Modèle Linéaire: Prédiction des Prix Immobiliers en Californie

Souhaila Benrachid, Sabine Mansour, Zakaria Souidaray

Année 2024/2025
Université Aix-Marseille
M1 Data science

I. Introduction

L'objectif de ce projet est de développer un modèle de machine learning permettant de prédire les prix immobiliers en Californie. Les données utilisées proviennent du *California Housing Dataset*, disponible dans la bibliothèque `sklearn` de Python. Ce jeu de données contient des informations détaillées sur les logements en Californie et offre une base solide pour effectuer des analyses prédictives.

Les variables du jeu de données sont les suivantes :

- **MedInc** : revenu médian du bloc.
- **HouseAge** : âge médian des logements du bloc.
- **AveRooms** : nombre moyen de pièces par ménage.
- **AveBedrms** : nombre moyen de chambres par ménage.
- **Population** : population du bloc.
- **AveOccup** : nombre moyen de personnes par ménage.
- **Latitude** : latitude du bloc.
- **Longitude** : longitude du bloc.

La variable cible est la **valeur médiane des logements** pour les districts de Californie, exprimée en centaines de milliers de dollars ($100,000).

Ce jeu de données a été dérivé du recensement américain de 1990. Chaque ligne représente un groupe de blocs de recensement, qui est la plus petite unité géographique pour laquelle le Bureau du recensement des États-Unis publie des données d'échantillon. Un groupe de blocs regroupe généralement une population comprise entre 600 et 3 000 personnes. Par ailleurs, un *ménage* désigne un groupe de personnes vivant dans le même logement.

Cette étude vise à tirer parti de ces données pour construire un modèle performant, capable de prédire avec précision les prix immobiliers en fonction des caractéristiques des logements et de leur environnement.

II. Préparation des Données : Training/test split et nettoyage des données

Avant de commencer toute analyse ou modélisation, il est essentiel de diviser le jeu de données en deux sous-ensembles distincts : un jeu d'entraînement et

un jeu de test(C'est ce qu'on appelle le train/test split). Cette séparation est une étape cruciale pour garantir la robustesse et la généralisation du modèle.

Le **jeu d'entraînement** est utilisé pour ajuster les paramètres du modèle. En d'autres termes, il permet au modèle d'apprendre les relations entre les variables explicatives et la variable cible. Cependant, un modèle bien ajusté sur les données d'entraînement ne garantit pas une bonne performance sur des données inconnues.

Le **jeu de test**, quant à lui, est réservé à l'évaluation finale du modèle. En le maintenant complètement indépendant des données d'entraînement, nous pouvons estimer comment le modèle se comportera face à de nouvelles données réelles. Cela permet de détecter les problèmes de surapprentissage (*overfitting*) ou de sous-apprentissage (*underfitting*).

Cette séparation dès le début du projet est essentielle pour éviter tout biais d'évaluation et garantir que les résultats obtenus reflètent la performance réelle du modèle. Sans cette étape, il serait difficile de savoir si le modèle est véritablement efficace ou s'il s'est simplement adapté aux particularités du jeu de données initial.

Une fois cette séparation réalisée, nous pourrons procéder au nettoyage et à la préparation des données, afin de garantir leur qualité et leur adéquation aux besoins du modèle.

Lors de la phase de nettoyage des données, nous avons observé que le jeu de données ne contenait ni valeurs manquantes ni doublons. Cette observation a considérablement simplifié notre tâche.

Cependant, nous avons identifié la présence de **valeurs aberrantes** (*outliers*) dans certaines variables explicatives. Ces valeurs, qui s'écartent significativement du reste des données, peuvent potentiellement influencer la performance du modèle. Une attention particulière a donc été accordée à ces outliers pour les identifier et les traiter. Des méthodes comme la transformation logarithmique ou la *winsorisation* ont été envisagées pour limiter l'impact de ces valeurs extrêmes tout en conservant la structure globale des données. Ces étapes sont essentielles pour s'assurer que les données sont adaptées à l'analyse et ne biaisent pas la performance du modèle.

III.Création de Variables de Moyenne des Prix Médian dans un Rayon Autour de Chaque Bloc

Après avoir nettoyé les données, nous avons jugé pertinent de remplacer les variables **longitude** et **latitude** par de nouvelles variables plus faciles à manipuler et à interpréter.ous avons appliqué cette transformation séparément pour le jeu d'entraînement et le jeu de test, en calculant les moyennes dans chaque ensemble de données indépendamment, afin de préserver leur intégrité et d'éviter toute fuite d'information. L'objectif était de mieux capturer les effets spatiaux tout en simplifiant l'analyse.

Pour cela, nous avons créé trois nouvelles variables représentant la moyenne des prix médians des logements dans un rayon de x km autour de chaque bloc (bloc exclu) pour les distances suivantes :

- Moyenne des prix médians dans un rayon de **5 km**,
- Moyenne des prix médians dans un rayon de **10 km**,
- Moyenne des prix médians dans un rayon de **15 km**.

On a trouvé que elles remplacent efficacement les coordonnées géographiques brutes, qui peuvent être difficiles à exploiter directement dans un modèle prédictif.

IV.Modélisation et Évaluation du Modèle de Prédiction des Prix Immobiliers

Avant d'entraîner le modèle, nous avons analysé les relations entre les variables en calculant la matrice de corrélation ainsi que les vecteurs propres de cette matrice.

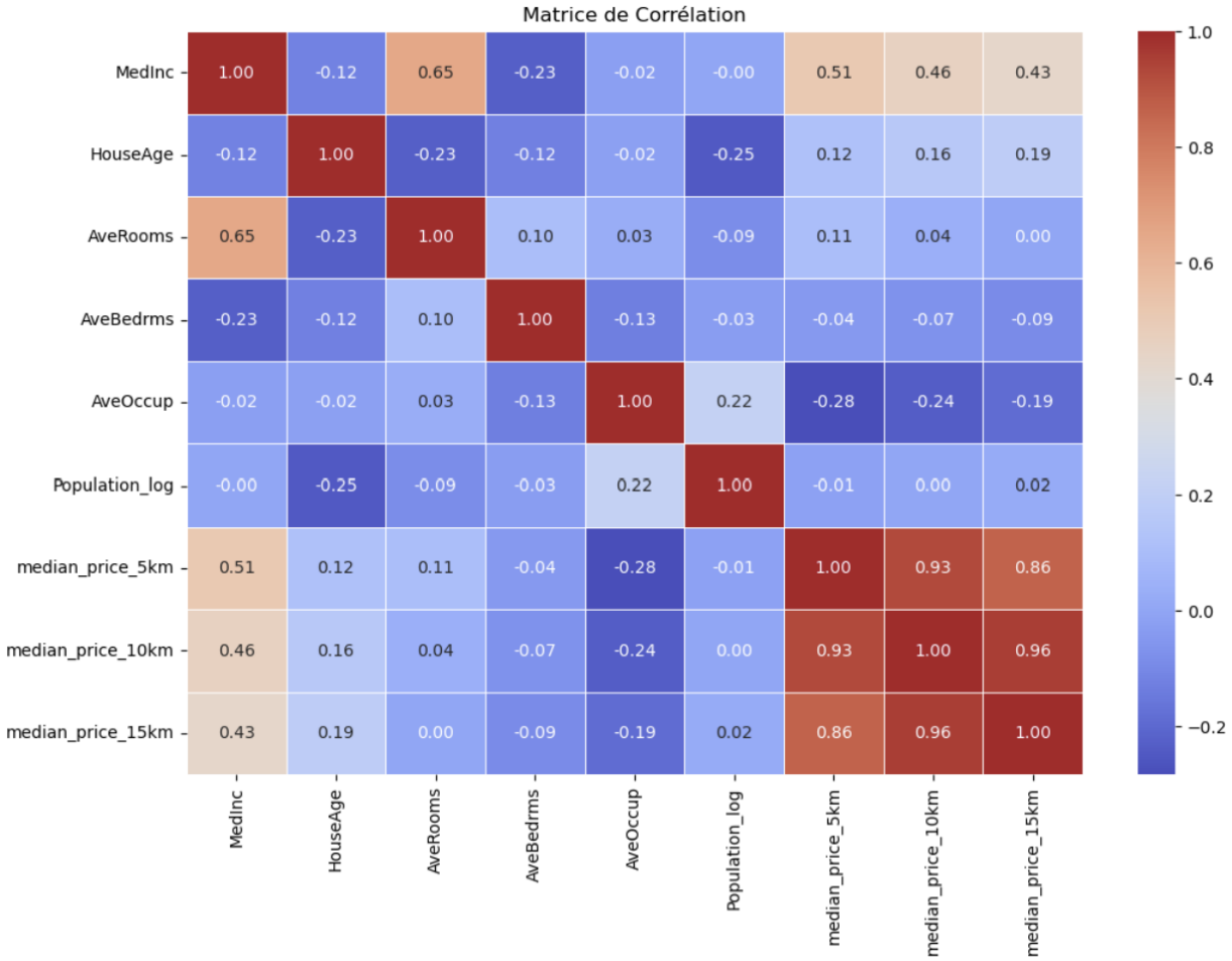

Figure 1: Matrice de Correlation.

```
Valeurs propres de la matrice de corrélation :
[0.02786483 0.12161714 0.18086141 0.54425071 0.79756073 1.14536537
 1.26675797 1.63825066 3.27747119]
```

Figure 2: Valeurs Propres de la matrice de correlation.

Nous avons constaté une forte colinéarité entre certaines variables, en particulier les prix médians dans les rayons de 5 km, 10 km et 15 km, ce qui est confirmé par des valeurs propres proches de zéro dans la matrice de corrélation. Ce phénomène peut poser des défis dans les modèles de régression classiques, notamment une instabilité dans l'estimation des coefficients. Pour y remédier et éviter tout risque de surapprentissage, nous avons opté pour la **régression Ridge**, qui introduit une régularisation permettant de réduire les coefficients excessifs, assurant ainsi la stabilité du modèle tout en préservant l'essentiel des informations véhiculées par ces variables. Pour déterminer le paramètre de régularisation λ, nous avons utilisé la validation croisée et considéré deux approches :

- $\widehat{\lambda}_{\text{min}}$: la valeur de λ qui minimise l'erreur de prédiction estimée par validation croisée,
- $\widehat{\lambda}_{\text{1se}}$: la plus grande valeur de λ telle que l'erreur de prédiction moins l'erreur-type reste inférieure à l'erreur minimale.

Ona ainsi obtenue:

$\widehat{\lambda}_{\text{min}}$ vaut 6.58 et $\widehat{\lambda}_{\text{1se}}$ vaut 1522.00

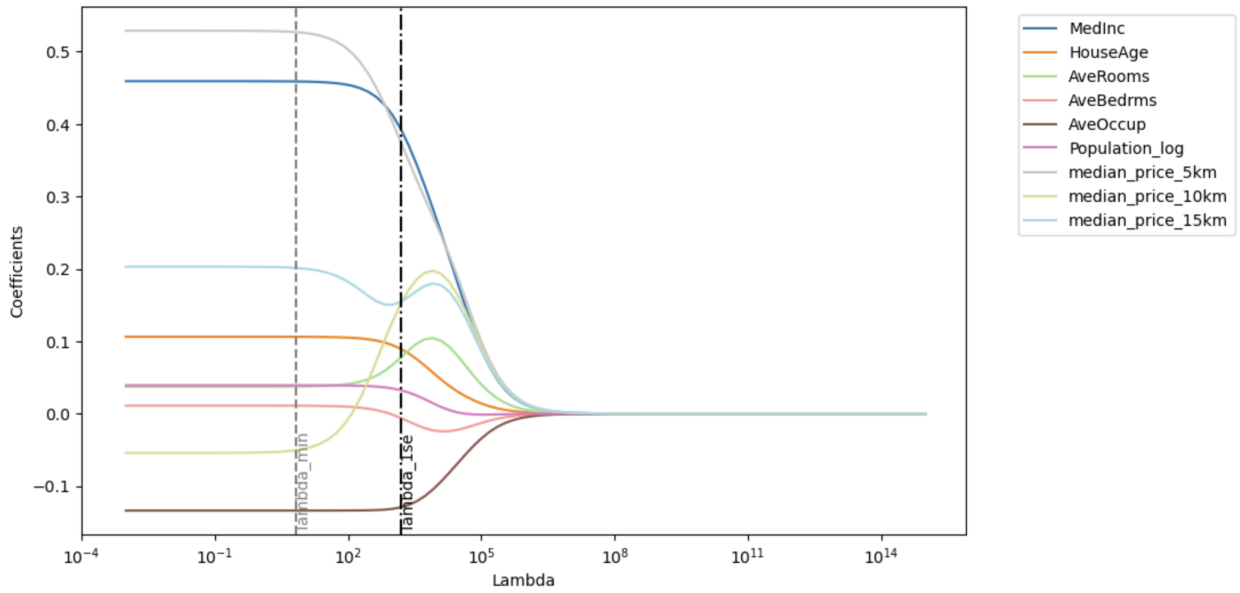

Figure 3: Evolution des coefficients en fonction de sur une échelle logarithmique.

Nous observons que les coefficients des variables diminuent progressivement à mesure que la valeur de λ augmente. Une régularisation plus forte réduit ainsi l'impact de ces coefficients, limitant la complexité du modèle. Les lignes verticales indiquent les valeurs de $\widehat{\lambda}_{\text{min}}$ et $\widehat{\lambda}_{\text{1se}}$, qui représentent respectivement le λ minimisant l'erreur et celui offrant un compromis entre une erreur acceptable et une régularisation plus importante. Nous ajustons alors le modèle de régression Ridge en utilisant le $\widehat{\lambda}_{\text{1se}}$, qui est plus conservateur que le $\widehat{\lambda}_{\text{min}}$, car il pénalise davantage les coefficients, ce qui le rend souvent plus robuste aux variations des données et plus stable.

V. Évaluation du Modèle sur le Jeu de Test

Lors de la phase de modélisation, nous avons utilisé le **jeu d'entraînement** pour construire et ajuster le modèle. Cependant, il est essentiel de vérifier si ce modèle est capable de bien **se généraliser** sur des données qu'il n'a jamais vues auparavant. Pour cela, nous utilisons le **jeu de test**, qui est resté indépendant tout au long du processus d'entraînement. Pour évaluer les performances du modèle, nous utilisons deux métriques principales : le **RMSE** et le $\mathbf{R^2}$.

```
RMSE sur les données d'entraînement : 0.5416349599586702
RMSE sur les données de test : 0.5646017241334396
R squared sur les données d'entraînement :  0.7828686687789165
R squared sur les données de test :  0.7590632067852194
```

Figure 4: Évaluation des Performances du Modèle

Le **RMSE** mesure l'écart moyen entre les valeurs prédites et les valeurs réelles. Un RMSE faible indique que le modèle fait des prédictions précises. Dans ce cas, le RMSE est de 0.55 sur les données d'entraînement et de 0.56 sur les données de test, ce qui reflète une performance stable et cohérente entre les deux ensembles de données.

Le $\mathbf{R^2}$ évalue la proportion de la variance expliquée par le modèle. Avec des valeurs de 0.78 pour les données d'entraînement et de 0.76 pour les données de test, le modèle explique efficacement la variation des prix immobiliers. Ces résultats montrent que le modèle est capable de capturer les relations importantes entre les variables explicatives et la variable cible, tout en maintenant une bonne capacité de généralisation.

VI.Conclusion

Ce projet avait pour objectif de prédire les prix immobiliers en Californie à partir du *California Housing Dataset*. En appliquant des techniques de régression linéaire, notamment la régression Ridge, nous avons procédé à une analyse approfondie des données et développé un modèle prédictif robuste. Après avoir nettoyé et préparé les données, nous avons identifié des problèmes tels que la multicolinéarité entre certaines variables et avons utilisé la régularisation pour stabiliser les coefficients et prévenir le surapprentissage.

Les résultats obtenus montrent que le modèle Ridge a bien performé, avec un RMSE faible et un R^2 élevé, attestant de sa capacité à généraliser efficacement sur les données de test.