'échelle du département (Hauts-de-Seine) lorsque j'étais plus à l'aise avec les données.

— Non-traitement des valeurs aberrantes : Au départ, j'avais choisi de conserver toutes les valeurs, même extrêmes, en supposant que les écarts de prix reflétaient simplement la diversité des biens immobiliers. Toutefois, certaines valeurs étaient manifestement aberrantes (prix nuls, valeurs extrêmement élevées pour de très petites surfaces, etc.) et perturbaient fortement l'apprentissage. Après les avoir identifiées et filtrées, les performances des modèles se sont nettement améliorées.

## Pistes explorées non retenues

— Ajout de l'année de construction du bâtiment via la BDNB : J'ai tenté d'enrichir le jeu de données avec des informations issues de la base de données nationale des bâtiments (BDNB), notamment pour intégrer une variable sur l'année de construction. Malheureusement, cette base contenait un grand nombre de valeurs manquantes pour les logements présents dans mon dataset, rendant cette variable peu exploitable. J'ai donc renoncé à l'inclure dans cette version du projet, même si cela reste une piste prometteuse pour des travaux futurs.

Ces erreurs et ces expérimentations font partie intégrante du processus d'apprentissage. Elles m'ont permis de mieux comprendre l'importance de la qualité des données, de la définition du périmètre d'analyse, et de la nécessité de tester différentes hypothèses, même si elles ne mènent pas toujours à des résultats exploitables. Cette démarche réflexive est, selon moi, essentielle dans tout projet de science des données.

# Bibliographie


[1] T. ALKAN, *INVESTIGATION OF THE EFFECT OF CURRENT VALUE ON REAL ESTATE VALUE USING MACHINE LEARNING ALGORITHMS, PROF. DR. COŞKUN ÖZALP, 2024*

[2] H.T. Sergoyan, G.V. Bezirganyan, *AUTOMATED REAL ESTATE VALUATION WITH MACHINE LEARNING : A CASE STUDY ON APARTMENT SALES IN YEREVAN, Technical University of Munich, 2022*

[3] G.E.C. Golondrino, R.G. Ospina, L.F.M. Sanabria, *Application of machine learning techniques in the prediction of real estate prices in the city of Bogota Colombia , Universidad de Cartagena, 2024*

[4] IBRAHIM, AA ; AYILARA-ADEWALE, OA ; ALABI, AA ; OLUSESI, DA, *Evaluation of Price Prediction of Houses in a Real Estate via Machine Learning , J. Appl. Sci. Environ. Manage., 2025*

[5] Davit Martirosyan, *Scalable Data Collection and Machine Learning for Automated Valuation of Armenian Real Estate , American University of Armenia, 2022*

[6] Ashutosh Srivastava, Dr. Om Prakash Yadav, Harsh wardhan, Akshat Pandey, Faizan Khan Siddiqui, *Revolutionizing Real Estate Price Prediction through Advanced Machine Learning Models : A Comparative Study , Scholar, CSE, Lovely Professional University, India, 2024*

[7] Dieudonné Tchuente Serge Nyawa, *Real estate price estimation in French cities using geocoding and machine learning , Annals of Operations Research, 2021*

[8] AYTEN YAĞMUR, MEHMET KAYAKUŞ, MUSTAFA TERZIOĞLU, *House price prediction modeling using machine learning techniques : a comparative study , AESTIMUM, 2022*

[9] Hasan Ahmed Salman, *Random Forest Algorithm Overview , Babylonian Journal of Machine Learning, 2024*

[10] Jung Min Ahn, *Ensemble Machine Learning of Gradient Boosting (XGBoost, LightGBM, CatBoost) and Attention-Based CNN-LSTM for Harmful Algal Blooms Forecasting , Toxins, 2023*

[11] Kalva Sindhu Priya, *Linear Regression Algorithm in Machine Learning through MATLAB , IJRASET, 2021*

[12] Tianqi Chen, Carlos Guestrin, *XGBoost : A Scalable Tree Boosting System , University of Washington, 2016*

[13] Yan-yan SONG, Ying LU, *Decision tree methods : applications for classification and prediction , Shanghai Archives of Psychiatry, 2015*

[14] John T. Hancock and Taghi M. Khoshgoftaar, *CatBoost for big data : an interdisciplinary review , AESTIMUM, 2022*

[15] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2023*

# Webographie

[1] https://fr.wikipedia.org/wiki/Hauts-de-Seine
[2] https://coursenligne.parisnanterre.fr/
[3] https://scikit-learn.org/stable/
[4] https://www.geeksforgeeks.org/ml-linear-regression/
[5] https://www.geeksforgeeks.org/decision-tree-introduction-example/
[6] https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/
[7] https://blent.ai/blog/a/xgboost-tout-comprendre
[8] https://www.data.gouv.fr/fr/datasets/r/087ec735-74fd-48a7-a82e-0b1cd3ea6fe9
[9] https://github.com/Niraiksan/estimation-immo

# Chapitre 8

# Annexes

| valeur_fonciere | longitude | latitude | code_secteur_ville | id_parcelle | adresse_code_voie | lots_surface_logement | nombre_pieces_principales | construction_recente |
|---|---|---|---|---|---|---|---|---|
| 218000.0 | 2.238498 | 48.925608 | 67 | 681 | 67 | 45.16 | 1.0 | 0 |
| 335402.0 | 2.252587 | 48.932724 | 1 | 5 | 1 | 68.0 | 3.0 | 1 |
| 237000.0 | 2.243444 | 48.915559 | 43 | 480 | 43 | 47.33 | 2.0 | 0 |
| 224340.0 | 2.246255 | 48.926389 | 65 | 676 | 65 | 58.91 | 5.0 | 0 |

FIGURE 1 – Extrait dataset encodé - Colombes

| valeur_fonciere | nom_commune | longitude | latitude | code_secteur_ville | id_parcelle | adresse_code_voie | lots_surface_logement | nombre_pieces_principales | construction_recente |
|---|---|---|---|---|---|---|---|---|---|
| 385000.0 | 27 | 2.176749 | 48.877211 | 927 | 13087 | 927 | 72.31 | 2.0 | 0 |
| 351000.0 | 27 | 2.198613 | 48.874248 | 917 | 12833 | 917 | 51.04 | 2.0 | 1 |
| 245000.0 | 26 | 2.235872 | 48.880499 | 904 | 12526 | 904 | 37.71 | 1.0 | 0 |
| 450000.0 | 26 | 2.244781 | 48.883831 | 909 | 12682 | 909 | 63.94 | 3.0 | 0 |

FIGURE 2 – Extrait dataset encodé - Hauts-de-Seine

| Modèle | $R^2$ CV train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.769 ± 0.011 | 0.736 | 48470.13 | 2349353865.23 | 36644.40 | lots_surface_logement<br>Importance : 0.51 | latitude<br>Importance : 0.16 | construction_recente<br>Importance : 0.09 | id_parcelle<br>Importance : 0.06 | adresse_code_voie<br>Importance : 0.06 |
| Arbre de Décision | 0.762 ± 0.013 | 0.750 | 47120.58 | 2220349302.21 | 33806.41 | lots_surface_logement<br>Importance : 0.92 | latitude<br>Importance : 0.05 | longitude<br>Importance : 0.02 | id_parcelle<br>Importance : 0.01 | construction_recente<br>Importance : 0.01 |
| Random Forest | 0.836 ± 0.012 | 0.840 | 37762.63 | 1426015937.70 | 26923.84 | lots_surface_logement<br>Importance : 0.79 | latitude<br>Importance : 0.08 | longitude<br>Importance : 0.06 | id_parcelle<br>Importance : 0.03 | construction_recente<br>Importance : 0.02 |
| XGBoost | 0.850 ± 0.012 | 0.862 | 35004.58 | 1225320540.98 | 24367.60 | lots_surface_logement<br>Importance : 0.60 | construction_recente<br>Importance : 0.12 | latitude<br>Importance : 0.09 | longitude<br>Importance : 0.06 | id_parcelle<br>Importance : 0.05 |
| CatBoost | 0.858 ± 0.009 | 0.870 | 34009.26 | 1156629957.00 | 24154.47 | lots_surface_logement<br>Importance : 0.55 | latitude<br>Importance : 0.12 | longitude<br>Importance : 0.11 | construction_recente<br>Importance : 0.07 | id_parcelle<br>Importance : 0.06 |

FIGURE 3 – Résumé des performances et variables importantes par modèle (Colombes)

| Modèle | $R^2$ CV train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.604 ± 0.003 | 0.600 | 102889.05 | 10586157387.61 | 77339.08 | lots_surface_logement<br>Importance : 0.49 | id_parcelle<br>Importance : 0.19 | adresse_code_voie<br>Importance : 0.06 | code_secteur_ville<br>Importance : 0.06 | nombre_pieces_principales<br>Importance : 0.06 |
| Arbre de Décision | 0.689 ± 0.001 | 0.690 | 90531.46 | 8195944672.31 | 67936.10 | lots_surface_logement<br>Importance : 0.82 | latitude<br>Importance : 0.12 | longitude<br>Importance : 0.04 | construction_recente<br>Importance : 0.01 | nom_commune<br>Importance : 0.00 |
| Random Forest | 0.839 ± 0.003 | 0.841 | 64915.26 | 4213990543.41 | 44201.99 | lots_surface_logement<br>Importance : 0.66 | latitude<br>Importance : 0.16 | longitude<br>Importance : 0.09 | id_parcelle<br>Importance : 0.04 | construction_recente<br>Importance : 0.02 |
| XGBoost | 0.833 ± 0.003 | 0.835 | 66074.50 | 4365838966.13 | 46278.01 | lots_surface_logement<br>Importance : 0.67 | latitude<br>Importance : 0.10 | construction_recente<br>Importance : 0.06 | longitude<br>Importance : 0.05 | nom_commune<br>Importance : 0.05 |
| CatBoost | 0.843 ± 0.004 | 0.845 | 64006.70 | 4096857750.36 | 44581.55 | lots_surface_logement<br>Importance : 0.44 | latitude<br>Importance : 0.29 | longitude<br>Importance : 0.13 | code_secteur_ville<br>Importance : 0.03 | construction_recente<br>Importance : 0.03 |

FIGURE 4 – Résumé des performances et variables importantes par modèle (Hauts-de-Seine)

| Modèle | $R^2$ CV train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.175 ± 0.034 | 0.178 | 938.85 | 881445.01 | 734.30 | latitude<br>Importance : 0.34 | id_parcelle<br>Importance : 0.16 | construction_recente<br>Importance : 0.15 | adresse_code_voie<br>Importance : 0.14 | code_secteur_ville<br>Importance : 0.14 |
| Arbre de Décision | 0.288 ± 0.041 | 0.358 | 829.58 | 688202.28 | 625.50 | latitude<br>Importance : 0.48 | longitude<br>Importance : 0.43 | id_parcelle<br>Importance : 0.08 | code_secteur_ville<br>Importance : 0.01 | construction_recente<br>Importance : 0.00 |
| Random Forest | 0.358 ± 0.064 | 0.430 | 781.47 | 610696.86 | 577.46 | latitude<br>Importance : 0.37 | longitude<br>Importance : 0.36 | id_parcelle<br>Importance : 0.20 | construction_recente<br>Importance : 0.03 | code_secteur_ville<br>Importance : 0.02 |
| XGBoost | 0.380 ± 0.051 | 0.484 | 743.72 | 553114.15 | 548.93 | latitude<br>Importance : 0.26 | longitude<br>Importance : 0.20 | id_parcelle<br>Importance : 0.19 | construction_recente<br>Importance : 0.18 | code_secteur_ville<br>Importance : 0.17 |
| CatBoost | 0.383 ± 0.054 | 0.504 | 729.10 | 531592.96 | 544.59 | latitude<br>Importance : 0.39 | longitude<br>Importance : 0.27 | id_parcelle<br>Importance : 0.14 | construction_recente<br>Importance : 0.11 | adresse_code_voie<br>Importance : 0.05 |

FIGURE 5 – Résumé des performances et variables importantes par modèle (Colombes)

| Modèle | $R^2$ CV train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.020 ± 0.001 | 0.016 | 1842.85 | 3396082.47 | 1460.05 | id_parcelle<br>Importance : 0.43 | code_secteur_ville<br>Importance : 0.14 | adresse_code_voie<br>Importance : 0.14 | nom_commune<br>Importance : 0.11 | latitude<br>Importance : 0.08 |
| Arbre de Décision | 0.412 ± 0.005 | 0.404 | 1434.68 | 2058310.04 | 1082.02 | latitude<br>Importance : 0.64 | longitude<br>Importance : 0.28 | construction_recente<br>Importance : 0.05 | code_secteur_ville<br>Importance : 0.01 | id_parcelle<br>Importance : 0.01 |
| Random Forest | 0.602 ± 0.004 | 0.604 | 1168.66 | 1365760.14 | 833.49 | latitude<br>Importance : 0.48 | longitude<br>Importance : 0.28 | id_parcelle<br>Importance : 0.15 | construction_recente<br>Importance : 0.05 | code_secteur_ville<br>Importance : 0.02 |
| XGBoost | 0.580 ± 0.004 | 0.577 | 1208.65 | 1460830.90 | 890.86 | construction_recente<br>Importance : 0.26 | latitude<br>Importance : 0.26 | nom_commune<br>Importance : 0.22 | longitude<br>Importance : 0.12 | id_parcelle<br>Importance : 0.07 |
| CatBoost | 0.591 ± 0.004 | 0.591 | 1188.95 | 1413606.96 | 870.96 | latitude<br>Importance : 0.49 | longitude<br>Importance : 0.26 | id_parcelle<br>Importance : 0.08 | construction_recente<br>Importance : 0.05 | code_secteur_ville<br>Importance : 0.05 |

FIGURE 6 – Résumé des performances et variables importantes par modèle (Hauts-de-Seine)

| Modèle | $R^2$ CV train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.289 ± 0.043 | 0.363 | 826.57 | 683210.14 | 649.81 | id_parcelle<br>Importance : 0.22 | latitude<br>Importance : 0.21 | construction_recente<br>Importance : 0.14 | type_appart<br>Importance : 0.11 | centre_ville_distance<br>Importance : 0.10 |
| Arbre de Décision | 0.334 ± 0.046 | 0.467 | 756.20 | 571834.10 | 572.91 | latitude<br>Importance : 0.31 | type_appart<br>Importance : 0.28 | centre_ville_distance<br>Importance : 0.15 | construction_recente<br>Importance : 0.10 | id_parcelle<br>Importance : 0.07 |
| Random Forest | 0.428 ± 0.055 | 0.545 | 698.73 | 488219.42 | 506.41 | latitude<br>Importance : 0.23 | type_appart<br>Importance : 0.17 | centre_ville_distance<br>Importance : 0.16 | longitude<br>Importance : 0.14 | id_parcelle<br>Importance : 0.13 |
| XGBoost | 0.467 ± 0.055 | 0.615 | 642.92 | 413344.61 | 476.25 | code_secteur_ville<br>Importance : 0.15 | latitude<br>Importance : 0.11 | construction_recente<br>Importance : 0.11 | centrecommercial_proximite<br>Importance : 0.11 | centre_ville_distance<br>Importance : 0.10 |
| CatBoost | 0.474 ± 0.056 | 0.625 | 633.71 | 401588.08 | 471.58 | latitude<br>Importance : 0.20 | type_appart<br>Importance : 0.17 | centre_ville_distance<br>Importance : 0.15 | longitude<br>Importance : 0.12 | construction_recente<br>Importance : 0.12 |

FIGURE 7 – Résumé des performances et variables importantes par modèle (Colombes)

| Modèle | $R^2$ CV train | $R^2$ test | RMSE test | MSE test | MAE test | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| Régression Linéaire | 0.261 ± 0.008 | 0.276 | 1571.10 | 2468364.37 | 1231.07 | id_parcelle<br>Importance : 0.18 | nb_supermarket_500m<br>Importance : 0.16 | type_appart<br>Importance : 0.09 | construction_recente<br>Importance : 0.09 | code_secteur_ville<br>Importance : 0.08 |
| Arbre de Décision | 0.402 ± 0.005 | 0.413 | 1414.93 | 2002036.85 | 1089.92 | latitude<br>Importance : 0.42 | nb_supermarket_500m<br>Importance : 0.29 | longitude<br>Importance : 0.11 | centre_ville_distance<br>Importance : 0.07 | construction_recente<br>Importance : 0.06 |
| Random Forest | 0.613 ± 0.008 | 0.648 | 1095.26 | 1199595.95 | 775.58 | latitude<br>Importance : 0.30 | nb_supermarket_500m<br>Importance : 0.15 | longitude<br>Importance : 0.14 | id_parcelle<br>Importance : 0.11 | centre_ville_distance<br>Importance : 0.10 |
| XGBoost | 0.601 ± 0.006 | 0.618 | 1141.02 | 1301929.61 | 832.89 | nb_supermarket_500m<br>Importance : 0.30 | latitude<br>Importance : 0.15 | construction_recente<br>Importance : 0.13 | nom_commune<br>Importance : 0.08 | code_secteur_ville<br>Importance : 0.06 |
| CatBoost | 0.622 ± 0.007 | 0.647 | 1096.41 | 1202110.72 | 796.82 | latitude<br>Importance : 0.38 | longitude<br>Importance : 0.16 | nb_supermarket_500m<br>Importance : 0.07 | centre_ville_distance<br>Importance : 0.07 | construction_recente<br>Importance : 0.05 |

FIGURE 8 – Résumé des performances et variables importantes par modèle (Hauts-de-Seine)

| prix_m2 | longitude | latitude | code_secteur_ville | id_parcelle | adresse_code_voie | construction_recente | type_appart | centre_ville_distance | nb_ecole_proximite_500m | nb_gare_proximite_500m | nb_supermarket_500m | nb_centrecommercial_proximite_500m | nb_hospital_500m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4827.280779450842 | 2.238498 | 48.925608 | 92025000BX | 92025000BX0193 | 4628 | 0 | T3 | 1.2173372740953 87 | 6 | 0 | 1 | 0 | 1 |
| 4932.382352941177 | 2.252587 | 48.932724 | 92025000DB | 92025000DB0252 | 10 | 1 | T4 | 1.1427448468794288 | 3 | 0 | 0 | 0 | 0 |
| 5007.394886963871 | 2.243444 | 48.915559 | 92025000AT | 92025000AT0227 | 5220 | 0 | T3 | 1.1172640747290452 | 6 | 0 | 0 | 0 | 0 |
| 3808.181972500424 8 | 2.246255 | 48.926389 | 92025000BS | 92025000BS0012 | 7165 | 0 | T3 | 0.7378413689384361 | 11 | 0 | 2 | 0 | 0 |
| 5578.598092077976 | 2.233084 | 48.910817 | 92025000CF | 92025000CF0207 | 135 | 0 | T3 | 2.0354224231775509 | 3 | 0 | 3 | 1 | 0 |
| 5620.042262678803 | 2.233542 | 48.91039 | 92025000CF | 92025000CF0198 | 1564 | 0 | T3 | 2.0408069790617 09 | 5 | 1 | 4 | 1 | 0 |
| 4386.509414442376 | 2.246603 | 48.923661 | 92025000BK | 92025000BK0158 | 65 | 0 | T3 | 0.5878122929267999 | 8 | 0 | 3 | 0 | 0 |

FIGURE 9 – Extrait dataset enrichie - Colombes

| prix_m2 | nom_commune | longitude | latitude | code_secteur_ville | id_parcelle | adresse_code_voie | construction_recente | type_appart | centre_ville_distance | nb_ecole_proximite_500m | nb_gare_proximite_500m | nb_supermarket_500m | nb_centrecommercial_proximite_500m | nb_hospital_500m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4210.836277974087 | Clichy | 2.305108 | 48.903617 | 920240000K | 920240000K0192 | 6140 | | 0 T1 | 0.09082226718894726 | 13 | 1 | 10 | 0 | 1 |
| 8250.0 | Clichy | 2.311397 | 48.901329 | 920240000Z | 920240000Z0088 | 3560 | | 1 T4 | 0.5006264447770217 | 9 | 1 | 11 | 0 | 2 |
| 7077.435470441299 | Clichy | 2.30897 | 48.906286 | 920240000J | 920240000J0031 | 4122 | | 0 T2 | 0.4874570184584975 | 10 | 1 | 8 | 0 | 2 |
| 4436.307525396982 | Clichy | 2.306151 | 48.896005 | 920240004F | 920240004F0022 | 1411 | | 0 T6+ | 0.7621126732340674 | 1 | 0 | 5 | 0 | 0 |
| 6596.8674219837 | Clichy | 2.306581 | 48.900427 | 920240000R | 920240000R0096 | 4420 | | 0 T3 | 0.2916379934695225 | 10 | 1 | 11 | 0 | 1 |
| 8068.181818181818 | Clichy | 2.300614 | 48.90294 | 920240000L | 920240000L0019 | 3525 | | 1 T2 | 0.3159998290322629 | 8 | 1 | 9 | 0 | 0 |
| 7377.777777777777 | Clichy | 2.312447 | 48.899447 | 920240000Y | 920240000Y0017 | 1490 | | 0 T2 | 0.6666481774461893 | 4 | 0 | 8 | 0 | 2 |
| 2680.834898025251 | Clichy | 2.301019 | 48.902312 | 920240000L | 920240000L0088 | 3535 | | 0 T2 | 0.2912864874895379 | 8 | 1 | 9 | 0 | 0 |

FIGURE 10 – Extrait dataset enrichie - Hauts-de-Seine