

Intelligence Artificielle (I-ILIA-029) -- Travaux Pratiques

TP 01 : Machine Learning : prétraitement, analyse et visualisation des données

• Introduction :

L'objectif de ce TP est de vous familiariser avec les librairies de Machine Learning (Pytorch, Scikit learn, etc.) sous Python. Vous serez amené à résoudre un problème de régression linéaire pour la prédiction des prix de logements en Californie. Nous utilisons pour cela une base de données publique qui contient des données concernant les maisons situées dans un district californien. Il est important de savoir que les données ne sont pas nettoyées, des étapes de pré-traitement sont donc nécessaires. Ce TP présente est composé de quatre parties :

1. **Partie 1** : prise de main avec l'environnement et se familiariser avec pandas (**TP1**) ;
2. **Partie 2** : préparation et analyse des données (**TP1**) ;
3. **Partie 3** : création des modèles de régression (avec et sans Deep Learning) (**TP2**) ;
4. **Partie 4** : évaluation du modèle avec des données de test : **TP2**.

1. Partie 1 : prise en main avec l'environnement :

1.1. connexion à l'environnement : dans un premier temps, il faudra se connecter sur la plateforme Google Colab (<https://colab.research.google.com/>). La connexion Google est obligatoire.

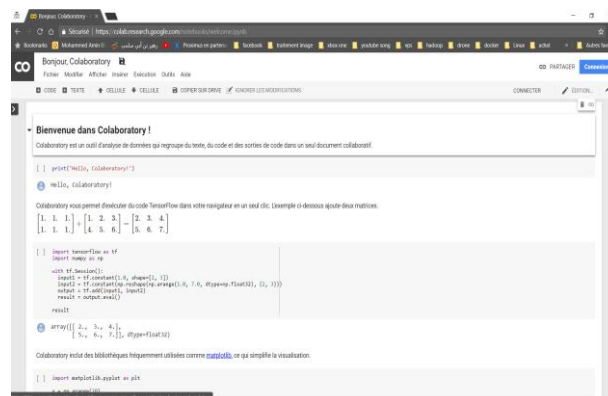


Figure 1 : La page d'accueil de Google Colab

1.2. Sélection du matériel HPC (GPU) : afin d'utiliser le processeur graphique GPU qui offre des accélérations très intéressantes (en temps de calcul), il suffit de sélectionner le GPU (carte graphique) en allant sur : Exécution -> Modifier type d'exécution -> Accélérateur matériel -> GPU (Figure 2).

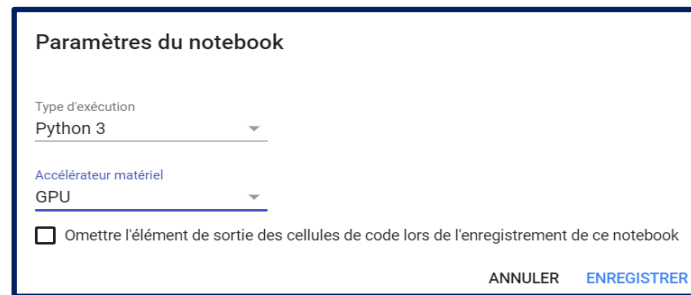


Figure 2 : Choix du type d'exécution

- Vérifier que le GPU est bien sélectionné à l'aide de la commande « *nvidia-smi* »

1.3. Importation du code démarrage : importer le code de démarrage (à compléter) partagé via Moodle « **Input_TP1_2025.ipynb** » en allant sur : Fichier -> Importer le Notebook

1.4. Création de DataFrame : une fois l'environnement *Colab* configuré, exécuter la cellule avec les code suivant :

```
import pandas as pd

# création d'un dictionnaire python de données suivants
data = {
    "id": [1, 2, 3, 4],
    "ville": ["Bruxelles", "Liège", "Namur", "Mons"],
    "population": [196828, 195778, 114042, 96358],
    "superficie_km2": [33.09, 68.65, 175.93, 147.56]
}

# le transformer en DataFrame
df = pd.DataFrame(data)
```

1.5. Exploration des statistiques : Utiliser *.head()* pour afficher un aperçu du DataFrame et *.describe()* pour analyser les statistiques descriptives des villes.

1.6. Tri des données : Trier le DataFrame par ordre décroissant par population puis par superficie en utilisant la fonction « *sort_values* » et afficher le résultat.

1.7. Visualisation des données : Utiliser la librairie *Seaborn* pour afficher un graphique en barres horizontales représentant la population des villes.

2. Partie 2 : préparation et analyse des données

2.1. Importation des librairies :

- **Question 1** : exécuter la cellule d'importation des librairies (Sk-learn, pandas, numpy, etc.)

2.2. Téléchargement et préparation des données : avant de procéder à l'apprentissage, il est important de bien préparer et nettoyer les **données**.

- **Question 2** : télécharger les données à l'aide de la librairie pandas et fonction ***pd.read_csv***

Note : les données (en extension csv) sont fournies via ce lien :

https://download.mlcc.google.com/mledu-datasets/california_housing_train.csv.

- **Question 3** : quelle est la taille de cette base de données ? (Utiliser la fonction « ***shape*** »)
- **Question 4** : afficher les 30 premières lignes de la base de données avec la fonction « ***head*** »
- **Question 5** : compléter la cellule pour vérifier la présence de valeurs nulles avant suppression
- **Question 6** : que constatez-vous ?
- **Question 7** : analyser et exécuter la fonction permettant de sélectionner les données d'apprentissage (***X***) et calculer une nouvelle caractéristique (« ***rooms per person*** »)
- **Question 8** : compléter la fonction de définition des données cibles (***Y*** = prix réel médian)
- **Question 9** : exécuter la cellule permettant de sélectionner les données d'apprentissage (12000 enregistrements)
- **Question 10** : sélectionner les derniers 5000 enregistrements pour validation (« ***tail*** »)
- **Question 11** : visualiser les données (training/validation) avec la fonction « ***data_visualization*** »

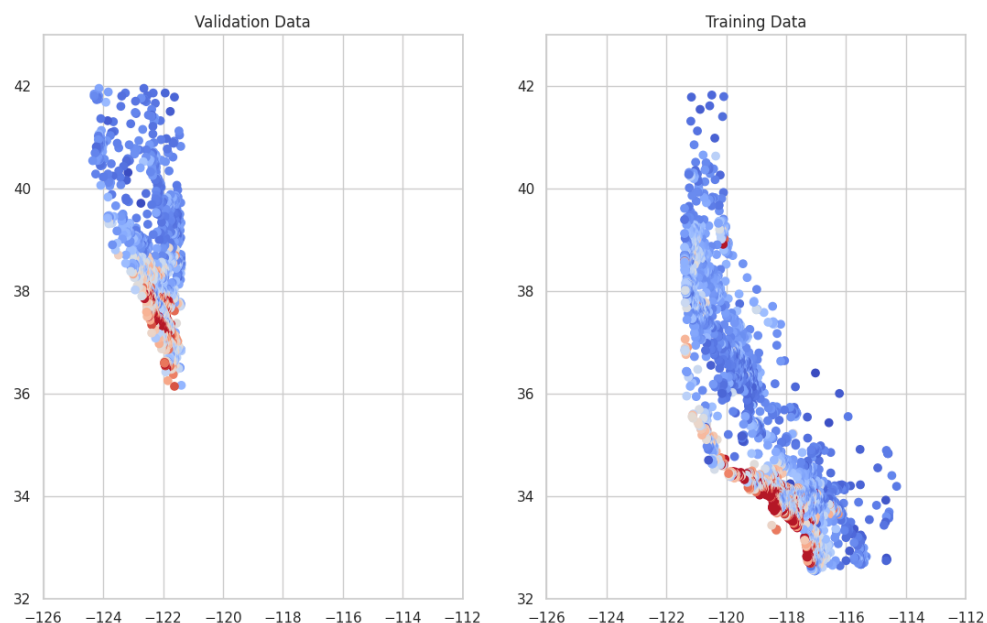


Figure 4 : Visualisation des données (entraînement et validation)

- **Question 12 :** que constatez-vous ? adaptez les données et visualisez de nouveau.
- **Question 13 :** Appliquer la méthode de corrélation « **Pearson** » sur les données (en utilisant les cartes « **heatmap** » de la librairie **Seaborn**)
- **Question 14 :** Analyser et interpréter les résultats

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}}$$

Figure 5: Formule de calcul de coefficient de corrélation de Pearson

- **Question 15 :** Afficher la distribution de variable cible "**median_house_value**" en utilisant **sns.distplot**
- **Question 16 :** Afficher le classement décroissant des valeurs corrélés avec la cible « **median house value** »
- **Question 17 :** Visualiser les données sur la carte avec la fonction « **Visualize_on_map** »

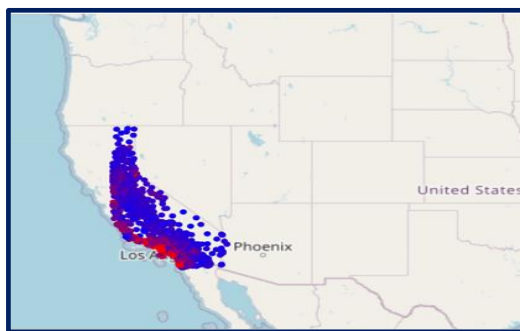


Figure 6 : Données d'entrainement

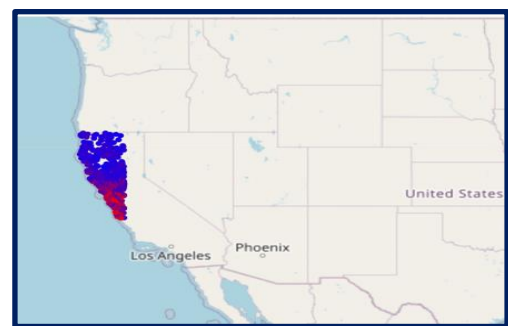


Figure 7 : données de validation

Note : le résultat de ce TP (données prétraitées) sera utilisé comme point de départ pour le prochain

TP : TP2. Veuillez télécharger et sauvegarder vos notebooks.