Real estate price prediction

Abdellah Laassairi
Anass El Moubaraki

APST : Apprentissage statistique
Option mathématiques appliquées
Diplôme d'ingénieur généraliste

28/10/2022 - 13/12/2022

Tuteur académique :
Pr. Claire Brécheteau
Claire.Brecheteau@univ-nantes.fr

CENTRALE NANTES

Résumé

Ce rapport est centré sur le data challenge "Real estate price prediction" fournies par l'Institut Louis Bachelier. Dans notre travail, nous avons établi un pipeline de base de régression basé sur des state-of-the-art modèles tels que XGBoost et Catboost.

Ensuite, nous nous sommes appuyés sur les images fournies afin d'extraire des caractéristiques importantes en utilisant des modèles CNN pré-entraînés pour la classification des images, puis nous avons mis en œuvre NIMA (Neural Network Image Assesement) comme établi dans [8] ce qui n'a pas amélioré notre score. En outre, nous avons utilisé des données extraites du géopositionnement, ce qui a légèrement amélioré notre score final.

Table des matières

1 Introduction

1.1 Institut Louis Bachelier

L'Institut Louis Bachelier (ILB) est un réseau de recherche parrainé en économie et en finance, créée en 2008 à l'initiative du Trésor et de la Caisse des Dépôts et Consignations. Le datalab de l'ILB a été chargé de collecter un grand nombre de données sur l'immobilier français, qui sont les données que nous allons utiliser au cours de ce projet.

1.2 Challenge Data

Le projet est une tâche de régression qui traite de la prédiction des prix de l'immobilier. L'objectif est de modéliser l'immobilier résidentiel français en particulier à partir de données tabulaires et de quelques photos afin d'étudier l'influence de l'ajout d'images sur l'efficacité et la précision de notre modèle, et dans la dernière section, nous allons ajouter certaines variables extraites à partir de ces images et étudier leur qualité et nous montrerons que cela n'a pas eu de valeur supplémentaire, et nous expérimenterons différentes techniques de prétraitement et d'encodage des données afin d'optimiser nos modèles finaux.

1.3 Environnement de travail

Nous avons utilisé python et Jupyter notebooks pour enregistrer nos expériences, et nous nous sommes appuyés sur Git pour la collaboration et le contrôle de version du projet. Tous les différents frameworks et outils que nous avons utilisés, ainsi que la documentation, sont disponibles sur notre dépôt. [1]. Le notebook principal est nommé notebook.ipynb, et le pipeline de prétraitement final est nommé submission.ipynb, afin de garantir des résultats reproductibles et un rythme d'itération des expériences plus rapide.

FIGURE 1 – Ensemble de frameworks et d'outils utilisés pendant le projet

2 Approche initiale

2.1 Analyse exploratoire des données

Analyse exploratoire des données (AED) est la première étape importante à franchir afin d'extraire une compréhension approfondie de l'ensemble des données. C'est pourquoi nous avons décidé de commencer par mettre en œuvre les procédures suivantes :

1. Load our data into Pandas Data-Frames and get an overview of each variable (categorical and numerical variables, distribution of each variable, number of missing values, etc.)
2. Determine the correlation between different variables and try to analyze the relationships between them.

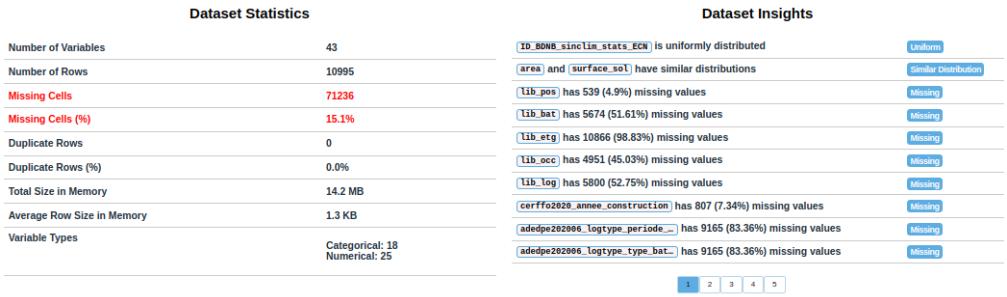

FIGURE 2 – Statistiques et aperçu de données de trainig à l'aide de l'AED

Nous avons décidé d'utiliser *dataprep.eda* pour faire l'AED.

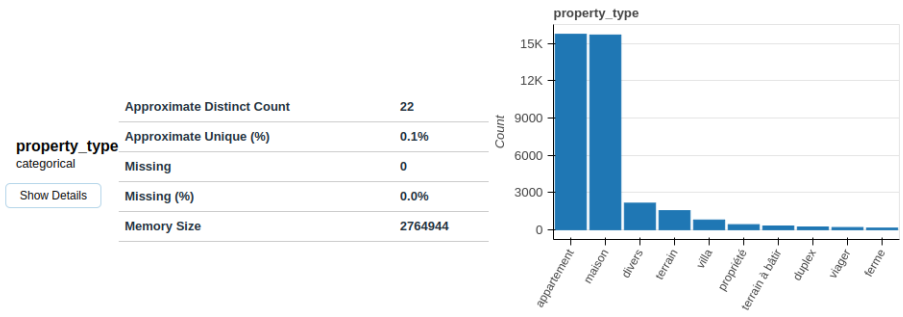

FIGURE 3 – Aperçu du variable Property_type à l'aide de l'AED

2.2 Prétraitement des données

À partir des différents éléments que nous avons établis dans la section précédente, nous pouvons maintenant commencer à nettoyer nos données et à les préparer pour notre modèle. Nous avons décidé d'appliquer les opérations suivantes à nos données :

1. Supprimer : $energy\_performance\_category$ et $ghg\_category$ car ces deux variables sont directement dérivées de ($energy\_performance\_value$, ghg_value).
2. Imputation constante : Nous allons remplir les valeurs manquantes du variable $floor$ avec une valeur de 0 pour les biens immobiliers qui ne sont pas des appartements.
3. Imputation avancée : Nous avons mis en œuvre plusieurs types d'imputation, comme l'imputation KNN (k nearest neighbors) [3], aainsi que l'imputation itérative [5]. L'imputation itérative nous a donné les meilleurs résultats.
4. Encodage : La variable ville a plus de 8000 valeurs différentes, ce qui va faire exploser la dimensionnalité, c'est pourquoi nous allons nous appuyer sur Frequency Encoding à la place (en remplaçant la valeur de la ville par sa fréquence dans notre ensemble de données) [9]. Nous avons expérimenté différentes méthodes d'encodage telles que Quantile Encoding et Target Encoding, mais elles ont entraîné un surajustement du modèle. Nous avons ensuite utilisé le Hot Encoding sur le reste des variables catégorielles.
5. Mise à l'échelle des variables : Nous avons expérimenté deux méthodes de mise à l'échelle.
   (a) Standard Scaler : qui suppose que les données sont normalement distribuées dans chaque caractéristique et les met à l'échelle de sorte que la distribution soit centrée autour de 0, avec un écart-type de 1.
   (b) Robust Scaler : qui supprime la médiane et met à l'échelle les données en fonction de la plage de quantiles, qui est plus robuste aux valeurs aberrantes. Et à partir de notre exploration des données, nous avons remarqué que beaucoup de nos variables contiennent diverses valeurs aberrantes, nous avons donc opté pour le Robust Scaler.

2.3 Transformation de la cible

Nous pouvons remarquer que la distribution de la variable *price* est asymétrique (non symétrique), ce qui viole certaines des hypothèses utilisées par certains modèles de régression que nous allons expérimenter dans les sections suivantes.

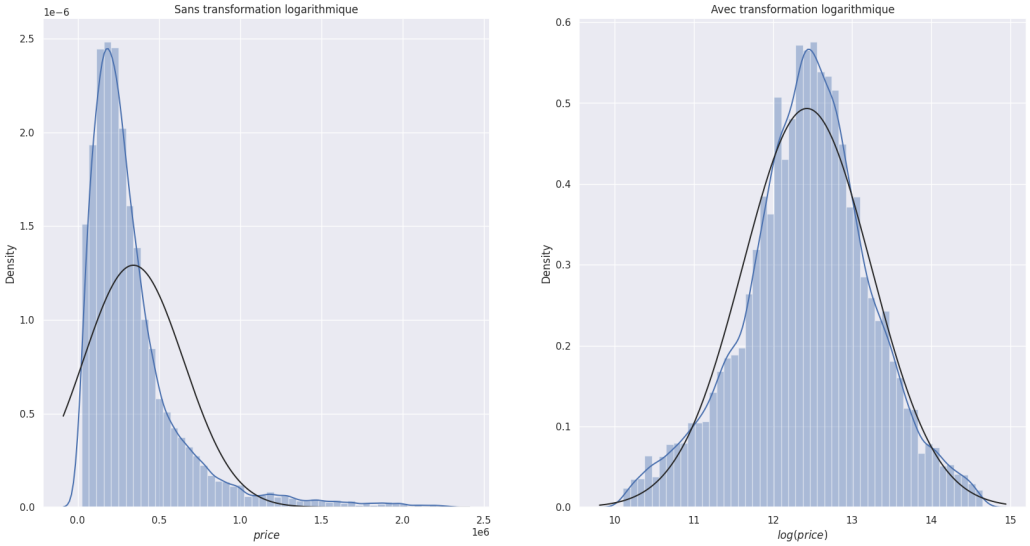

FIGURE 4 – Distribution du prix avec et sans transformation *log*

Nous pouvons également utiliser un transformateur quantile de sklearn au lieu d'appliquer une fonction logarithmique à notre cible.

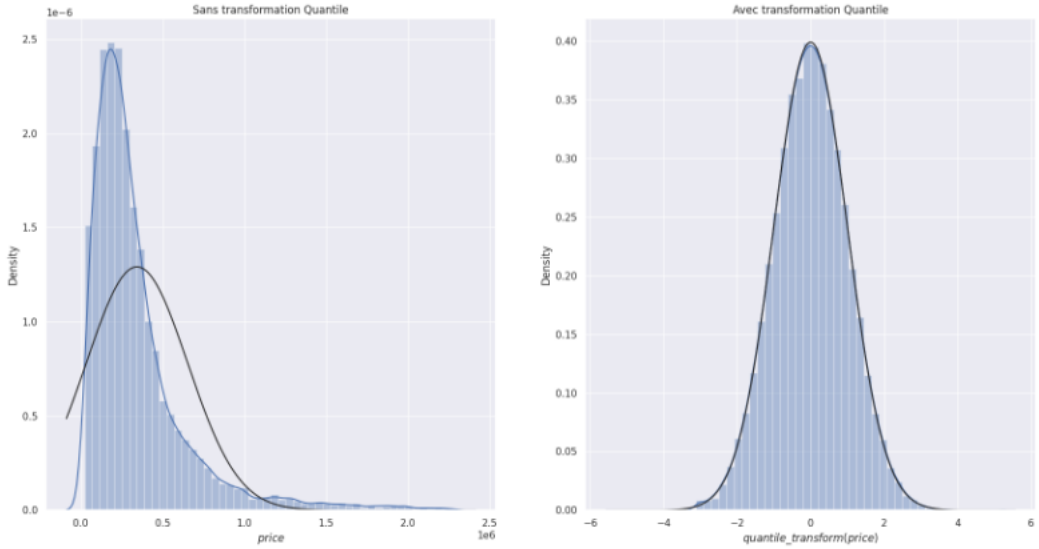

FIGURE 5 – Distribution du prix avec et sans transformation quantile

2.4 Sélection des modèles

Il existe plusieurs modèles de l'état de l'art qui s'appuient sur différentes technologies et architectures pour traiter les problèmes de régression. Afin de déterminer les meilleurs modèles possibles pour notre cas d'utilisation particulier, nous avons décidé d'utiliser Pycaret. PyCaret est une bibliothèque open-source d'apprentissage automatique en Python qui automatise les flux d'apprentissage automatique. Il s'agit essentiellement d'une enveloppe Python autour de plusieurs bibliothèques et frameworks d'apprentissage automatique tels que scikit-learn, XGBoost [4], LightGBM, CatBoost, etc.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	0.2739	0.1437	0.3791	0.7804	0.0288	0.0224	5.1710
lightgbm	Light Gradient Boosting Machine	0.2951	0.1622	0.4026	0.7523	0.0306	0.0241	0.5700
rf	Random Forest Regressor	0.2883	0.1680	0.4098	0.7434	0.0311	0.0236	13.6080
et	Extra Trees Regressor	0.2925	0.1727	0.4154	0.7363	0.0315	0.0239	8.5930
gbr	Gradient Boosting Regressor	0.3491	0.2159	0.4645	0.6703	0.0351	0.0284	9.9540
knn	K Neighbors Regressor	0.4098	0.3021	0.5495	0.5387	0.0414	0.0333	1.4720
ada	AdaBoost Regressor	0.4594	0.3398	0.5829	0.4812	0.0438	0.0372	3.6530
dt	Decision Tree Regressor	0.4150	0.3525	0.5936	0.4618	0.0450	0.0338	0.5850
lr	Linear Regression	0.5030	0.4273	0.6535	0.3474	0.0486	0.0407	0.0550
br	Bayesian Ridge	0.5030	0.4273	0.6536	0.3473	0.0486	0.0407	0.1950
ridge	Ridge Regression	0.5030	0.4274	0.6536	0.3472	0.0486	0.0407	0.0190
omp	Orthogonal Matching Pursuit	0.5323	0.4830	0.6948	0.2619	0.0515	0.0432	0.0220
huber	Huber Regressor	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.7840
xgboost	Extreme Gradient Boosting	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0560
lasso	Lasso Regression	0.6285	0.6553	0.8094	-0.0004	0.0608	0.0512	0.0520
en	Elastic Net	0.6285	0.6553	0.8094	-0.0004	0.0608	0.0512	0.0510
llar	Lasso Least Angle Regression	0.6285	0.6553	0.8094	-0.0004	0.0608	0.0512	0.0210
par	Passive Aggressive Regressor	0.9537	7.8580	2.3494	-11.0049	0.1313	0.0768	0.1120
lar	Least Angle Regression	149017317.7369	3876252326065508352.0000	635628021.4549	-5910379638688139264.0000	7.0194	11940351.7421	0.0310

FIGURE 6 – Comparaison des modèles basée sur le MAE

Nous pouvons constater que les meilleurs modèles pour ce problème sont Catboost, Xgboost, LightGBM, Random Forrest Regressor et Extra Trees Regressor. Dans les sections suivantes, nous utiliserons Catboost, Xgboost et LightGBM.

2.5 Optimisation des hyperparamètres

Après avoir déterminé les modèles les mieux adaptés à notre cas d'utilisation, nous allons optimiser leurs hyperparamètres respectifs. Au lieu de s'appuyer sur GridSearch ou RandomSearch. Nous allons utiliser Optuna. [2]. Optuna est un framework open-source d'optimisation d'hyperparamètres pour automatiser la recherche d'hyperparamètres.

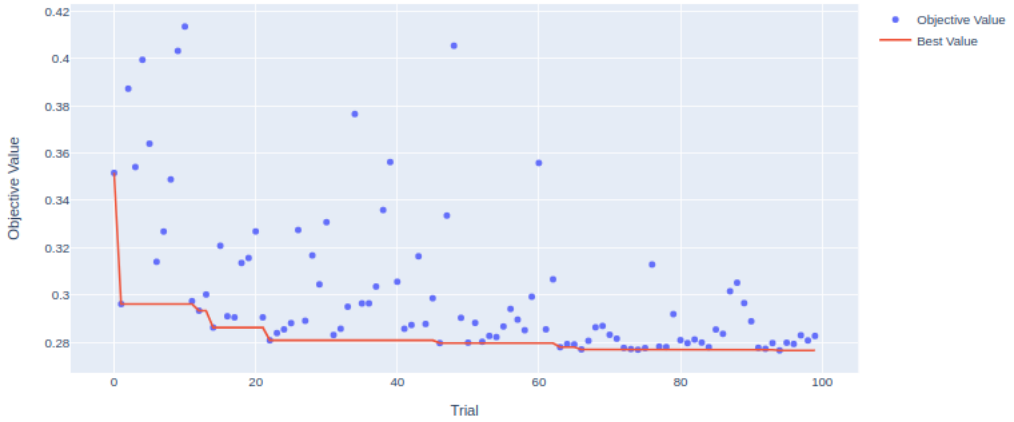

FIGURE 7 – Historique de l'optimisation de Xgboost

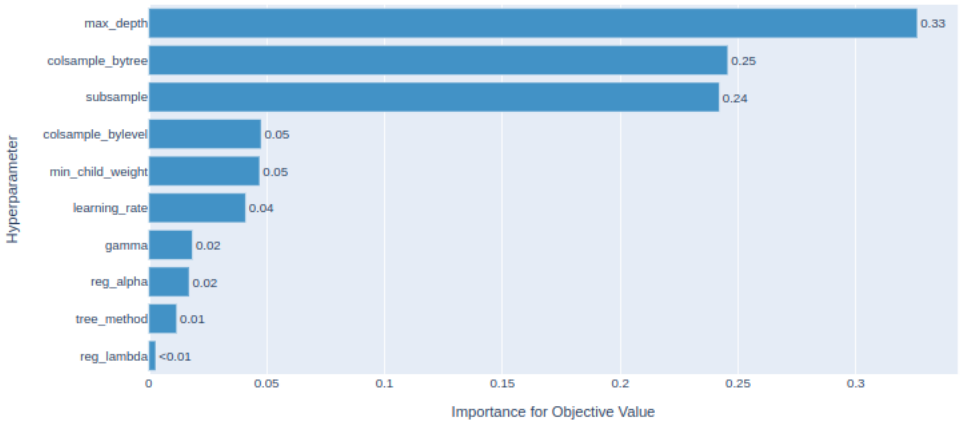

FIGURE 8 – Importance des hyperparamètres Xgboost

3 Ingénierie des variables de géo-positionnement

Nous savons par principe que l'un des facteurs les plus importants et les plus influents pour déterminer le prix d'un bien immobilier est son emplacement. Les variables *postal_code* et *city* ne sont pas suffisantes pour encapsuler complètement les données de position de chaque élément. De plus, la longitude et la latitude sont représentées séparément.

3.1 Représentation polaire

La première idée est d'ajouter des coordonnées polaires créées à partir de variables cartésiennes de latitude et de longitude pour un traitement ultérieur.

$$r = \sqrt{x^2 + y^2} \quad \text{et} \quad \phi = \arctan\left(\frac{x}{y}\right)$$

3.2 Rotation des variables de géolocalisation

La deuxième idée est de faire pivoter les coordonnées, en les faisant pivoter, elles fourniraient davantage d'informations spatiales pour les modèles de type arbre. qui sont extrêmement bénéfiques par rapport aux coordonnées x, y normales. Elles aident à visualiser les coordonnées selon différentes perceptions et donnent un aperçu des données que le modèle peut apprendre.

$$x_\omega = x \times \cos(\omega) + y \times \sin(\omega) \quad \text{et} \quad y_\omega = x \times \sin(\omega) - y \times \cos(\omega)$$

3.3 ACP des variables de géolocalisation

Nous pouvons également utiliser l'ACP pour faire pivoter l'espace de coordonnées cartésiennes, ce qui faciliterait la division des arbres de décision. En utilisant l'ACP du sklearn, il apprend automatiquement le meilleur angle de rotation basé sur la densité des points dans l'espace de coordonnées et peut être utilisé pour transformer ces points dans le même espace, ce qui donne plus de précision.

3.4 Distance de la Haversine au centre ville

Nous utiliserons la formule d'Haversine pour calculer la distance d'Haversine entre le centre ville du bien immobilier et l'emplacement du bien immobilier lui-même.

$$d = 2R \arcsin \sqrt{\sin^2 \frac{\varphi_1 - \varphi_2}{2} + \cos\varphi_1 \cos\varphi_2 \sin^2 \frac{\lambda_1 - \lambda_2}{2}}$$

3.5 Proximité des biens immobiliers

Dans cette section, nous avons extrait des variables basées sur la localisation en utilisant google maps (places API) afin d'extraire des variables précieuses, par exemple le nombre d'hôpitaux à proximité du bien, le nombre de gares, le nombre d'écoles, etc. ainsi que la distance minimale entre le bien immobilier et chacun de ces établissements.

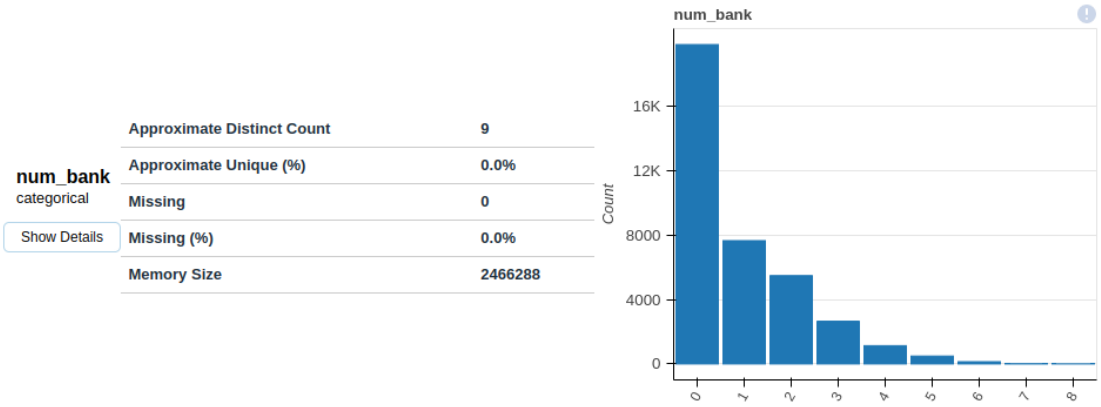

FIGURE 9 – Nombre de banques à proximité des biens immobiliers (rayon de 5 km)

Nous avons ensuite concaténé ces résultats à nos données tabulaires après avoir appliqué certaines étapes de prétraitement.

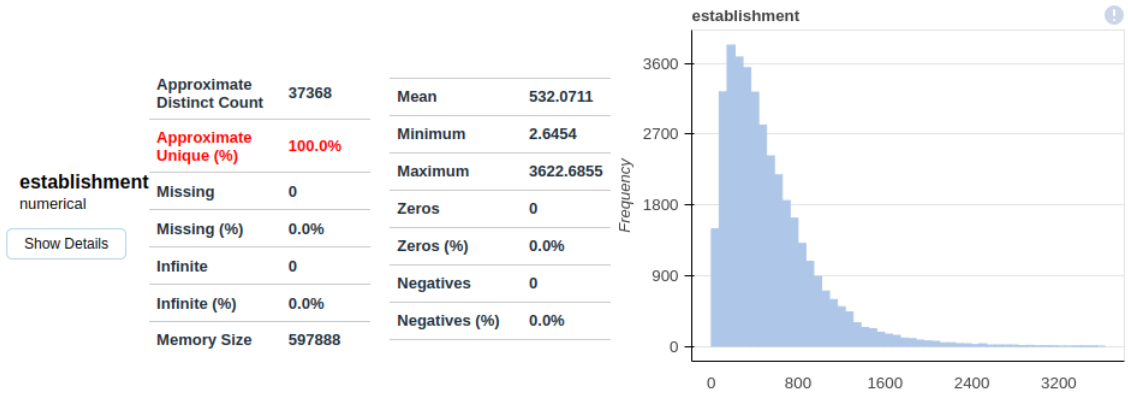

FIGURE 10 – Distance jusqu'à l'établissement le plus proche d'un bien immobilier (en m)

4 Classification des images et extraction de variables

4.1 Variables statistiques

La première approche serait d'ajouter des variables statistiques extraites des images telles que la luminosité, le contraste, la couleur dominante, etc. et de les concaténer à nos données tabulaires. Cependant, en utilisant notre modèle de base pour évaluer cette approche, nous avons rapidement remarqué que l'ajout de telles variables ne fait que détériorer considérablement nos performances et que ces caractéristiques n'agissent que comme du bruit.

4.2 Classification des images

Un autre facteur clé qui affecte la valeur d'un bien immobilier donné est son apparence intérieure et extérieure. D'où l'idée d'utiliser des réseaux neuronaux convolutifs profonds sur l'ensemble de données de photos pour en extraire des variables significatives. Nous avons suivi l'approche de l'article Vision-based real estate price estimation [7] et mis en œuvre un modèle Efficientnet pour classer les images en 7 classes à partir du jeu de données fourni par l'auteur de l'article précédent, mais en raison de ressources matérielles insuffisantes, nous avons opté pour un modèle pré-entraîné sur 6 classes à la place : bedroom, bathroom, kitchen, Frontyard, Backyard and LivingRoom.

FIGURE 11 – Exemple d'images classées

Nous avons ensuite exécuté l'inférence sur les images de l'ensemble d'entraînement et de l'ensemble de test et nous les avons classées, puis nous avons ajouté le nombre de chaque classe comme une nouvelle colonne à nos données tabulaires.

4.3 Évaluation de la qualité des images par réseau neuronal

Nous avons également mis en œuvre NIMA (Neural Network Image Assessement) [8] qui est une méthode d'évaluation d'image basée sur CNN entraînée sur des ensembles de données de qualité à la fois esthétique et au niveau du pixel afin de comparer la qualité et l'esthétique d'images précédemment classées et de les concaténer à nos données tabulaires.

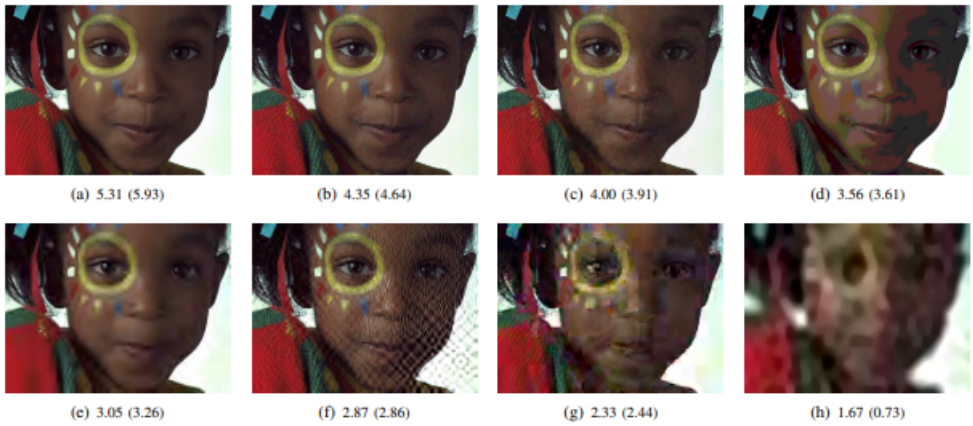

FIGURE 12 – Classement des images en utilisant le modèle NIMA [8]. Les scores prédits (et la vérité terrain) sont indiqués sous chaque image.

5 Résultats

Il y a trop de variables à prendre en compte pour comparer les trois approches. Notre pipeline d'inférence ressemble à la figure suivante.

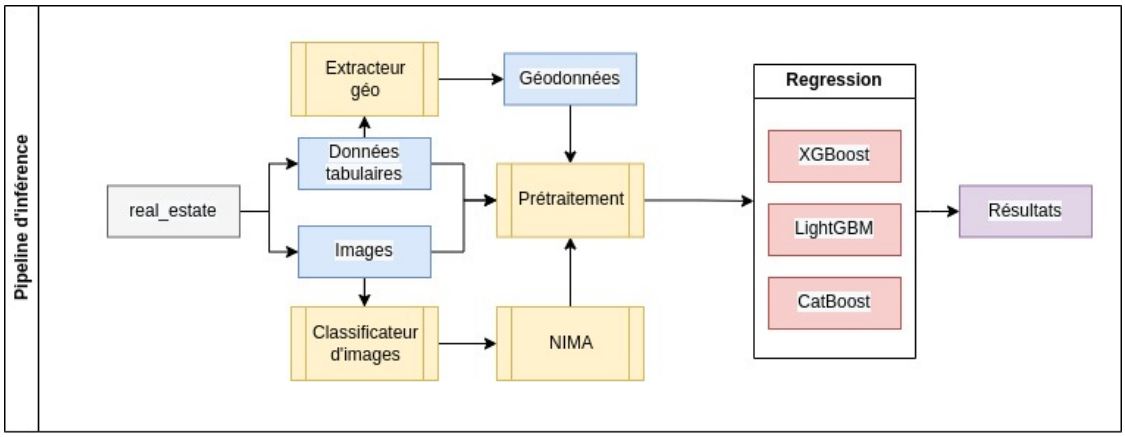

FIGURE 13 – Pipeline d'inférence

La meilleure stratégie que nous avons obtenue est l'utilisation du 'Ensemble modeling' optimisée (Xgboost, LightGBM, Catboost), de l'ingénierie des géodonnées avec un score de soumission final de **24.00** Et nous avons été classés 5ème au classement public, cela en comparaison avec un score de référence de 36.78. L'ajout d'images a légèrement amélioré le score lors de l'ajout de variable nFrontyards pendant l'une de nos expériences, mais cela n'a pas été reproductible.

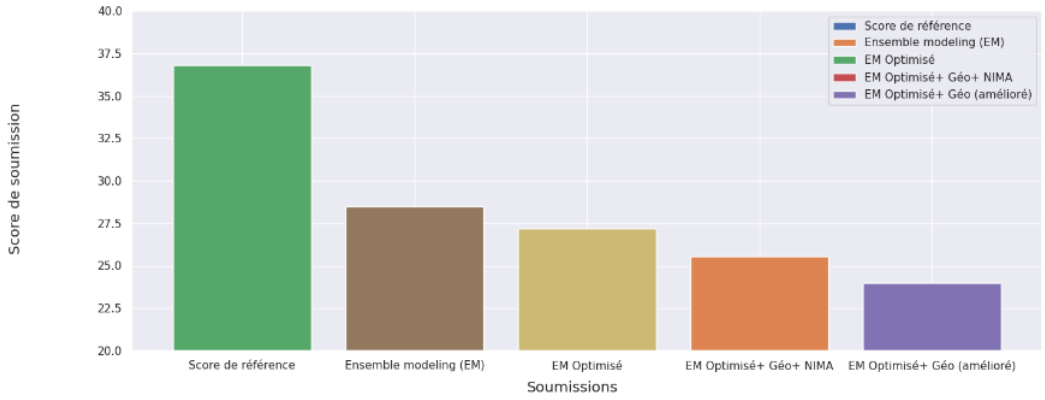

FIGURE 14 – Comparaison de différentes scores de soumission

6 Défis

Nous avons fait face à plusieurs défis pendant que nous nous attaquions à ce projet, de la collaboration sur le même code source, et le maintien de résultats lisibles et reproductibles, à la manipulation de grands ensembles de données d'entraînement sur nos machines. Les principaux obstacles étaient :

1. Entraînement du modèle de classification des images : L'un des principaux problèmes auxquels nous avons été confrontés a été l'entraînement et l'exécution de l'inférence sur le jeu de données de plus de 30 Go fourni sur nos machines, car il n'était pas possible de le télécharger sur google collab, et nous n'avions pas les fonds nécessaires pour utiliser un autre service de cloud computing.
2. Surajustement our Sur-apprentissage : Nous avons également été confrontés à la difficulté de déboguer nos modèles et d'essayer d'identifier les chemins exacts à suivre pour optimiser nos résultats sans sur-ajustement, un exemple clair de cela était apparent lorsque nous avons implémenté Quantile Encoding [6] pour notre variable 'city', et nous a donné un score de validation croisée de 23 MAPE localement, mais était en fait de 28 lors de la soumission.

7 Conclusion

Les aspects les plus importants de ce défi étaient en fait le prétraitement et le nettoyage des données, ainsi que l'ingénierie des variables, qui ont eu le plus grand impact sur notre score par rapport à toutes les autres implémentations que nous avons réalisées. Nous avons réussi à augmenter encore notre score en utilisant des données de géopositionnement. Cependant, les images étaient beaucoup trop variables en taille/qualité et il y avait parmi elles de nombreuses images non représentatives de l'immobilier (images du logo de l'entreprise, images sans rapport), ce qui ne nous a pas permis d'en extraire des données pertinentes. Nous manquons également d'un ensemble de données d'entraînement étiquetées pour le niveau de luxe ou de qualité de l'image afin d'obtenir des améliorations de performance valables.

8 Remerciements

Nous aimerions tout d'abord remercier l'équipe Data (ENS Paris) et le Collège de France pour la gestion de ce concours ainsi que le DataLab de l'Institut Louis Bachelier pour avoir fourni le jeu de données et le défi, nous aimerions également remercier notre professeur Claire Brecheteau pour les diverses connaissances qu'elle nous a enseignées pendant notre cours d'apprentissage statistique et que nous avons appliquées à notre solution.

Références

[1] Laassairi Abdellah. Data challenge 2022 project repository. `https://github.com/Abdellah-Laassairi/real-estate-price-prediction`, 2022.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna : A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[3] Gustavo Batista and Maria-Carolina Monard. A study of k-nearest neighbour as an imputation method. volume 30, pages 251–260, 01 2002.

[4] Tianqi Chen and Carlos Guestrin. XGBoost : A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[5] Daniel Jarrett, Bogdan Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. Hyperimpute : Generalized iterative imputation with automatic model selection. 2022.

[6] Carlos Mougan, David Masip, Jordi Nin, and Oriol Pujol. Quantile encoder : Tackling high cardinality categorical features in regression problems. *CoRR*, abs/2105.13783, 2021.

[7] Omid Poursaeed, Tomáš Matera, and Serge Belongie. Vision-based real estate price estimation. *Machine Vision and Applications*, 29(4) :667–676, 2018.

[8] Hossein Talebi and Peyman Milanfar. Nima : Neural image assessment. *Transactions on Image Processing*, 2018.

[9] Asli Uyar, Ayse Bener, H. Ciray, and Mustafa Bahceci. A frequency based encoding technique for transformation of categorical variables in mixed ivf dataset. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2009 :6214–7, 09 2009.