Université Abdelmalek Essaâdi
Faculté des Sciences et Techniques d'Al-Hoceima
Département de Mathématiques et Informatique

Filière : LST- IDLL
Module : Intelligence artificielle
A .U : 2023/2024
Pr : ZANNOU Abderrahim

Version 1

TP N°1
Énoncé

L'objectif de ce TP est de réaliser la phase de **la préparation de données** (Plus précisément **la transformation de données**) telles que :

- ✓ **Nettoyage de données**
- ✓ **Normalisation de données**
- ✓ **Encodage des variables catégoriques**
- ✓ **Réduction des attributs**

Environnements

Avec la distrbution Anaconda, on peut utiliser plusieurs IDE (environnements de développement intégrés) pour programmer en **Machine Learning (ML)** et **Deep Learning (DL)**.

Voici quelques-uns des IDE couramment utilisés pour ces tâches :

1. **JupyterLab :** C'est une interface utilisateur plus avancée que *Jupyter Notebook*. Il offre une expérience de développement plus riche avec une meilleure gestion des fichiers, des extensions personnalisables et une interface utilisateur plus moderne. *JupyterLab* est également adapté pour le développement en *machine learning et deep learning.*

2. **Jupyter Notebook :** C'est un environnement de développement interactif largement utilisé dans le domaine du *machine learning* et du deep learning. Il permet d'écrire et d'exécuter du code Python dans des cellules, ce qui facilite l'expérimentation, la visualisation des résultats et le partage du code.

3. **Spyder :** C'est un IDE spécialement conçu pour la science des données. Il offre une interface utilisateur conviviale avec un éditeur de code, une console IPython intégrée et des outils de débogage. *Spyder* est bien adapté pour le développement en *machine learning* et *deep learning* grâce à ses fonctionnalités de complétion automatique, de visualisation de données et de gestion de projets.

4. **PyCharm :** C'est un IDE Python très populaire développé par JetBrains. Il offre des fonctionnalités avancées pour le développement en *machine learning* et *deep learning*, notamment une complétion automatique

intelligente, un débogueur intégré, des outils de refactoring et une intégration avec des frameworks populaires tels que *TensorFlow* et *PyTorch*.

Ces IDE sont tous inclus dans la distribution *Anaconda* et peuvent être installés et utilisés selon les préférences. Chacun a ses propres avantages et fonctionnalités.

Dans ce TP, on utilisera l'IDE JupyterLab

Bibliothèques à utiliser

1. **NumPy :** C'est une bibliothèque fondamentale pour la manipulation de tableaux multidimensionnels et le calcul numérique en Python. Elle fournit des structures de données efficaces pour les tableaux, ainsi que des fonctions mathématiques et algébriques pour effectuer des opérations sur ces tableaux. NumPy est largement utilisé en science des données pour le traitement des données et les calculs numériques.
2. **Pandas :** C'est une bibliothèque puissante pour la manipulation et l'analyse des données en Python. Elle offre des structures de données flexibles et performantes, telles que les DataFrame, qui permettent de manipuler et d'analyser facilement des jeux de données tabulaires. Pandas est couramment utilisé pour le nettoyage, la transformation et l'exploration des données avant l'analyse et l'apprentissage automatique.
3. **Matplotlib :** C'est une bibliothèque de visualisation de données en Python. Elle permet de créer une grande variété de graphiques, tels que des graphiques linéaires, des histogrammes, des diagrammes en boîte, des diagrammes de dispersion, etc. Matplotlib est souvent utilisé pour visualiser les données et les résultats d'analyse de manière claire et informative.
4. **scikit-learn (sklearn) :** C'est une bibliothèque d'apprentissage automatique en Python qui propose une large gamme d'algorithmes et d'outils pour la classification, la régression, le regroupement, la réduction de dimension, la sélection de modèles, etc. Elle offre une interface conviviale pour appliquer ces algorithmes et évaluer les performances des modèles d'apprentissage automatique.

Ces bibliothèques sont toutes open source, bien documentées et largement utilisées dans la communauté de la science des données. Elles offrent des fonctionnalités essentielles pour la manipulation, l'analyse, la visualisation et l'apprentissage automatique des données.

L'organisation de l'environnement de travail

- Créer un dossier nommé « *Intelligence artificielle* », dans ce dossier créez deux dossiers nommés « *Machine Learning* » et « *Deep Learning* »
- Dans le dossier « *Machine Learning* », créez deux dossiers nommés « *TPs*» et « *Comptes rendus* »
  - ✓ Dans le dossier « *TPs*», créez un dossier nommé « *TP1* » qui contiendra les fichiers python de ce TP1 et un autre dossier nommé « *datasets* » qui contiendra les base de données à utiliser dans ce TP1
- Même arborescence de « *Machine Learning* » s'applique au dossier « *Deep Learning* »

Tâches à réaliser

1. **Nettoyage de données :** Gérer les données manquantes de la base de données « **achat** ».

   Nommer le fichier en : **TP1_1_Nettoyage_Achat.ipynb**
2. **Normaliser les donnée**s en utilisant la méthode **Min-Max** de la base de données « **salaire** ».

   Nommer le fichier en : **TP1_2_Normalisation_Min_Max_Salaire.ipynb**
3. **Normaliser les données** en utilisant la méthode **Maximum Absolu** de la base de données « **salaire** ».

   Nommer le fichier en : **TP1_3_Normalisation_Maximum_Absolu_Salaire.ipynb**
4. **Normaliser les données** en utilisant la méthode **Standardisation** de la base de données « **salaire** ».

   Nommer le fichier en : **TP1_4_Normalisation_Standardisation_Salaire.ipynb**
5. **Normaliser les données** en utilisant la méthode **Robuste** de la base de données « **salaire** ».

   Nommer le fichier en : **TP1_5_Normalisation_Robuste_Salaire.ipynb**
6. **Encoder des variables catégoriques** de la base de données « **achat** ».

   Nommer le fichier en : **TP1_6_Encodage_OneHotEncoder_Achat.ipynb**
7. **Réduction des attributs** de la base de données « **iris** ».

   Nommer le fichier en : **TP1_7_Réduction_PCA_Iris.ipynb**

   Remarque : la base de données « **iris** » doit être importé de *sklearn.datasets*