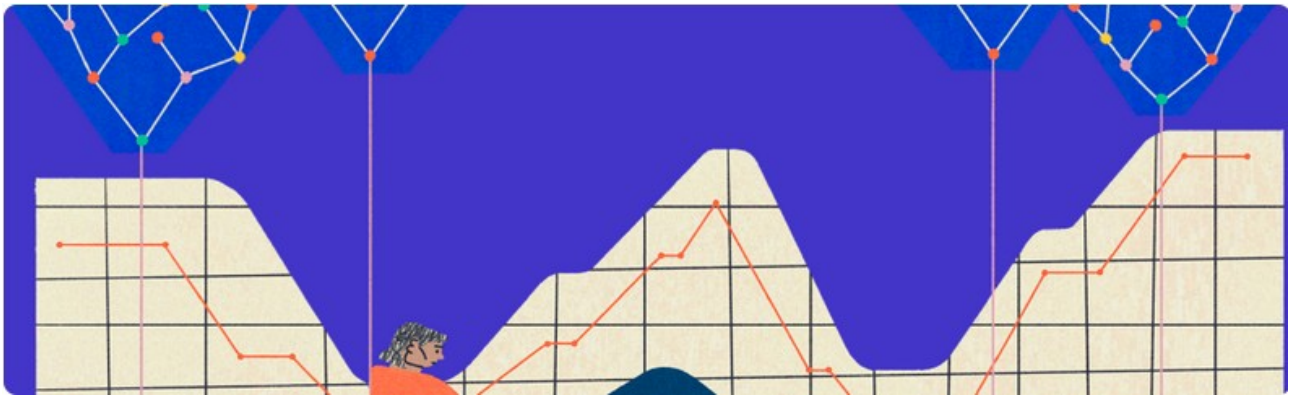


Regression - DecisionTreeRegressor RandomForestRegressor

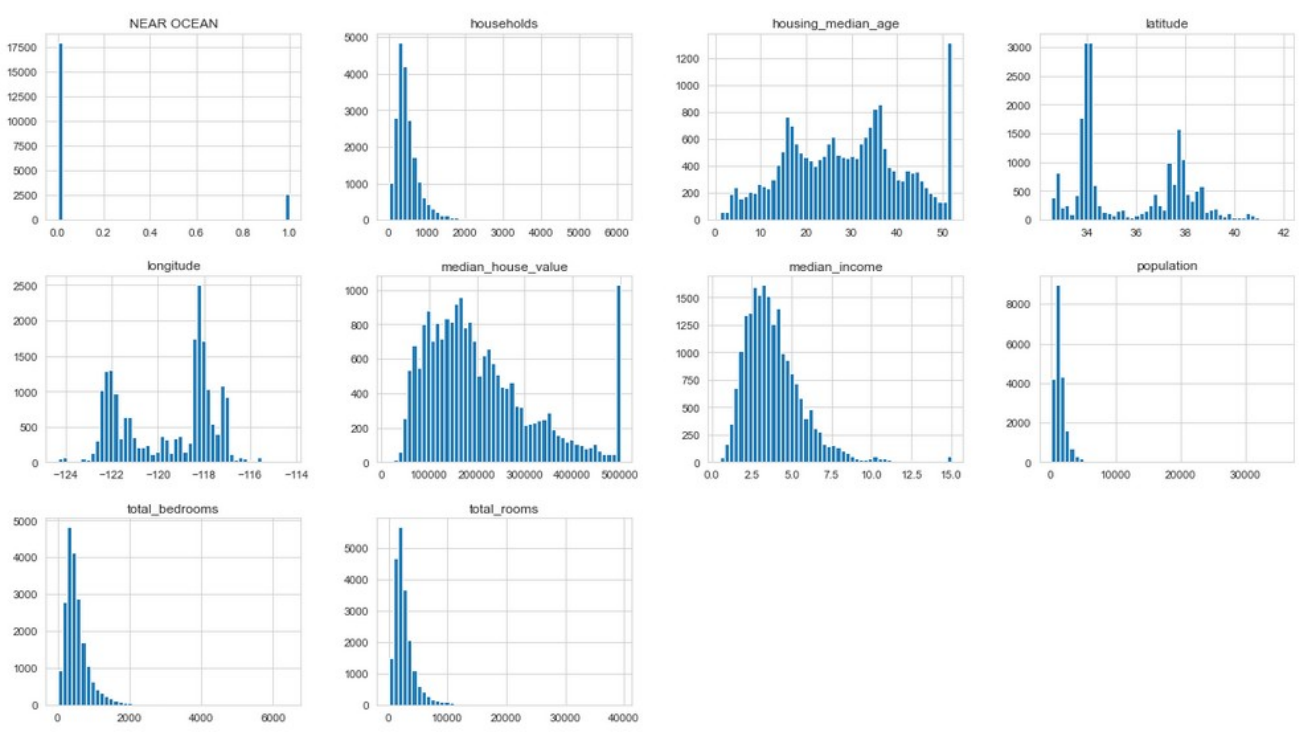
Aude Pertron

1) Préparation des données

Dans cette étape, j'ai décidé de procéder immédiatement au retraitement des données, à savoir :

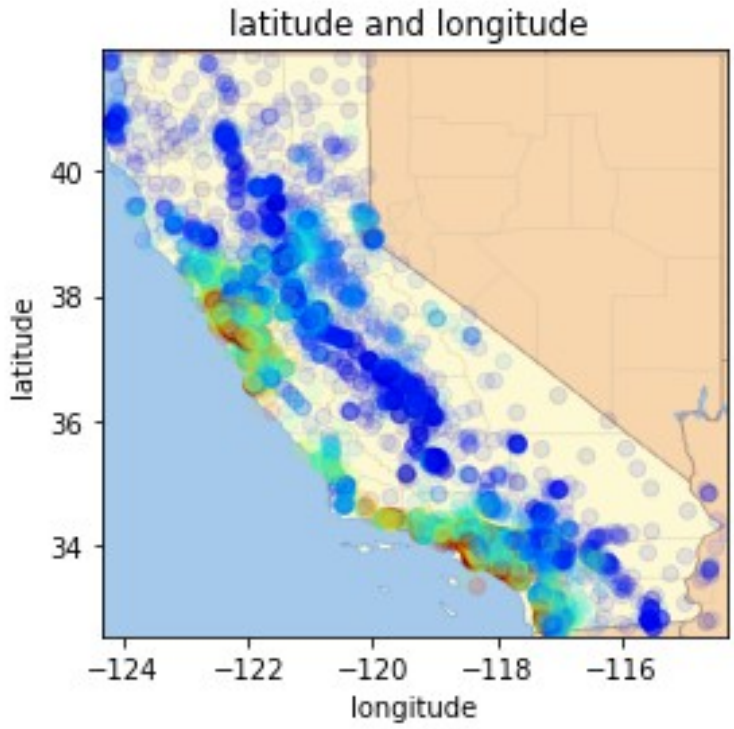
- Remplacer les valeurs nulles de la colonne total_bedrooms
- Utiliser la fonction get_dummies de pandas pour remplacer les valeurs de la colonne ocean_proximity par des booléens
- définir X et Y en utilisant la fonction drop de pandas

Ceci afin d'éviter un double traitement sur les jeux de données X_test et X_train par la suite (idem pour y_train et y_test)



Ces histogrammes permettent d'observer la distribution des données de chaque colonne. Par exemple, on peut observer que :

- la courbe de revenus est assez centrée autour de 2 à 5k
- la courbe d'âge moyen des maisons paraît étrange, avec un pic à 50. Peut-être certaines données ont-elles été entrées de manière incorrecte ?
- de même notre target, median_house_value, présente un pic à 500000



Ici on peut deviner que les maisons les plus chères sont situées sur la côte, au centre de San Francisco et Los Angeles.

median_house_value	1.000000
median_income	0.688075
<1H OCEAN	0.256617
NEAR BAY	0.160284
NEAR OCEAN	0.141862
total_rooms	0.134153
housing_median_age	0.105623
households	0.065843
total_bedrooms	0.049454
ISLAND	0.023416
population	-0.024650
longitude	-0.045967
latitude	-0.144160
INLAND	-0.484859

Plus loin, l'étude de la corrélation nous indique deux détails intéressant :

- la valeur d'une maison est liée positivement au revenu moyen du secteur
- et négativement à l'éloignement par rapport à la mer

2) Sélection, apprentissage et évaluation du modèle

La régression linéaire nous donne un score RMSE de **68717.79**
Le decision Tree regressor quand à lui à un RMSE de **0.0** quand on l'applique sur l'échantillon d'apprentissage.
Par contre, lorsque l'on revérifie ce score via la méthode des k_folds, le RMSE passe à **91577.**

Cette différence nous démontre clairement un cas d'overfitting : le premier modèle de decision tree est parfait sur l'échantillon d'apprentissage, mais performe mal sur un échantillon test.
Losque l'on applique la méthode des k_folds à la régression linéaire, le RMSE passe à **72213**. Nous avons là plutôt un problème d'underfitting, qui pourrait être dû à une quantité de données insuffisante ou à un modèle peu performant.

Entre le decision tree et la régression linéaire, **la régression linéaire est pour l'instant le meilleur choix sur nos datas,** même s'il n'est pas optimal.

3) Fine-Tunning

Les essais avec le grid_search sont moins faciles du fait de la quantité de calculs effectués, qui prennent plusieurs minutes à chaque instanciation.
J'ai pû réaliser que je n'aurais pas dû effectuer l'action reshape sur Y, cela a généré des warnings, mais le calculs se sont effectués quand même.
Le grid_shape a dégagé les paramètres optimaux :

- max features = 4
- n_estimators = 30

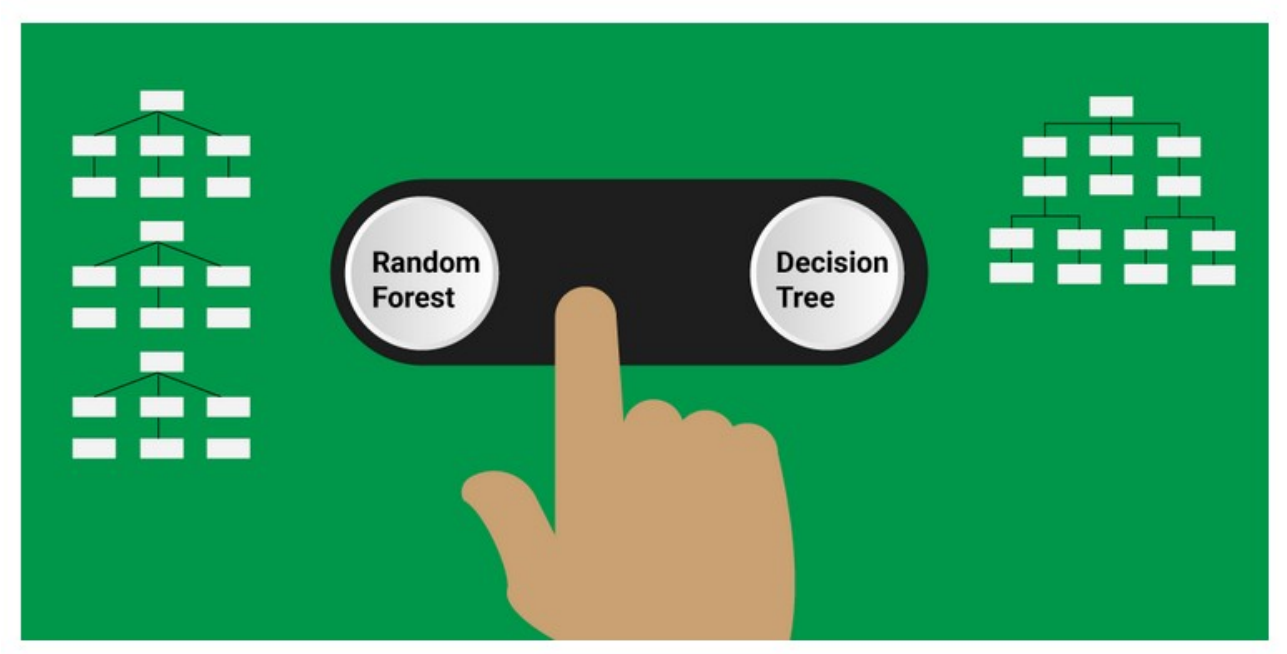
max_features correspond au nombre de features à rechercher lors de la cration d'une nouvelle branche
n_estimators correspond au nombre d'arbres

lorsque l'on s'intéresse aux différents scores obtenus lors des essais de valeurs des paramètres, on constate qu'augmenter les features finit par être contre-productif, avec un allongement du temps de calcul et une nette régression du score :

Mean Score	Parameters
80596.00	{'max_features': 2, 'n_estimators': 3}
76114.34	{'max_features': 2, 'n_estimators': 10}
72972.06	{'max_features': 2, 'n_estimators': 30}
81613.74	{'max_features': 4, 'n_estimators': 3}
73579.04	{'max_features': 4, 'n_estimators': 10}
71356.29	{'max_features': 4, 'n_estimators': 30}
79133.78	{'max_features': 6, 'n_estimators': 3}
78972.15	{'max_features': 6, 'n_estimators': 10}
72871.68	{'max_features': 6, 'n_estimators': 30}
93921.91	{'max_features': 8, 'n_estimators': 3}
80860.47	{'max_features': 8, 'n_estimators': 10}
76189.99	{'max_features': 8, 'n_estimators': 30}

Finalement, l'évaluation sur la base de test nous rend un score RMSE de **18828.**
Ce résultat est surprenant considérant que la performance sur le training set était de **71356.**
Cela nous indique qu'un erreur s'est possiblement glissé dans notre évaluation.
Faute de temps nous devrons en rester là.

4) Résumé sur Decision Tree et random forest :



Une random forest est un ensemble aléatoire de decisions trees, ce qui rend son résultat en général plus cohérent avec l'ensemble des données. Cependant, elle nécessite beaucoup plus de calculs et son temps d'entraînement est donc plus long. Son analyse est également souvent plus complexe, ce qui incite beaucoup de personnes à utiliser les decisions trees à la place.

Random forest		**Decision tree**	
+	-	+	-
Pertinence des résultats	Long et complexe à entraîner	Facile d'utilisation	Très dépendant du jeu de données
Moins sensible au jeu de données	Plus complexe à analyser	Analyse facile	Résultats parfois peu pertinents