Ntambwe Dan

2A SIE

Rapport Projet ML

I/Exploration statistique des données et visualisations des données

L'exploration statistique des données est une étape cruciale pour comprendre les données avec lesquelles nous travaillons. Dans ce projet, après avoir chargé le jeu de données 'California Housing' et créé un DataFrame à partir de celui-ci, nous avons commencé par examiner les statistiques descriptives de base de nos données en utilisant la fonction 'df.describe()'. Cela nous donnera des informations telles que la moyenne, le minimum, le maximum et les quartiles pour chaque caractéristique.

Toutefois, avant tout affichage, nous vérifions d'abord s'il y a des valeurs manquantes dans nos données avec la fonction 'df.isnull().sum()'. Cela nous permet de remplacer des valeurs manquantes par la moyenne ou de les supprimer. Par la suite, nous réalisons une normalisation des données afin de rendre les résultats de votre modèle plus précis en nous assurant que toutes les caractéristiques sont à la même échelle. Enfin, pour mieux comprendre les relations entre les différentes caractéristiques et la variable cible, nous avons utilisé certains outils de visualisations de données : nous avons choisi de créer

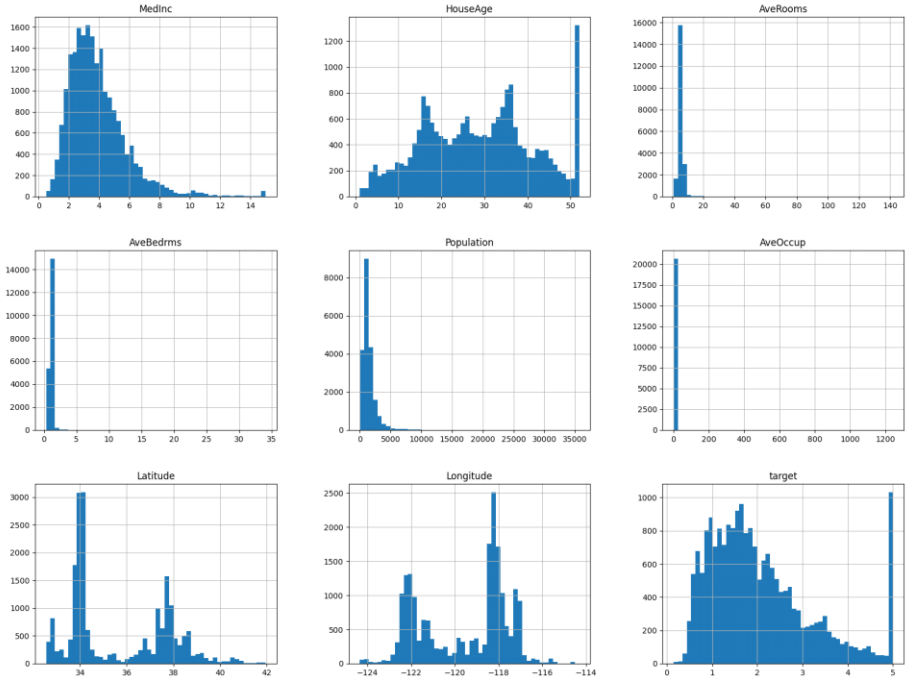
des histogrammes, des diagrammes de dispersion et la méthode 'corr()' pour voir la relation entre différentes caractéristiques et la variable cible.

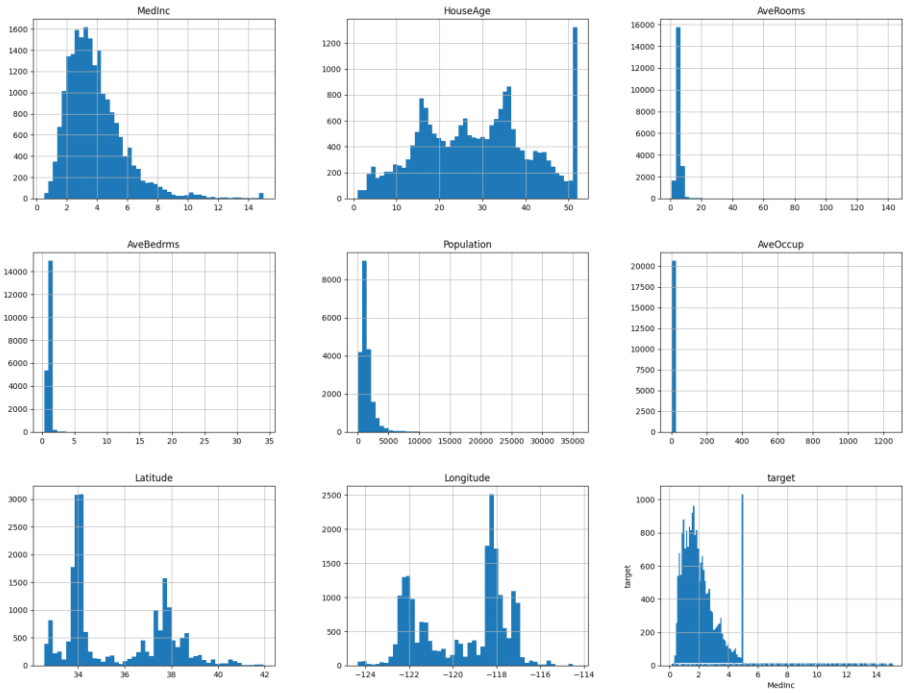
Voici les résultats:

```
EXPLORATION STATIQUE DES DONNEES
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	target
count	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	3.870671	28.639486	5.429000	1.096675	1425.476744	3.070655	35.631861	-119.569704	2.068558
std	1.899822	12.585558	2.474173	0.473911	1132.462122	10.386050	2.135952	2.003532	1.153956
min	0.499900	1.000000	0.846154	0.333333	3.000000	0.692308	32.540000	-124.350000	0.149990
25%	2.563400	18.000000	4.440716	1.006079	787.000000	2.429741	33.930000	-121.800000	1.196000
50%	3.534800	29.000000	5.229129	1.048780	1166.000000	2.818116	34.260000	-118.490000	1.797000
75%	4.743250	37.000000	6.052381	1.099526	1725.000000	3.282261	37.710000	-118.010000	2.647250
max	15.000100	52.000000	141.909091	34.066667	35682.000000	1243.333333	41.950000	-114.310000	5.000010

Nous obtenons les différentes informations statistiques par rapport à la cible.






```
Coefficients de correlation

target        1.000000
MedInc        0.688075
AveRooms      0.151948
HouseAge      0.105623
AveOccup     -0.023737
Population   -0.024650
Longitude    -0.045967
AveBedrms    -0.046701
Latitude     -0.144160
Name: target, dtype: float64
```

Via les coefficients de corrélation, nous déduisons que :

Lorsque le revenu médian des ménages augmente, le prix médian des maisons a tendance à augmenter également.

Lorsque le nombre moyen de pièces augmente, le prix médian des maisons a tendance à augmenter légèrement.

Lorsque l'âge médian des maisons augmente, le prix médian des maisons a tendance à augmenter très légèrement.

Il n'y a pas de corrélation significative entre le prix médian des maisons et la population, la longitude, le nombre moyen de chambres et de pièces par maison.

II/Mise au points et mesure des modèles de régression

Pour chaque modèle, nous avons défini une grille d'hyperparamètres à optimiser. Ces grilles sont stockées dans le dictionnaire param_grids, où chaque clé est le nom du modèle et chaque valeur est la grille d'hyperparamètres correspondante.

Par la suite, nous créons un objet GridSearchCV avec le modèle et sa grille d'hyperparamètres comme arguments. Cet objet effectue une recherche sur grille pour trouver la meilleure combinaison d'hyperparamètres pour le modèle. Le modèle est ensuite entraîné sur l'ensemble d'entraînement à l'aide de la méthode 'fit()' de l'objet GridSearchCV.

Nous avons utilisé la fonction 'train_test_split()' pour diviser les données, en attribuant 80% des données à l'ensemble d'entraînement et 20% à l'ensemble de test. Cela permet d'entraîner le modèle sur une grande partie des données tout en conservant une partie pour tester la performance du modèle sur des données qu'il n'a jamais vues auparavant.

Voici les résultats :

Les meilleurs paramètres pour chaque modèle sont affichés, ainsi que la racine de l'erreur quadratique moyenne (RMSE) et le coefficient de détermination R^2.

```
Meilleurs paramètres pour LinearRegression: {}
Calcul et affichage du coefficient de Détermination

RMSE pour LinearRegression: 0.15372748628104133
R^2 pour LinearRegression: 0.5757877060324512
```

```
Meilleurs paramètres pour Lasso: {'alpha': 0.1}
Calcul et affichage du coefficient de Détermination

RMSE pour Lasso: 0.23605188314893116
R^2 pour Lasso: -0.00021908714592466794
```

```
Meilleurs paramètres pour RandomForestRegressor: {'bootstrap': True, 'max_depth': None, 'min_samples_split': 2, 'n_estimators': 200}
Calcul et affichage du coefficient de Détermination

RMSE pour RandomForestRegressor: 0.10380460827431248
R^2 pour RandomForestRegressor: 0.8065746164045953
Resultat : le modèle RandomForestRegressor est le meilleur
```

Le RMSE étant une mesure de l'erreur de notre modèle, plus elle est faible, meilleur est le modèle. Dans notre cas, le modèle **RandomForestRegressor** a le RMSE le plus faible (0.104), ce qui signifie qu'il a la plus petite erreur parmi les trois modèles.

Le R^2 est une mesure de la qualité de l'ajustement de notre modèle. Il varie de $-\infty$ à 1, où 1 indique un ajustement parfait et une valeur de 0 ou moins indique que le modèle n'est pas meilleur qu'un modèle naïf qui prédit simplement la moyenne. Même conclusion que précemment, dans notre cas, le modèle

RandomForestRegressor a le R^2 le plus élevé (0.805), ce qui signifie qu'il explique le plus de variance dans les données parmi les trois modèles.

A noter que d'autre facteurs tels que la complexité du modèle, le temps d'entraînement,d'inférence ou même la compréhensibilité du modèle peuvent également être importants pour l'issue du résultat.

III/Poursuite avec le meilleur modèle

De l'étape précédente,nous avons conclu que le modèle RandomForestRegressor était le meilleur des 3. Nous avons défini une nouvelle grille d'hyperparamètres pour celui-ci et effectué une nouvelle recherche sur grille pour trouver la meilleure combinaison d'hyperparamètres. Le modèle a ensuite été réentraîné avec ces meilleurs hyperparamètres et utilisé pour faire des prédictions sur l'ensemble de test.

Voici les résultats:

```
Calcul et affichage du coefficient de Détermination

RMSE: 0.10412983218593061
R^2: 0.805360699117743
```

Nous constatons que :

Le RMSE a légèrement augmenté après l'optimisation, ce qui indique une légère augmentation de l'erreur de prédiction du modèle.

le coefficient de détermination R^2 a légèrement diminué après l'optimisation, ce qui suggère que le modèle optimisé explique une légèrement plus petite proportion de la variance dans la variable cible.

Cela peut être du à plusieurs raisons comme par exemple le fait est que le modèle initial était déjà assez bien ajusté et que l'optimisation des hyperparamètres n'a pas réussi à trouver une meilleure combinaison. Il est également possible que la grille d'hyperparamètres utilisée pour l'optimisation n'incluait pas les valeurs optimales pour certains hyperparamètres.

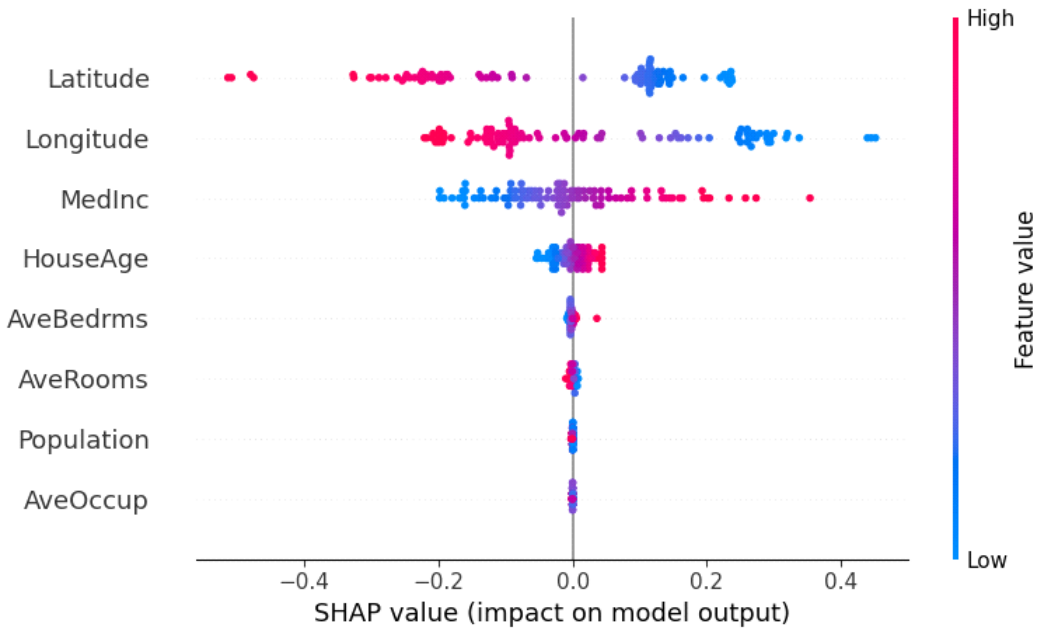
IV/Analyse de l'importance des caractéristiques dans la prédiction des prix des maisons

Nous avions prévu utilisé la bibliothèque SHAP pour analyser l'importance des caractéristiques dans la prédiction des prix des maisons. Cependant, en raison du temps de calcul important requis par SHAP, nous avons choisi d'utiliser un modèle de régression Ridge pour cette analyse.

Nous avons entraîné un modèle de régression Ridge sur l'ensemble d'entraînement permettant d'éviter le surajustement.

Nous avons créé un explainer SHAP pour le modèle Ridge. Cela nous permet d'expliquer les prédictions de n'importe quel modèle de machine learning en attribuant à chaque caractéristique une importance pour la prédiction.

Voici le résultat :



On constate que :

Le revenu médian a le plus grand impact sur le prix des maisons. Les valeurs élevées de MedInc augmentent généralement le prix prédit (points rouges principalement à droite de 0), tandis que les valeurs faibles diminuent le prix prédit (points bleus principalement à gauche de 0).

Les valeurs élevées de Latitude semblent avoir un impact négatif sur le prix prédit tandis que les valeurs faibles ont un impact positif. Cela pourrait suggérer que les maisons situées plus au sud en Californie (latitude plus faible) ont tendance à être plus chères.

De même, la longitude semble avoir un impact significatif sur le prix prédit.Cela suggére que les maisons situées plus à l'est en Californie (longitude plus élevée) ont tendance à être prédites comme étant moins chères, tandis que les maisons situées plus à l'ouest (longitude plus faible) sont prédites comme étant plus chères. Cela pourrait être dû à des facteurs géographiques et économiques, comme la proximité de l'océan Pacifique et des grandes villes.

Les autres caractéristiques ont un impact moins important sur le prix prédit.

Conclusion

En conclusion, nous avons travaillé sur la prédiction des prix des maisons en Californie en utilisant plusieurs modèles de régression. Nous avons effectué un prétraitement rigoureux des données, y compris la gestion des valeurs manquantes et la normalisation. Nous avons utilisé la recherche sur grille pour optimiser les hyperparamètres de nos modèles et évaluer leurs performances via le RMSE et le R^2. Enfin, l'utilisation de la bibliothèque SHAP nous a permis d'analyser l'importance des caractéristiques dans la prédiction des prix des maisons.

Pour une grande partie des implémentations, nous avons choisi des paramètres diminuant le temps d'exécution et par conséquent impactant la précision de nos simulations. Dans l'optique d'obtenir une meilleure précision de nos résultats, nous devrions changer nos hyperparamètres.

Pour des explorations futures, nous pourrions envisager d'essayer d'autres types de modèles de régression, comme la régression par processus gaussien ou les réseaux de neurones.