Brief : Développez une application pour 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.

Vous trouverez le jeu de données sur ce lien.

Ce dataset porte sur les ventes de maison dans le conté de King dans l'état de Washington aux États-Unis entre 2014 et 2015.

Planning :

Jérémy	Jérémy	Antoine	Antoine	Antoine
lundi	**mardi**	**mercredi**	**jeudi**	**vendredi**
cours	Analyse des données		Modélisation	
Jérémy	Jérémy	Antoine	Antoine	
lundi	**mardi**	**mercredi**	**jeudi**	**vendredi**
Retour sur la première partie / modélisation / application	Application / déploiement		Soutenance / rex	*Vacances*

Pour ce projet vous aurez besoin de certains packages 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 vidéos pour en apprendre plus sur ce package incontournable, ne pas hésiter à explorer la documentation qui est très bien)

Partie 1 : Analyser et modéliser des données

Pour cette partie, je vous conseille de travailler avec 3 notebooks différents :

- data_cleaning_analysis.ipynb
- data_preparation.ipynb
- data_modelisation.ipynb

Pour passer vos données d'un notebook à l'autre vous pouvez utilser les méthodes to_csv() et read_csv(). Par exemple, à la fin du notebook numéro 1, vous exportez vos données à l'aide de la méthode to_csv() et ensuite vous importez vos données au début du notebook numéro 2 avec la fonction read_csv().

Etape 1 : Explorer et nettoyer vos données

Quelques tâches de nettoyage :

- Valeur manquante / dupliquée
- Simplifier le nom des colonnes
- Harmoniser les valeurs
- Est-ce que toutes les valeurs ont un type de donnée pertinent ? Parfois, une feature qui contient des chiffres doit-être transformé dans un autre type de données. Une question que j'aime me poser est : "Est-ce qu'il est pertinent de calculer la moyenne de cette feature ?"
- etc.

L'exploration consiste à comprendre les données sur lesquelles on travaille en répondant à ces questions :

- Quelles sont les caractéristiques principales du jeu de données ?
- A quoi correspondent les features ?
- Quelles valeurs prennent chaque feature ?
- Quelles sont les corrélations entre les variables ?
- Quelles seront les features utiles pour mon modèle ?
- Etc.

Pour explorer vos données, vous pouvez utiliser des fonctions, des graphiques, des tableaux, etc.
Une bonne idée est d'utliser la latitude et la longitude pour positionner des *data points* sur une carte. En les combinant avec une troisième dimension comme le prix, il est possible de créer une visualisation qui nous permet de bien comprendre nos données.

Afin d'en apprendre plus sur l'exploration des données, je vous invite à suivre ce cours et lire cet article de toward data science.

Pour mieux comprendre vos données, vous pouvez vous appuyer sur ces descriptions de variables :

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

Pourquoi la colonne bathrooms peut prendre des décimals ?

Etape 2 : Préparer vos données pour la modélisation

Dans cette deuxieme partie vous allez apprendre à préparer vos données pour les utiliser dans un algorithme de machine learning.

Voici quelques exemples de préparation :

- 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 : Entraîner un algorithme de régression linéaire

Une fois que vos données sont nettoyées et préparées, vous pouvez passer aux choses sérieuses : entraîner un modéle de machine learning.

Il existe un package très populaire pour faire du machine learning avec python : scikit-learn. Vous pouvez découvrir les capacités de ce package avec cette vidéo.

Pour cette étape, je vous conseille de vous limiter aux différentes variantes de l'algorithme de régression linéaire (ridge, lasso et elasticnet).

Malgré une sélection restreinte d'algorithme, il y a beaucoup de marge de manœuvre, notamment avec votre pipeline de données. Je vous conseille également de creuser l'étape de polynomial features.

Il est toujours intéressant de créer un point de comparaison avec un modèle naïf. (Voir *dummy regressor* dans la documentation de sklearn.)

Certains de ces algorithmes auront des hyperparamètres que le modèle ne peut pas optimiser par lui-même. Afin de trouver les paramètres optimaux, vous pouvez utiliser le **GridSearchCV** et le **RandomizedSeachCV.**

Pour vérifier que votre algorithme apprenne correctement vous pouvez faire les graphiques des courbes d'apprentissage.

Le domaine du machine learning est peuplé de métriques diverses et variées. Pour comparer et sélectionner le meilleur modèle parmi ceux que vous entrainer, je vous conseille d'utilser le MSE (RMSE) et surtout le R^2 aussi appelé le coefficient de détermination. (vidéo pour en savoir plus sur les métriques)

Etape 4 : Analyse des résultats

Cette étape consiste à mesurer l'impact négatif que peuvent avoir certaines observations de notre jeu de données.

En vous appuyant sur ce chapite de cours

Partie 2 : Créer une application qui utilise votre modèle

Utiliser votre modélisation pour créer une **application web qui permet de prédire le prix des maisons dans le conté de King.**

Pour cela vous utiliserez le package Streamlit qui permet de créer des applications web très rapidement. Cet outil ne permet pas de personnaliser aussi finement un application web que d'autres packages comme Django ou Flask mais il permet de créer des petites applications rapidement. C'est très pratique pour présenter votre modélisation en un temps record à un client. Ici on parle plutot de développement Quick and Dirty.

Le design le plus simple pour votre application est de créer un formulaire où chaque champs correspond à une variable utilisée par votre modéle (comme les m² par exemple).

Concernant le fonctionnement de votre application, l'approche la plus simple est de charger votre modéle directement dans application streamlit. Une approche plus avancée est de créer une API pour votre modèle (avec Flask ou FastAPI par exemple).

Quand vous aurez une application qui fonctionne en local, il faudra déployer votre application. Pour cela je vous conseille d'utiliser Heroku (rechercher un tuto pour utiliser heroku).

Livrable :

- Un github avec les parties 1 et 2. Vous pouvez également rendre deux github distincts.
- présentation du projet

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)

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)

Modalités de la présentation :

- 3 minutes : Introduction du jeu de données, nettoyage effectuée
- 5 minutes : Exploration du dataset
- 10 minutes : Modélisations (feature engineering, data leakage, choix du modèle, pipeline, scores)
- 3 minutes : Démonstration de votre application
- 2 minutes : Conclusion, pistes d'amélioration, difficultés rencontrées
- 5 minutes : Questions-réponses