université
PARIS-SACLAY

Projet machine learning

Prédiction du prix de l'immobilier en Californie

SOMMAIRE:

Introduction

La prédiction du prix de l'immobilier est une tâche complexe qui peut bénéficier de l'utilisation de techniques de machine learning. Dans ce rapport, nous décrivons la création d'un modèle de machine learning utilisant la méthode K-neighbors et la méthode de régression arborescente pour prédire le prix de l'immobilier. Nous avons suivi plusieurs étapes clés pour optimiser nos modèles, notamment la suppression des outliers, la sélection des meilleures colonnes, la standardisation des données et l'ajustement des paramètres tels que la distance métrique, le nombre de voisins et la profondeur de l'arbre et le nombre minimal d'observations pour diviser un noeud.Les détails de ces étapes sont présentés dans les sections suivantes.

Lien du dataset :

https://www.kaggle.com/datasets/dhirajnirne/california-housing-data

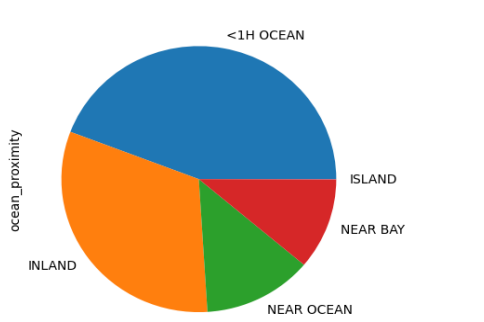
Première visualisation des données:

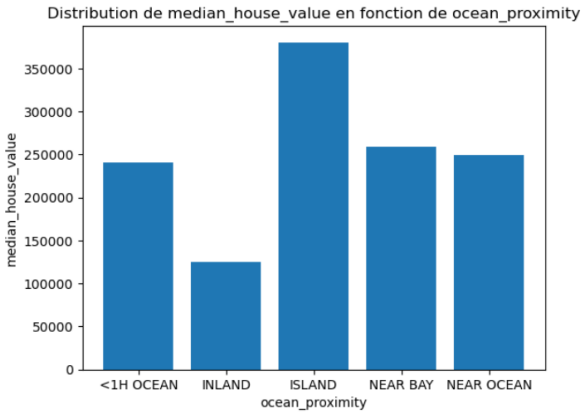
Notre objectif durant ce projet était de prédire le prix de l'immobilier en Californie avec des caractéristiques donnés, afin de se situer sur le marché de l'immobilier, en nous référant à l'attribut: " *median_house_value*".

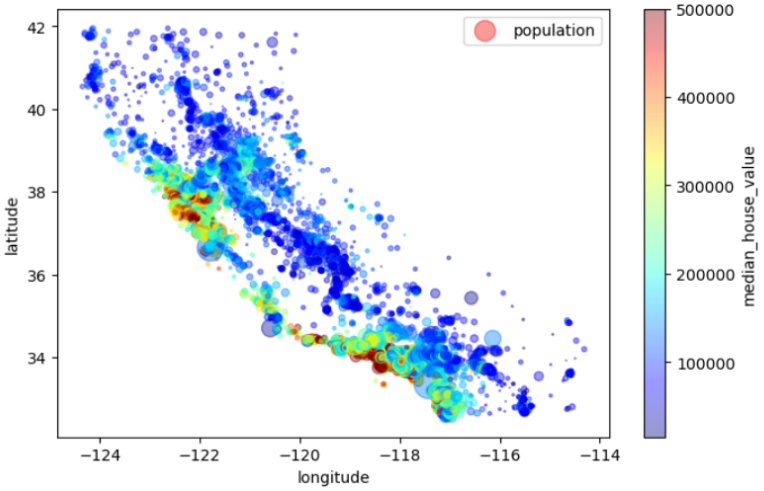
	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY
...	...	...	...	...	...	...	...	...	...	...
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.0	1.5603	78100.0	INLAND
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.0	2.5568	77100.0	INLAND
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.0	1.7000	92300.0	INLAND
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.0	1.8672	84700.0	INLAND

Pré-visualisation des données

Nous nous sommes tout d'abord assurés de la présence de valeurs Null dans des lignes du dataset, destinées à être supprimées lors de la phase suivante.
Par la suite, nous avons affiché les différentes valeurs possibles de l'attribut catégoriel("ocean_proximity")
Ensuite, nous avons également ,une analyse des variables quantitatives pour un ensemble de données de maisons. Nous affichons la moyenne de la variable quantitative "median_house_value" pour chaque groupe de la variable catégorielle "ocean_proximity".
Finalement nous avons généré une carte de visualisation de données pour notre ensemble de données qui représente un ensemble de nuages avec l' axe x représentant la longitude et l'axe de y représentant l'altitude et la taille des points représentant la population, la couleur du point représente la valeur de la variable "median_house_value ".







Preprocessing

Nettoyage des données

Passons maintenant à la phase de preprocessing, il nous incombait de procéder tout d'abord à un nettoyage des données. Ce nettoyage a pour but de rendre les données de notre dataset exploitables.

Standardisation des données : La standardisation des données est une étape essentielle pour garantir que toutes les variables d'entrée de notre modèle ont la même échelle. Nous avons utilisé des méthodes de scaling telles que la standardisation (z-score) pour normaliser les valeurs de nos colonnes et éviter que certaines variables aient un poids disproportionné dans notre modèle en raison de leur échelle différente.

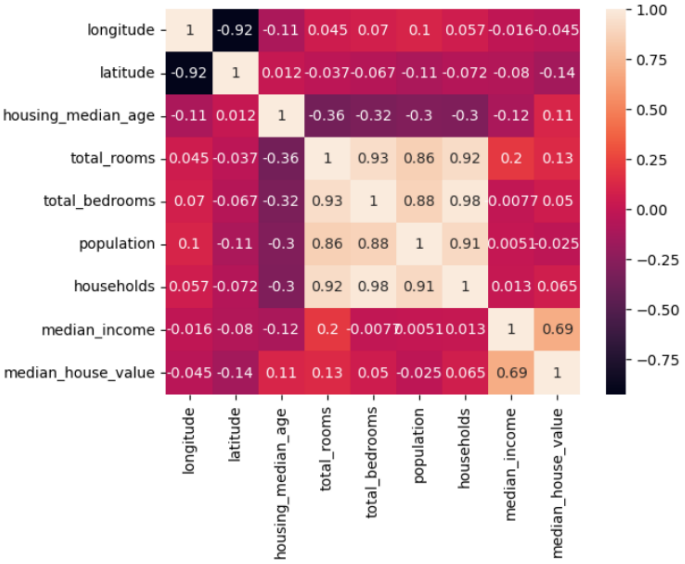
Suppression des outliers : Les outliers sont des valeurs aberrantes qui peuvent fausser les résultats d'un modèle de machine learning. Pour garantir l'intégrité de notre modèle, nous avons utilisé des méthodes statistiques pour détecter et supprimer les outliers de notre jeu de données. Nous avons utilisé des techniques telles que la méthode des écarts

interquartiles ou encore le boxplot pour identifier les valeurs aberrantes et les éliminer de notre ensemble de données.

supprimer valeurs aberrantes : Elles ont Influencé négativement sur les résultats statistiques tel que la moyenne, la variance et la corrélation mais surtout nous avons constaté qu'il y avait des erreurs de saisie dans les colonnes du data set.

Sélection des meilleures colonnes : La sélection des meilleures colonnes est une étape importante pour améliorer la performance de notre modèle. Nous avons examiné la corrélation entre les différentes colonnes de notre jeu de données pour identifier les colonnes qui ont une corrélation élevée avec la variable cible, c'est-à-dire le prix de l'immobilier. Les colonnes présentant une corrélation faible ou nulle ont été éliminées pour réduire la dimensionnalité de notre modèle et améliorer son efficacité.

One-hot coding : le one-hot coding permet de traiter les variables catégorielles ("ocean_proximity) avec un grand nombre de catégories(4 catégories dans notre dataset).
Nous avons créé la fonction encodage qui consiste à créer des variables binaires ou indicatrices pour chaque catégorie de la variable catégorielle "ocean proximity".
Ci-dessous la matrice de corrélation :



Efficacité des modèles

Pour conclure ce projet, nous avons eu recours à plusieurs modèles d'apprentissage statistique :l'algorithme de **K-Neighbors** et l'algorithme de **régression arborescente**. Ces derniers nécessitent le choix d'un hyper-paramètre. (que nous allons cité)
Ces deux algorithmes ont été étudiés à travers plusieurs indicateurs de performance : le score pour chacun,la variance,l'erreur quadratique.

1- K-Neighbors pour la prédiction du prix de l'immobilier

1. Définition :

K-nearest neighbors (KNN) regressor est un algorithme de machine Learning supervisé utilisé pour la prédiction de valeurs continues. Il se base sur les valeurs de k voisins les plus proches dans l'espace des caractéristiques pour effectuer une prédiction.

Le choix de ces hyperparamètres peut affecter la performance et la généralisation du modèle.

2. Choisir l'hyper paramètre optimal :

Je résume dans ce tableau comment nous avons procédé :

ID	Suppression des outliers	Normaliser les données :	Sélection des colonnes	n_neighbors	Metric	Weights	Score
0	Non	Non	Non	3	Défaut	Uniform	0.223
1	Non	Non	Non	Défaut	Défaut	Défaut	0.255
2	Non	Non	Non	15	Défaut	Défaut	0.317
3	Oui	Non	Non	15	Défaut	Défaut	0.317
4	Oui	Oui	Oui	19	Minkowski	Uniform	0.7
5	Oui	Oui	Oui	10	Manhattan	Uniform	0.73
7	Oui	Oui	Non	9	Manhattan	Distance	0.76

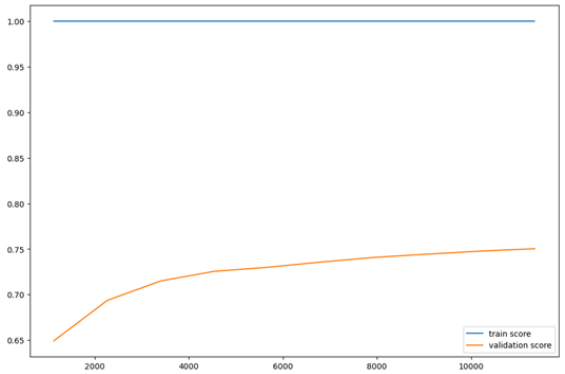
Pour atteindre les meilleures performances (un grand score et un petit overfitting), nous avons procédé de manière systématique en manipulant les hyper paramètres essentiels, notamment "n_neighbors" et "Metric", en utilisant une double boucle imbriquée pour évaluer le score en fonction de ces deux hyperparamètres et trouver les meilleures valeurs.

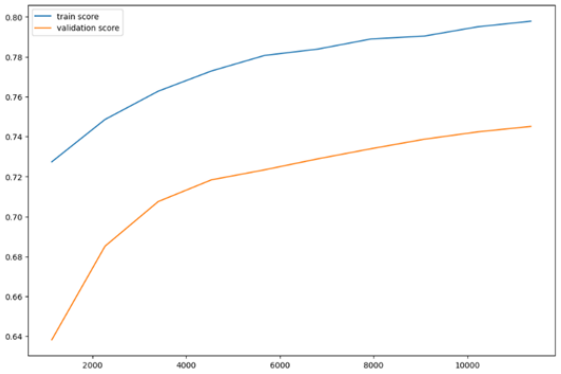
Nous avons également modifié les pourcentages du train set et du test set pour évaluer leur impact sur les résultats.

Il est intéressant de noter que **la suppression des outliers** avant et après d'avoir fait **feature engineering** a conduit à une augmentation du score de 1%, et que nous avons observé que toutes les colonnes de notre dataset étaient importantes, car la performance du modèle a diminué en sélectionnant seulement 13/15 colonnes.

Nous avons également utilisé **la validation croisée** pour estimer la capacité de généralisation du modèle à de nouvelles données non vues auparavant, et tracé des **courbes d'apprentissage** pour observer comment la performance du modèle évolue en fonction du nombre d'exemples d'entraînement utilisés.KNN regressor est très sensible aux grandes valeurs en particulier si la distance utilisée pour mesurer la similarité entre les points de données est sensible aux écarts entre les valeurs des caractéristiques.

Modifier les pourcentages du train set et du test set afin de visualiser son impact sur les résultats ,Il est important de noter que nous avons observé que **la variance de l'erreur** est inférieure à **l'erreur quadratique moyenne**, ce qui indique que notre modèle était capable de faire des prédictions cohérentes et précises, évitant ainsi le surajustement (overfitting), ce qui est souhaitable pour un modèle performant. **Learning curve** est tracée en utilisant l'erreur d'entraînement et l'erreur de validation du modèle en fonction du nombre d'exemples d'entraînement utilisés. Elle permet d'observer la performance du modèle.





2-Régression arborescente pour la prédiction du prix de l'immobilier

L'algorithme de régression arborescente est souvent utilisé pour prédire le prix de l'immobilier car il peut traiter des ensembles de données complexes avec de nombreuses variables prédictives(total_rooms,total_bedrooms, ocean_proximity...). En divisant l'ensemble de données en sous-groupes basés sur ces caractéristiques, l'algorithme de régression arborescente peut fournir une prédiction précise du prix de l'immobilier pour de nouvelles propriétés en fonction de leurs caractéristiques.
Les différents hyperparamètres de l'arbre de régression:

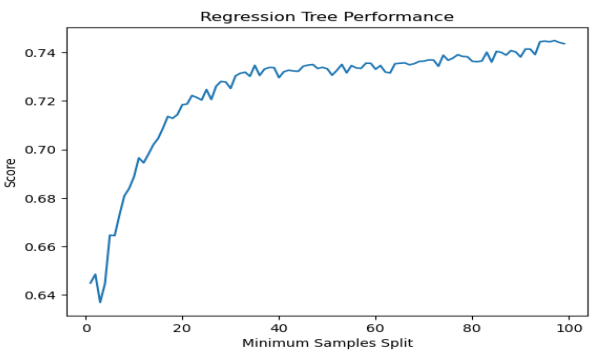
- Le nombre minimum d'échantillons requis dans une feuille
- La profondeur maximale de l'arbre
- Le nombre minimum d'échantillons requis pour diviser un noeud
- Le critère de division

Choix des hyperparamètres :

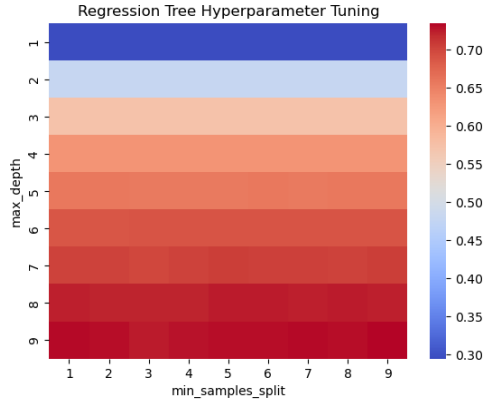
Pour les hyperparamètres nous avons choisi à travailler avec la profondeur maximale de l'arbre et le nombre minimum d'échantillons requis pour décomposer un noeud car ils sont les hyperparamètres les plus importantes à considérer lors de la construction d'un arbre de régression , ils contrôlent la complexité du modèle qui est bien une condition primordiale.
La profondeur maximale de l'arbre permet la détermination du nombre maximal de nœuds que l'arbre peut avoir. Si la profondeur maximale est trop élevée, l'arbre risque de surajuster les données d'entraînement et d'avoir une mauvaise performance sur les nouvelles données. D'un autre côté, si la profondeur maximale est trop faible, le modèle a manqué de complexité pour bien capturer les relations entre les caractéristiques et la variable cible.
Le nombre minimum d'échantillons requis pour scinder un nœud contrôle la taille minimale des échantillons à considérer pour créer une nouvelle division.

Variation du score avec l'hyper paramètre max depth

Variation du score avec l'hyper paramètre min sample split



Variation du score avec les 2 l'hyper paramètre au même temps:



la prédiction du prix de l'immobilier

```
Mean cross-validation R^2 score:  0.7145794948427415
Test R^2 score:  0.7283832198912141
3640464574.379937
```

	Prix_predits	median_house_value
0	161865.573770	112500.0
1	301068.367347	357000.0
2	188191.776316	165200.0
3	169450.934579	133500.0
4	223102.728883	280500.0
...	...	...
4032	123823.050847	139100.0
4033	282735.714286	500001.0
4034	413825.115385	500001.0
4035	144484.000000	100000.0
4036	160028.951049	158200.0

4037 rows × 2 columns

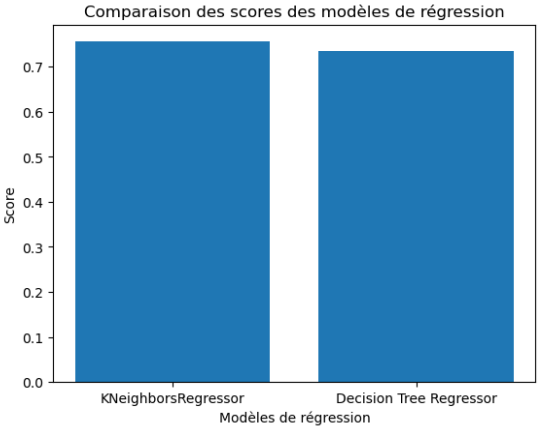
nous avons effectué le même principe précédent ,nous avons réalisé une analyse approfondie et systématique de notre modèle de régression arborescente en manipulant les hyperparamètres donc la profondeur maximale et le nombre minimum d'échantillons requis pour diviser un noeud , en évaluant l'impact de la suppression des outliers, en sélectionnant des colonnes importantes et en utilisant la validation croisée pour estimer la capacité de généralisation du modèle. Nous avons également utilisé les mêmes train set et test set pour évaluer leur impact sur les performances du modèle de régression arborescente . Finalement, nous avons tracé une courbe d'apprentissage pour observer la performance du modèle en fonction du nombre d'exemples d'entraînement utilisés.En résultat nous avons trouvé que Les meilleurs hyperparamètres sont : profondeur maximale = 9, fraction minimale de feuilles requises pour effectuer une division = 8. Le score estimé correspondant est de 0,734639395037303.

Comparaison entre les deux modèles

En machine learning, le score et la performance d'un modèle sont des mesures importantes pour évaluer l'efficacité de l'algorithme d'apprentissage automatique utilisé. En général, plus le score d'un modèle est élevé, meilleure est sa performance. Cependant, il est important de noter que la fiabilité d'un modèle dépend de plusieurs facteurs, tels que la qualité des données, la méthode d'apprentissage utilisée et les hyper paramètres choisis.
Dans notre cas, après avoir entraîné et évalué deux modèles différents, à savoir K-Neighbors et la régression arborescente, nous avons constaté que le modèle K-Neighbors était plus performant que la régression arborescente. Cette observation a été confirmée en examinant les graphes de performance des deux modèles, où nous avons constaté que le modèle K-Neighbors avait une meilleure précision et une meilleure récupération que la régression arborescente.Ces résultats suggèrent que le modèle K-Neighbors est mieux adapté pour résoudre le problème de classification en question que la régression arborescente. Il est important de noter que ces résultats ne sont valables que pour notre ensemble de données particulier et peuvent ne pas se généraliser à d'autres ensembles de données. Il est donc toujours recommandé d'évaluer plusieurs modèles différents et de

choisir le modèle le plus adapté en fonction des caractéristiques des données et des objectifs de l'application.

	Scores
KNeighborsRegressor	0.755722
Decision Tree Regressor	0.734891



Conclusion

Ce rapport décrit la création d'un modèle de prédiction du prix de l'immobilier en utilisant les méthodes K-neighbors et Regression Tree Algorithm. Des techniques telles que la suppression des outliers, la sélection des meilleures colonnes en fonction de leur corrélation, la standardisation des données et l'ajustement des paramètres essentiels ont été utilisées pour optimiser le modèle. En outre, une optimisation du poids des voisins a permis d'améliorer les performances du modèle de 0,02%. Enfin, l'utilisation du modèle Regression Tree Algorithm avec les hyperparamètres maxdepth et min sample split a été explorée pour améliorer davantage les performances de prédiction. Les résultats obtenus sont présentés en détail dans le rapport.