SPAC Evelina
TSANGALI Eleni
BRISSON Capucine

Mai 2025

Rapport du Data Challenge de l'Institut Louis Bachelier

Dans ce rapport, nous présenterons notre approche au développement d'un modèle de prédiction des prix de l'immobilier, dans un contexte de participation au Data Challenge lancé par l'Institut Louis Bachelier.

Valeur finale de la prédiction : 28,439666

Modèle **XGBoost** (colsample_bytree = 1, learning_rate = 0.1, max_depth = 10, n_estimators = 200, subsample = 0.8)

I- Introduction

L'Institut a mis en place un challenge de prédictions du prix de biens immobiliers en France. Ainsi, tout participant dispose d'un jeu de données issu d'une technique de pricing usuelle appelée la *méthode hédonique* qui considère que le prix d'un bien dépend de certaines caractéristiques jugées essentielles. Les données généralement utilisées pour estimer le prix des maisons ne sont pas considérées comme suffisantes, c'est pour ces raisons que des photos sont ajoutées à la disposition des participants pour observer si ces dernières apportent des informations complémentaires à la prédiction. Le Data Challenge est composé de 50 000 offres immobilières divisées en données d'entraînement (40 000) et de test (10 000). Rappelons que les prix dans le dataset sont les prix des offres immobilières et non pas les prix des transactions. Nous y trouvons 26 variables comme le type de biens, le nombre de pièces, la localisation (qui a été bruitée pour anonymiser les données), etc.

Une métrique est un critère d'évaluation utilisé pour mesurer les performances d'un modèle de machine learning. Elle permet de quantifier à quel point les prédictions du modèle sont proches de la réalité. Dans le contexte du challenge, la métrique utilisée est la MAPE : Mean Pourcentage Absolute Error. On cherche à prédire le prix du bien à plus au moins x% de la valeur réelle. Ainsi, plus le MAPE est faible, meilleur est le modèle. Une seconde information fortement utile est le benchmark ou modèle de référence. Ici, le benchmark est un XGBoost qui prédit un score de 36,78% (+-11). Cela signifie que le modèle de base en moyenne à 36,78% d'erreur relative sur les prédictions avec une certaine variabilité de 11 selon le split de validations.

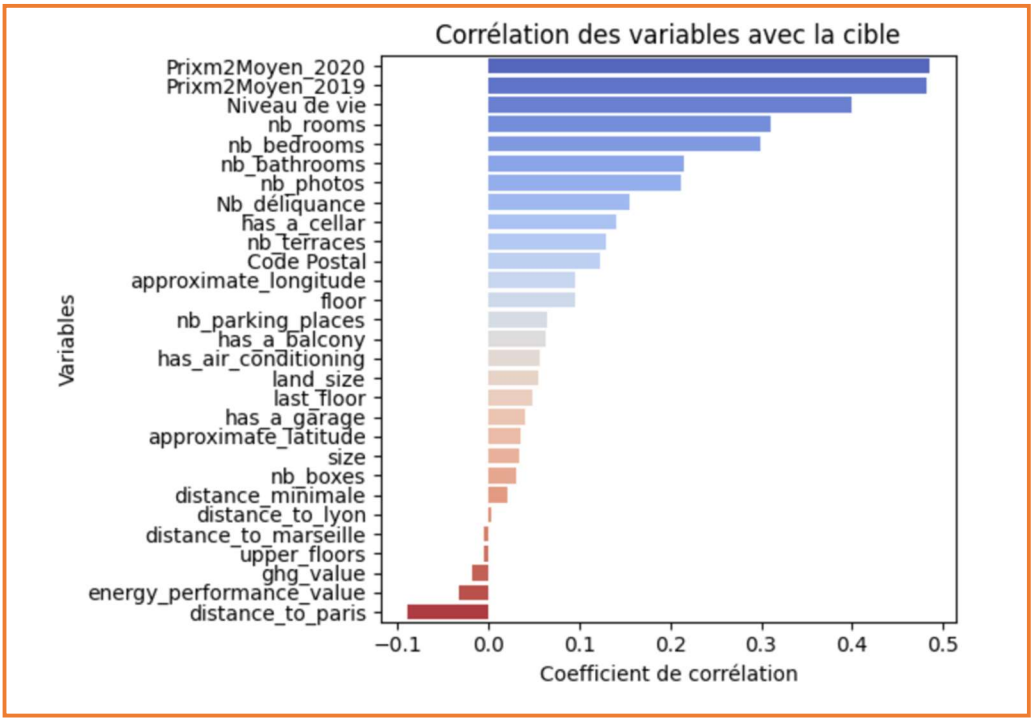
Ce challenge fournit une vraie motivation : l'immobilier est un secteur économique majeur. Prédire les prix de l'immobilier peut être considéré comme une tâche complexe. Néanmoins, la prédiction de prix est directement liée à la décision d'investissement. De ce fait, savoir estimer les prix revient à répondre à des enjeux tant financiers que quotidien. Le challenge permet de comprendre les facteurs qui influencent les prix et les dynamiques du marché immobilier. De plus, à l'issu de ce projet, nous avons pu développer un modèle qui pourrait être utile à des particuliers pour prédire facilement le prix de leur bien immobilier. Même si les résultats obtenus ne sont pas exacts, notre modèle permet d'avoir une bonne estimation d'un bien avec seulement quelques critères faciles à déterminer.

II- Présentation des données additionnelles, choix et motivations

Pour ce Data Challenge nous avons laissé libre cours à notre imagination et avons décidé de rajouter des données au dataset initial pour contribuer à l'amélioration de notre score. Nous avons réfléchi à des variables additionnelles qui peuvent jouer un rôle crucial dans la prédiction immobilière. Nous avons donc pensé au *prix du mètre carré* et *au revenu moyen par commune* mais aussi *au niveau d'insécurité*.

En effet, concernant le revenu, si le pouvoir d'achat dans une ville est plus élevé alors on pourrait observer des prix plus élevés. Tout comme pour le niveau d'insécurité, si la délinquance est forte au sein de la ville alors moins de personnes souhaiteraient s'y installer ce qui pourrait expliquer des prix plus bas dans certaines localisations.

A la suite de l'ajout des données mentionnées, nous pouvons observer une corrélation positive entre le prix des biens et le prix moyen du m2 par commune, le niveau de vie et de délinquance. À l'inverse, la distance à Paris présente une corrélation négative, ce qui signifie que plus un bien est éloigné de Paris, plus sa valeur tend à diminuer.



Ainsi, le graphique ci-dessus confirme la pertinence de ces nouvelles colonnes dans la prédiction des prix de l'immobilier que nous avons donc décidé de garder pour l'entrainement futur de notre modèle de prédiction. Au contraire, nous avons testé certains ajouts de données qui nous semblaient pertinents au premier abord, mais qui ne le furent pas.

Notamment, un facteur qui nous a semblé important était la localisation, c'est pour cela que nous avons voulu ajouter pour chaque bien la distance au centre-ville de la commune, mais les résultats n'étant pas convaincants. Nous avons supposé que le bruit ajouté à la localisation exacte des biens ait pu erroné les résultats. Ainsi, après réflexion, nous avons décidé qu'il serait plus intéressant de comparer la distance aux plus grandes villes de France. La réflexion derrière est la suivante : un bien situé proche de grandes villes peut apporter de nombreux avantages

pour la mobilité professionnelle (travail, aéroports, etc.) et l'accès aux services de premières nécessités. Nous avons donc ajouté la distance minimale à l'une des plus grandes villes, ainsi que la distance à Paris, Lyon et Marseille.

III- Préparation des données

En explorant le dataset, nous avons rapidement remarqué qu'un nombre significatif de variables comportaient des valeurs manquantes. Ces données incomplètes peuvent poser un problème pour l'entraînement des modèles, qui nécessitent généralement des entrées numériques complètes. Nous avons donc effectué une série de transformations afin de les rendre exploitables.

Nous avons commencé par identifier le type de chaque variable afin d'appliquer un traitement adapté : numériques ou catégorielles. Les valeurs numériques peuvent prendre des valeurs continues ou discrètes, tandis que les catégorielles (comme le type de propriété, la performance énergétique ou l'exposition) peuvent être ordinales ou simplement nominales (c'est à dire qu'elles n'ont pas d'ordre logique entre elles).

Pour le traitement des données numériques manquantes, nous avons choisi d'imputer la médiane de la variable correspondante. Cette méthode a l'avantage d'être plus robuste que la moyenne face aux valeurs extrêmes et donc de mieux représenter la tendance centrale dans des distributions asymétriques ou bruitées.

Les modèles de Machine Learning classiques ne peuvent pas traiter directement des variables catégorielles. Nous avons donc utilisé le OneHotEncoder, qui transforme chaque catégorie d'une variable catégorielle en une nouvelle colonne binaire (0 ou 1). Cette méthode permet d'éviter d'introduire une fausse hiérarchie entre les catégories. Nous avons également testé l'Ordinal Encoder, mais il attribue des valeurs numériques entières aux catégories, ce qui peut induire le modèle en erreur en lui faisant croire qu'il existe un ordre entre elles, ce qui n'est pas pertinent dans le cas de variables purement nominales, comme le "type de propriété" par exemple.

Nous avons ensuite évalué la corrélation entre les variables et la variable cible afin de détecter les éventuelles redondances ou variables peu informatives. Toutefois, nous avons décidé de conserver l'ensemble des variables, car une faible corrélation linéaire n'exclut pas nécessairement l'utilité d'une variable dans des modèles non linéaires, comme XGBoost ou les DecisionTrees, qui sont capables de capter des interactions plus complexes entre les variables.

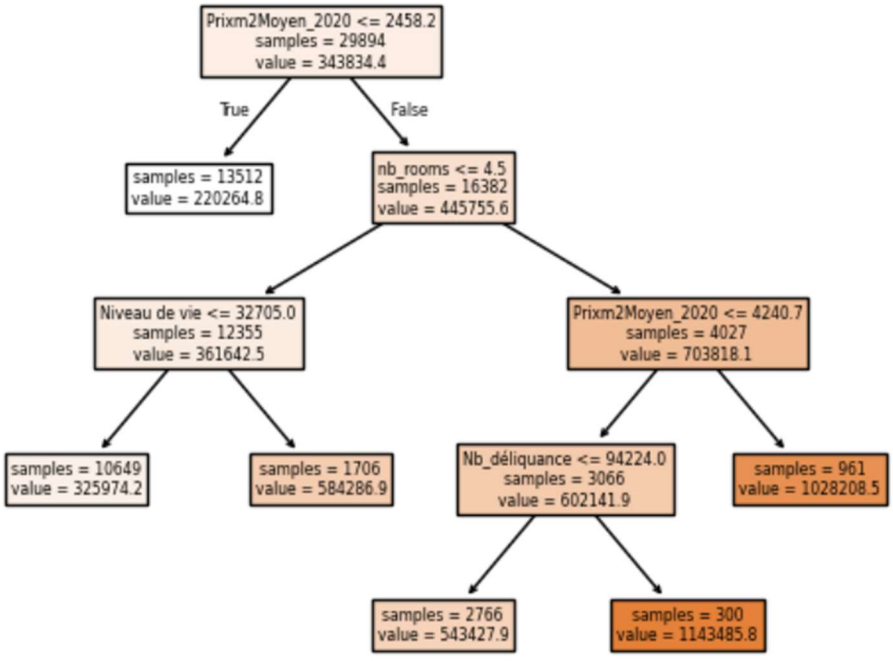
Avant de soumettre nos prédictions finales, nous avons effectué un split 80/20 du jeu de données d'entraînement, afin d'avoir une première évaluation de notre modèle. Cette séparation nous a permis de tester différentes approches, de mesurer les performances et d'ajuster les hyperparamètres. Pour la phase finale de soumission, le modèle a été réentraîné sur l'ensemble du jeu de données d'entraînement.

IV- Sélection et entrainement du modèle

Après avoir sélectionné les données et traitées ces dernières, nous avons dû entraîner notre modèle. Pour ce faire, nous avons commencé par tester les différents modèles évoqués lors de ce cours d'introduction à l'apprentissage statistique. Les scores que nous avons obtenus sur ces modèles n'étaient pas satisfaisant en comparaison à celui du Benchmark.

Modèle utilisé	Paramètres	Score MAPE
KNeighborsRegressor	n_neighbors = 3	0.564843
LinearRegression	/	0.596957
DecisionTreeRegressor	max_leaf_nodes = 6 random_state = 0	0.826138
RandomForestRegressor	n_estimators = 300 max_depth = 50	0.319805

Par exemple, un modèle comme un arbre de décisions, à de nombreux avantages : c'est un modèle très explicite qui permet de comprendre facilement comment sont obtenues les prédictions en suivant le chemin de la racine de l'arbre jusqu'aux feuilles. Cependant, la qualité des précisions du modèle a ses limites.



Nous avons donc voulu entrainer un modèle de forêts aléatoires : ce type de modèle permet de réduire la variance et donc l'overfitting d'un arbre de décisions classique pour obtenir de meilleures prédictions. La forêt aléatoire est l'un des modèles qui semblait le plus adapté à notre challenge : le modèle crée un ensemble d'arbres indépendants de différentes profondeurs sur un sous-ensemble du dataset d'entrainement. Ainsi, chaque arbre de la forêt est implémenté sur un Bootstrap issu d'un processus de tirage indépendant et aléatoire. L'arbre n'est donc implémenté que sur une sous-partie des colonnes du dataset.

Les arbres de la forêt sont ensuite agrégés : la prédiction finale du modèle est la moyenne de la valeur prédite par chaque arbre. La forêt aléatoire est une amélioration de l'arbre de décisions qui permet d'ajouter du biais et décorréler les différents arbres pour obtenir de meilleures prédictions.

Grâce aux nouvelles données que nous avons ajouté et au modèle de forêt aléatoire, nous avons pu obtenir des résultats très satisfaisant. En cherchant à améliorer encore nos prédictions, nous avons vu que le modèle XGBoost était souvent utilisé pour les challenges Kaggle, de plus, c'est le modèle qui est utilisé par le benchmark.

En effet, le modèle XGBoost reprend le principe d'une forêt aléatoire, mais à la différence du modèle précédent, les arbres ne sont pas indépendants : les arbres sont créés un par un pour corriger les erreurs des précédents. Nous parlons de construction séquentielle des arbres. Chaque nouvel arbre ne cherche plus à prédire directement le prix du bien mais une valeur résiduelle qui correspond à l'erreur entre la prédiction de l'arbre précédent et la vraie valeur du bien immobilier. Chaque arbre est entrainé pour prédire l'erreur résiduelle du précédent. Cette méthode permet de gagner en précision et réduire l'erreur liée à la construction de chaque nouvel arbre séquentiellement. Finalement, après chaque arbre, la prédiction est mise à jour avec la formule suivante :

*Nouvelle prédiction = prédiction de l'arbre précédent + learning rate * prédiction de l'arbre*

Une fois le modèle final sélectionné, il ne restait plus qu'à chercher les hyperparamètres optimaux pour notre modèle XGBoost. Pour ce faire, nous avons utilisé la fonction GridSearch qui effectue de manière automatique une cross-validation. La cross-validation a pour objectif de tester plusieurs combinaisons d'hyperparamètres pour sélectionner la combinaison la plus performante.

V- Conclusion

En conclusion, notre participation au Data Challenge de l'Institut Louis Bachelier a constitué à prédire le prix de l'immobilier en jouant avec les variables à notre disposition ainsi qu'avec d'autres données ajoutées. Ce travail nous a permis d'appliquer des outils vus en cours concernant la prédiction de données et à aboutir à un score en dessous du benchmark. Nous pouvons, tout de même, nous questionner sur d'éventuelles améliorations et modifications. Nous pensons à la prise en compte des photos dans le modèle. En effet, une visualisation des biens immobilier permettrait une meilleure estimation du fait de la perception de la qualité et de l'état du bien. Pour finir, nous souhaitions mentionner l'importance du traitement des données. Avant de réfléchir à quelles autres variables nous pouvions ajouter au modèle, un travail provisoire de traitement des données est essentiel. Appliquer le bon modèle de prédiction et trier correctement les données contribue fortement à améliorer les prédictions.

Sources des données additionnelles :

https://www.data.gouv.fr/fr/datasets/indicateurs-immobiliers-par-commune-et-par-annee-prix-et-volumes-sur-la-periode-2014-2023/
https://www.data.gouv.fr/fr/datasets/correspondance-entre-les-codes-postaux-et-codes-insee-des-communes-francaises/
https://www.data.gouv.fr/fr/datasets/bases-statistiques-communale-departementale-et-regionale-de-la-delinquance-enregistree-par-la-police-et-la-gendarmerie-nationales/
https://www.insee.fr/fr/statistiques/7752770?sommaire=7756859