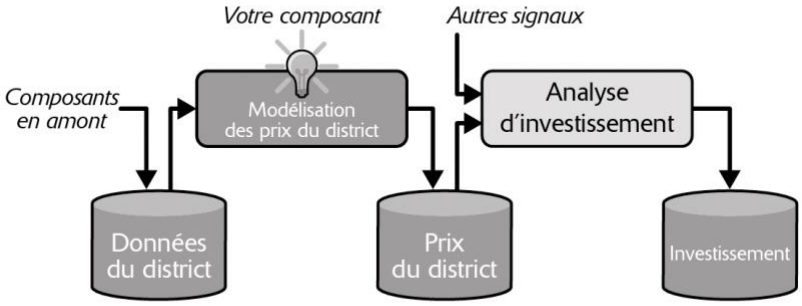
Ynov Toulouse
2023 – 2024
M2 DATA

Mettre en place un projet d'apprentissage de données de bout en bout

Dans ce TP, vous allez suivre de bout en bout un exemple de projet, pour lequel nous supposerons qu'une compagnie immobilière vient de vous embaucher comme expert en analyse de données. Cet exemple de projet est fictif, le but étant d'illustrer les principales étapes d'un projet d'apprentissage automatique, et non d'apprendre quoi que ce soit sur le marché de l'immobilier.

Objectif. La sortie de votre modèle (une prédiction du prix médian des habitations dans un district) servira à alimenter un autre système d'apprentissage automatique
Ce système en aval utilisera cette information, ainsi que d'autres signaux/informations, pour déterminer s'il est intéressant ou non d'investir dans un district donné.



Voici les principales étapes que vous allez parcourir :

1) Prendre du recul pour une vision d'ensemble

- Quel est l'objectif professionnel ?
- Construire un modèle n'est probablement pas le but ultime.
- Comment l'entreprise compte-t- elle utiliser ce modèle et en tirer parti ?
- La sortie de votre modèle (une prédiction du prix médian des habitations dans un district) servira à alimenter un autre système d'apprentissage automatique
- Quel existant : Existe-t-il des références sur lesquelles je puis m'appuyer ?
- Ce système en aval utilisera cette information, ainsi que d'autres signaux/informations, pour déterminer s'il est intéressant ou non d'investir dans un district donné

Ynov Toulouse
2023 – 2024
M2 DATA

2) Récupérer les données

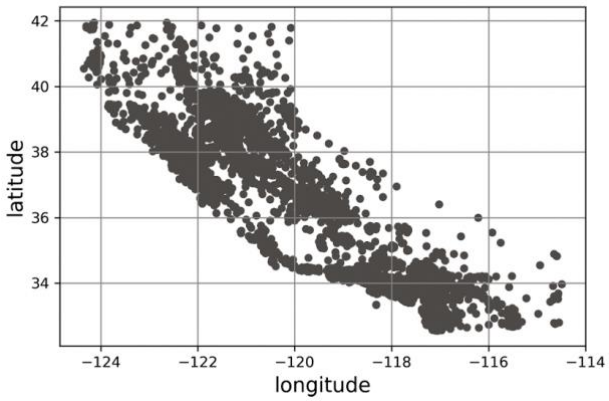
- Nous utiliserons le jeu de données California Housing Prices (prix immobiliers en Californie). Ce jeu de données est constitué de données du recensement californien. **Le jeu de données est à récupérer sur moodle.**

- Dans la plupart des environnements, les données nécessaires sont disponibles dans une base de données relationnelle ou NoSQL, et réparties entre plusieurs tables/documents/fichiers. Vous devrez d'abord vous familiariser avec l'organisation de ces données.

- Plutôt que de télécharger et de décompresser manuellement vos données, il est d'ordinaire préférable d'écrire une fonction automatisée pour réaliser cette opération. C'est particulièrement utile si les données sont très mouvantes. Automatiser le processus de récupération des données est également utile si vous devez les installer sur plusieurs machines

3) Explorer et visualiser les données pour mieux les comprendre

1) Visualisation des données
   - Étant donné qu'il y a des informations géographiques (latitude et longitude), il est judicieux de créer une représentation à deux dimensions de tous les districts pour visualiser leur
     emplacementhous.plot(kind="scatter",x="longitude", y="latitude", grid=True) plt.show()



2) Étant donné que le jeu de données est de taille raisonnable, vous pouvez aisément calculer le coefficient de corrélation (appelé également r de Pearson) entre chaque couple de variables en utilisant la méthode corr()

```
>>> corr_matrix["median_house_value"].sort_values(ascending=False)
median_house_value    1.000000
median_income         0.688380
total_rooms           0.137455
housing_median_age    0.102175
households            0.071426
total_bedrooms        0.054635
population           -0.020153
longitude            -0.050859
latitude             -0.139584
Name: median_house_value, dtype: float64
```

3) Une autre façon de vérifier les corrélations entre variables consiste à utiliser la fonction *scatter_matrix* de Pandas qui croise deux à deux toutes les variables quantitatives (c .-à-d . numériques)

4) Préparer les données pour les algorithmes d'apprentissage automatique.

4.1) Préparer les données pour vos algorithmes d'apprentissage automatique peut devenir une opération fastidieuse. Au lieu de le faire manuellement, vous pourriez écrire des fonctions, cela a de nombreux avantages :

- cela vous permettra de reproduire aisément ces transformations sur chacun des jeux de données (c'est-à-dire chaque fois que vous actualiserez vos données),
- vous construirez peu à peu une bibliothèque de fonctions de transformation que vous pourrez réutiliser dans vos futurs projets,
- vous pourrez utiliser ces fonctions sur votre système en production pour transformer de nouvelles données avant de les fournir à vos algorithmes,
- vous pourrez essayer des transformations diverses et voir quelle combinaison de transformations convient le mieux

4.2)

- Pensez à séparer les prédicteurs et les étiquettes : on n'applique pas forcément nécessairement appliquer les mêmes transformations aux variables explicatives et à la variable à expliquer.

4.3) Nettoyer les donnéees

- La plupart des algorithmes d'apprentissage automatique ne supportent pas les valeurs manquantes : créer une fonction pour corriger les valeurs manquantes ; bous avez en général, 3 options :
  - Les suuprimer. Exemple, vous débarrasser des districts correspondants,
  - supprimer cette variable,

- remplacer les données manquantes par une valeur (zéro, la moyenne, la médiane, etc .) . C'est ce qu'on appelle l'imputation

Le package sklearn.impute comporte également d'autres classes d'affectation de données manquantes plus puissantes (toutes deux s'appliquant uniquement aux variables quantitatives) :

– KNNImputer remplace chaque valeur manquante par la moyenne des valeurs de cette variable pour les k plus proches voisins (en anglais k nearest neighbors ou KNN). La distance est calculée sur l'ensemble des variables disponibles.

– IterativeImputer applique d'abord un modèle de régression à chaque variable pour prédire les valeurs manquantes en se basant sur toutes les autres variables disponibles. Puis il applique à nouveau le modèle aux données ainsi mises à jour et répète le processus plusieurs fois, améliorant les modèles et les valeurs de remplacement à chaque itération.

4.4) Gérer des variables qualitatives

- Penser à examiner également les valeurs qualitatives. Gérer des variables qualitatives (combien avez-vous dans l'exemple ?)
- La plupart des algorithmes d'apprentissage automatique préfèrent travailler sur des nombres, alors penser à convertir ces chaînes de caractères en nombres . Pour cette tâche, vous pouvez utiliser la classe OrdinalEncoder de Scikit-Learn

4.5) Recalibrage et transformation des variables (normalisation des données)

- L'une des transformations les plus importantes que vous devez appliquer à vos données est le recalibrage , encore appelé réduction ou changement d'echelle , des variables (en anglais, feature scaling) . À peu d'exceptions près, les algorithmes d'apprentissage automatique ne fonctionnent pas très bien lorsque les variables numériques en entrée ont des échelles très différentes

5) Sélectionner un modèle et l'entraîner.

- Vous avez cerné le problème, vous avez récupéré les données et les avez explorées, vous en avez extrait un jeu d'entraînement et un jeu de test, et vous avez programmé des pipelines de transformation pour nettoyer et préparer automatiquement vos données pour les algorithmes d'apprentissage automatique. Vous êtes maintenant prêt à sélectionner et à entraîner un modèle de Machine Learning

6) Régler avec précision votre modèle.

- Recherche par quadrillage
  - Vous pouvez utiliser GridSearchCV de Scikit-Learn qui cherchera pour vous les meilleures combinaisons pour trouver les meilleurs hyperparamètres (cela évite aussi de faire manuellement cette recherche/optimisation). Vous devez simplement lui dire quels sont les hyperparamètres que vous souhaitez faire varier et quelles valeurs utiliser, et il évaluera pour vous toutes les combinaisons possibles des valeurs des hyperparamètres par validation croisée

7) Présenter votre solution.
8) Lancer, surveiller et maintenir votre système