Brief : prédire le prix d'une maison

Vous êtes chargé de développer un algorithme qui pourra prédire le prix d'une maison en fonction d'autres paramètres, comme la surface en m², le nombre de chambres, etc.

Avant de faire cette modélisation, vous devez effectuer une phase d'exploration de vos données (**EDA** = **E**xploratory **D**ata **A**nalysis). Cette phase vous permettra de mieux comprendre vos données.

Après l'EDA, vous pouvez passer à la phase suivante, la préparation de vos données pour le machine learning. (feature scaling, data transformation, etc.)

Quand vos données seront prêtes vous pourrez entraîner votre modèle. En utilisant les métriques d'évaluation comme le R^2 et le *MSE* vous pourrez faire des aller-retour entre l'entraînement et le *feature engeneering* afin d'améliorer vos résultats.

Enfin abordez la régression linéaire du côté statistique (validation des hypothèses, analyse des résultats, etc.)

Vous trouverez le jeu de données à ce lien

Pour ce projet vous aurez besoin de certains package Python :

- Pandas pour charger vos données
- Un package pour visualiser vos données (matplotlib, seaborn, plotly)
- Scikit-learn pour la partie modélisation (voir cette série de vidéo pour en apprendre plus sur ce package incontournable, ne pas hésiter à explorer la documentation qui est très bien)
- Statsmodel pour la régression linéaire du point de vue statistique

Etape 1 : Explorer et nettoyer vos données

Afin d'en apprendre plus sur l'exploration des données, je vous invite à suivre ce cours :

Variable	Description
Id	Unique ID for each home sold
Date	Date of the home sale
Price	Price of each home sold
Bedrooms	Number of bedrooms
Bathrooms	Number of bathrooms, where .5 accounts for a room with a toilet but no shower
Sqft_living	Square footage of the apartments interior living space
Sqft_lot	Square footage of the land space
Floors	Number of floors
Waterfront	A dummy variable for whether the apartment was overlooking the waterfront or not
View	An index from 0 to 4 of how good the view of the property was
Condition	An index from 1 to 5 on the condition of the apartment,
Grade	An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design
Sqft_above	The square footage of the interior housing space that is above ground level
Sqft_basement	The square footage of the interior housing space that is below ground level
Yr_built	The year the house was initially built
Yr_renovated	The year of the house's last renovation
Zipcode	What zipcode area the house is in
Lat	Lattitude
Long	Longitude
Sqft_living15	The square footage of interior housing living space for the nearest 15 neighbors
Sqft_lot15	The square footage of the land lots of the nearest 15 neighbors

details sur les décimals dans la colonne bathrooms

En apprendre plus et ici

Etape 2 : Préparer vos données pour la modélisation

- Séparer vos données en *train*, *validation*, *test sets*
- Créer des nouvelles variables
- Normaliser vos données
- Transformer vos données si nécessaire
- Etc.

En savoir plus

Etape 3 : Entrainer un algorithme de régression linéaire

- Essayer plusieurs variantes de la régression linéaire (Ridge, Lasso)
- Faites varier vos hyperparamètres avec un grid search.
- Vérifier que votre algorithme apprend à l'aide de la courbe d'apprentissage
- Utiliser le MSE et R^2 pour évaluer les différentes versions de votre modele

Etape 4 : Aller plus loin dans la régression linéaire

Vous pensez avoir tout compris sur la régression linéaire ? On va y ajouter un peu de rigueur mathématiques en l'abordant désormais sous un angle de vue statistique grâce à ce cours.

A propos de ce cours :

Attention il y a des différences en termes de notations. (*n* pour le nombre d'échantillons à la place de *m*, *beta* à la place de *theta* pour les coefficients).

De plus, les informations sous les titres "aller plus loin" sont assez complexes, pas de panique si vous ne saisissez pas tout.

Implémenter ces concepts dans votre modélisation (ajouter une partie "analyse des résultats")

Livrable :

- un github avec un jupyter notebook
- Présentation de votre notebook

Modalités :

- Vous disposez de temps pour effectuer ce projet, profitez-en pour vous former en parrallèle grâce aux ressources fournies (et aux autres que vous trouverez)
- Durée : 4 à 5 jours.

Pas assez de ressource ?

- Kaggle du projet (Explorer vous même le jeu de données avant de regarder les autres notebooks)
- Machine learning (Très bon cours sur le machine learning mais les projets sont en Octave)