Rapport de projet - Prédiction des prix immobiliers

Table des matières

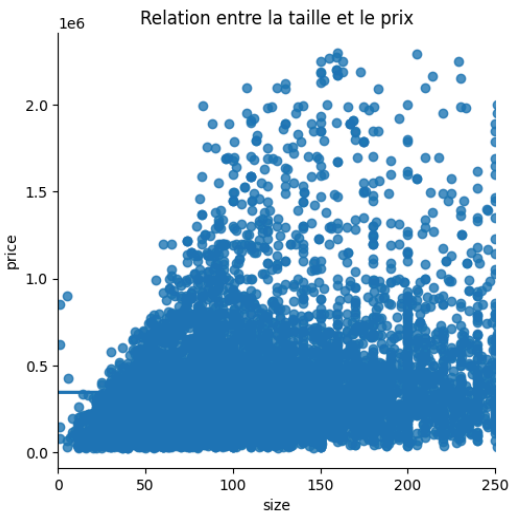
1. Introduction

Ce projet porte sur un problème de prédiction des prix de biens immobiliers en France à partir d'un jeu de données contenant 27 variables, incluant le type de propriété, la surface, et la localisation géographique (latitude/longitude). L'objectif était de prédire au mieux le prix d'un bien immobilier à partir de ces données, tout en mettant en œuvre les compétences acquises pendant le semestre.

Ce projet constituait notre première expérience de modélisation sur des données réelles et brutes, ce qui nous a amenés à expérimenter différents modèles, techniques de traitement et choix méthodologiques.

2. Première approche : modèle de base

Nous avons tout d'abord pris le temps d'explorer et de comprendre les données grâce à des méthodes telles que .describe(), .info() et l'analyse des valeurs manquantes.



Pour une première approche, nous avons construit un modèle de régression linéaire très simple, utilisant uniquement trois variables: la taille de la surface habitable, la latitude et la longitude, qui nous semblaient être parmi les plus importantes pour prédire le prix d'un bien. Cela nous a permis de nous familiariser avec la structure des données et de poser un point de clair.

Les valeurs manquantes de la surface habitable ont été remplacées par 0 signifiant que les biens ne possédaient pas de surface habitable. C'est une supposition que nous avons faite après avoir regardé plus précisément nos données. Le modèle a atteint un MAPE (Mean Absolute Percentage Error) d'environ 0,5855, ce qui nous a paru satisfaisant pour une première version.

3. Pré-traitement des données

Une fois le modèle de base testé, nous avons entamé un travail beaucoup plus approfondi de préparation et de sélection des données, dans l'objectif d'améliorer les performances de notre modèle final.

3.1 Traitement des valeurs manquantes

Pour les variables nombre de chambres, de salles, de salles de bain et la taille de la surface habitable, nous avons utilisé un Iterative Imputer avec un Random Forest Regressor. Ce choix vient du fait que certaines de ces variables sont assez liées : par exemple, le nombre de chambres et de salles de bain sont très fortement corrélés (0,87 : calculé par la valeur de la corrélation de Pearson, de formule : covariance de X et Y divisé par la multiplication de l'écart type de X par celui de Y et représentant le degré de relation linéaire entre deux variables quantitatives), et les salles de bain montrent également une corrélation modérée avec ces deux variables (~0,42).

Même si la surface habitable est moins corrélée avec le reste (entre -0,12 et -0,03), nous avons décidé de la garder car un modèle comme le Random Forest peut apprendre des relations plus complexes que ce que la simple corrélation montre. Nous avons également vérifié qu'il y avait un nombre suffisant de valeurs non manquantes pour entraîner efficacement l'imputeur. Ce modèle nous a finalement permis de gagner environ un point sur notre score de prédiction, comparé à une imputation plus simple par la médiane.

D'autres variables (le nombre d'émissions de gaz à effet de serre, et la consommation d'énergie du bien) ont été imputées par la médiane, parce que près de 50% des valeurs étaient manquantes, donc notre prédiction aurait été coûteuse sans pour autant améliorer notre modèle.

Nous avons remplacé la variable de l'étage par 0, après avoir vérifié qu'une absence d'information pouvait logiquement correspondre à l'absence d'étage, et la taille du terrain par la valeur de la surface habitable, car après avoir étudié les biens qui comportaient des valeurs nulles pour la taille du terrain, nous avons supposé que cela signifiait qu'ils n'avaient pas de terrain supplémentaire.

Nous avons également supprimé la variable d'exposition, car il y avait 75% de données manquantes. La conserver aurait nécessité une imputation lourde ou aurait ajouté beaucoup d'incertitude au modèle. Cela aurait pu nuire à la qualité des prédictions en introduisant du bruit, et augmenter le risque d'overfitting, notamment si le modèle apprenait des schémas erronés à partir de valeurs imputées peu fiables.

3.2 Encodage des variables catégorielles

Pour le type de propriété qui est une variable catégorielle nominale sans ordre particulier, nous avons choisi d'utiliser un OneHotEncoding. Cette méthode permet de représenter chaque catégorie par un 0 ou 1, évitant donc d'introduire une relation d'ordre artificielle entre les catégories, ce qui est important pour que le modèle apprenne correctement les différences entre types de propriétés.

En revanche, pour les variables ghg_category et energy_performance_category, qui représentent respectivement la performance des émissions de gaz à effet de serre et la performance énergétique selon des échelles ordonnées définies par la réglementation française, nous avons opté pour un Label Encoding Ordonné. Ce choix permet de conserver l'information d'ordre naturel des catégories, ce qui aide le modèle à comprendre la progression des performances environnementales sans créer de variables supplémentaires. Nous avons traité les valeurs manquantes du nombre d'émissions de gaz à effet de serre, comme une catégorie à part entière, en les remplaçant par la chaîne "nan", afin que le modèle les considère explicitement.

3.3 Variables non retenues

Nous avons choisi de supprimer certaines variables de notre jeu de données car elles posaient des problèmes techniques ou apportaient peu d'informations utiles pour la prédiction. La variable de la ville a été retirée car elle contient un très grand nombre de modalités, difficiles à encoder de manière fiable. En plus, plus de 200 villes présentes dans le jeu de test n'étaient pas dans les données d'entraînement, ce qui pose un problème de généralisation.

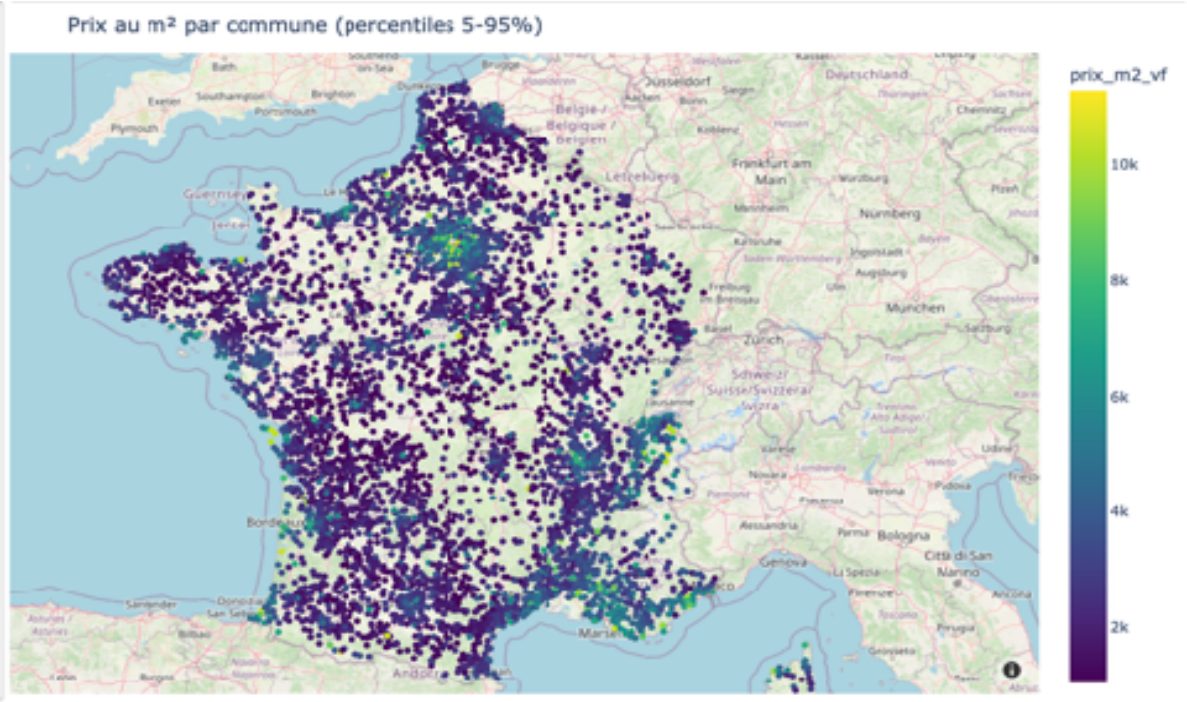
Comme nous avions déjà ajouté une variable « prix_m2_vf » (le prix moyen au m² par commune), qui

résume efficacement l'information géographique et économique locale, il n'était plus nécessaire de garder la ville. Cela nous a aussi évité un encodage complexe.

De la même manière, la variable du code postal a été supprimée pour ne pas induire le modèle en erreur : en tant que donnée numérique, elle aurait pu être interprétée comme une valeur continue alors qu'elle ne l'est pas. À la place, nous avons créé une nouvelle variable indiquant la fréquence d'apparition de chaque code postal, ce qui nous donne une idée de l'activité immobilière dans chaque zone. Les endroits avec beaucoup de ventes sont souvent des zones où les prix sont plus élevés.
D'autres variables ont aussi été supprimées, comme l'exposition, car elle contenait plus de 75% de valeurs manquantes, rendant son exploitation peu fiable.

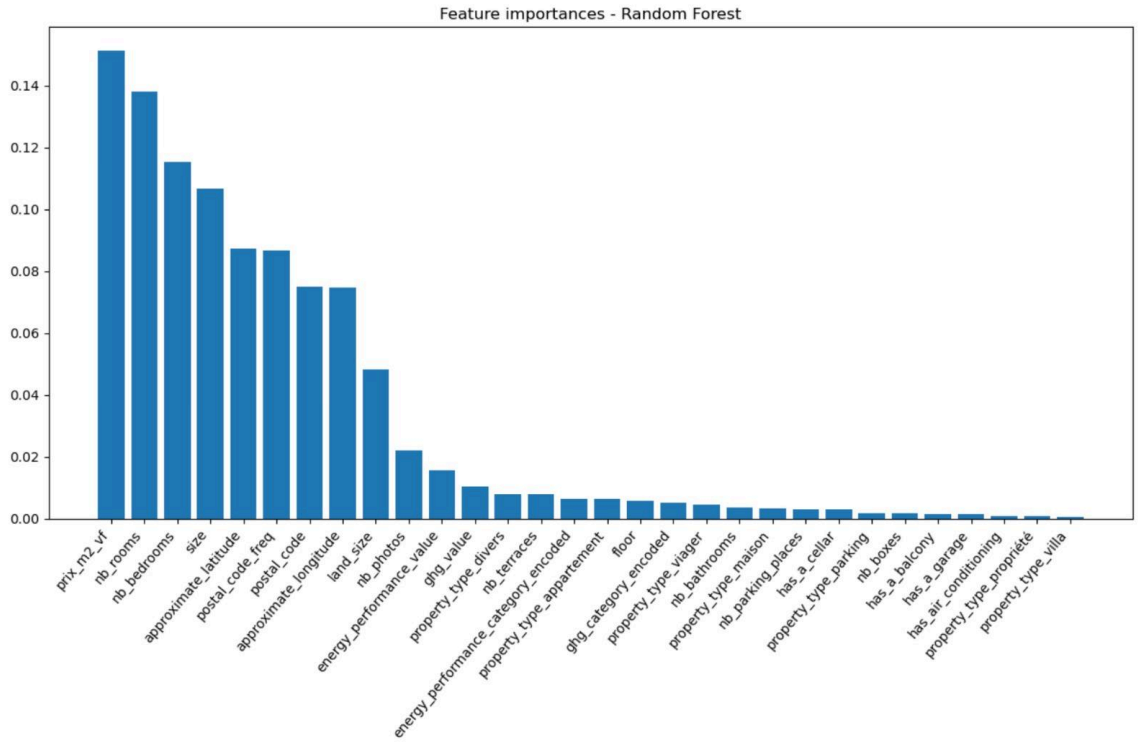
4. Enrichissement par des données externes

Nous avons décidé d'intégrer les valeurs foncières de 2020, 2021 et 2022, issues de data.gouv.fr. Ces données regroupent toutes les transactions immobilières enregistrées (prix, adresse, surface, type de bien, etc.).



Nous avons utilisé ces informations pour calculer le prix moyen au m² par commune, et ajouté cette variable comme nouvelle feature dans notre dataset. Cela a permis de fournir au modèle une connaissance contextuelle économique du marché local, souvent absente dans les seules variables de localisation. L'ajout de la variable du prix/ m2/ commune s'est révélé particulièrement pertinent, puisqu'elle ressort comme la variable la plus importante dans la mesure d'importance des features produite par le modèle Random Forest .

Un score de feature importance de 0,15 pour la variable prix_m2_vf signifie que cette variable a contribué à environ 15 % de la réduction totale de l'erreur (de la variance) dans les arbres de la forêt, ce qui en fait un facteur prédictif majeur du modèle Random Forest utilisé.



5. Modélisation et sélection du modèle

Dans notre première approche nous avons utilisé une régression linéaire avec peu de features, puis au fur et à mesure nous avons complexifié notre modèle. Dans une seconde approche, nous avons testé plusieurs modèles avec la quasi-totalité des features que nous avions commencé à modifier. Pour cela, nous avons effectué une cross-validation 5-fold avec la métrique MAPE, afin de mesurer l'erreur relative moyenne sans biais dû au découpage du jeu de données.

5.1 Résultats des modèles de base

Modèle	MAPE avec CV
LinearRegression	0.820274
Ridge	0.820757
Lasso	0.820291
ElasticNet	0.898066
DecisionTreeRegressor	0.505081
RandomForestRegressor	0.401341

Nous observons que les modèles linéaires sous-performent largement par rapport aux modèles à base d'arbres. Cela s'explique par le fait que de nombreuses variables sont catégorielles, transformées par OneHotEncoding : les relations entre features et prix ne sont pas linéaires, ce qui limite l'efficacité de modèles comme la régression linéaire ou Ridge.

Nous avons donc décidé de concentrer la suite du travail sur des modèles non linéaires, et en particulier sur le Random Forest Regressor, qui offrait le meilleur compromis entre performance et robustesse.

Finalement notre modèle de Random Forest final avait une profondeur maximale de 45 et minimale de 34,

et notre forêt comptait 100 arbres.

6. Améliorations

6.1 Passage au logarithme de la variable cible

Une transformation qui a fortement amélioré les performances est le passage à l'échelle logarithmique pour le prix (log(price)). Les prix immobiliers étant très dispersés, le logarithme réduit l'influence des valeurs extrêmes et rend la relation entre les variables explicatives et la cible plus linéaire et plus stable. Cette transformation facilite l'apprentissage, même pour certains modèles non linéaires.

7. Difficultés rencontrées

Nous avons été confrontés à plusieurs difficultés techniques et méthodologiques. La gestion de la mémoire sur Google Colab a posé problème, certaines opérations comme l'ajout des données foncières ou l'imputation avec RandomForest provoquaient de nombreux crashs (déconnexion du runtime ou dépassement de mémoire). Ceci nous a complexifié la tâche car chaque petit changement de code nous a pris beaucoup de temps.

L'encodage des villes s'est révélé problématique. L'utilisation initiale d'Ordinal Encoder pour la ville a échoué, car plus de 200 villes présentes dans le test n'existaient pas dans l'entraînement. Cela nous a surpris car nous n'avions pas pensé à cette possibilité, nous avons donc supprimé cette feature. Mais cela n'a pas dégradé notre programme car avec l'ajout de la feature "prix_m2_commune" nous avions déjà une donnée propre à la ville.

Enfin, lors de certaines phases de test, notre modèle a renvoyé des erreurs liées à des valeurs “infinies” dans les prédictions. Cela bloquait l’exécution de notre pipeline, avec des messages d’erreur difficiles à tracer. Pour comprendre l’origine de ce problème, nous avons d’abord vérifié l’absence de NaN dans les variables d’entrée du modèle. Ensuite, nous avons inspecté les outliers (valeurs extrêmes), car certains points avec des valeurs très élevées (notamment sur la cible ou certaines features comme la taille de propriété) pouvaient perturber l'apprentissage. C’est pour cela qu’on a passé la variable du prix à l’échelle logarithmique pour réduire l’influence des valeurs extrêmes.

8. Pistes d'amélioration

Plusieurs pistes pourraient être explorées pour améliorer davantage les performances du modèle. Une première amélioration consisterait à exploiter les données d’images fournies dans le challenge, en extrayant par exemple des informations comme la luminosité ou l’état apparent du bien, afin d’ajouter une dimension visuelle aux données tabulaires. Par ailleurs, l'utilisation de modèles plus performants comme XGBoost, pourrait permettre d’obtenir de meilleurs résultats que ceux que nous avons obtenus avec les forêts aléatoires.

9. Conclusion

Ce projet nous a permis d'appliquer concrètement les notions que nous avons vues en cours, et de nous confronter à des problématiques réelles de data science : qualité des données, choix du modèle, coût de la complexité, performances mesurables.

Nous avons progressivement construit un pipeline robuste, combinant imputation intelligente, enrichissement externe et sélection de modèle efficace. Le modèle final, basé sur une Random Forest enrichie des prix au m² par commune, atteint un MAPE d'environ 0.32, ce qui constitue une performance très satisfaisante étant donné la nature du problème et le modèle utilisé.