Atelier 0 : Configuration de l'environnement de travail pour le Machine Learning

Objectif : Installer et configurer un environnement de travail incluant VS Code, Python, Pandas, Scikit-learn, et Matplotlib, afin de préparer un espace prêt pour des projets de Machine Learning.

Étape 1 : Installation de Python

- Accéder au site web : python.org.
- Téléchargez la dernière version stable de Python (assurez-vous que "Add Python to PATH" est coché lors de l'installation).
- Ouvrez une invite de commande ou terminal.
- Testez avec la commande python –version, cela devrait afficher la version installée.

Étape 2 : Installation de VS Code

- Accéder au site web : code.visualstudio.com.
- Téléchargez et installez Visual Studio Code.
- Dans VS Code, installez les extensions suivantes :
  - **Python** : Pour le support de Python.
  - **Jupyter** : Pour exécuter des notebooks directement dans VS Code.
  - **Code Runner** : Pour exécuter rapidement du code Python.

Étape 3 : Configuration d'un environnement virtuel (Optionnel)

Dans une invite de commande, exécutez la commande : python -m venv venv

1. Cela crée un environnement virtuel nommé `venv`.
2. Activer l'environnement avec la commande venv\Scripts\activate
3. L'invite de commande devrait afficher `(venv)` avant le chemin.

Étape 4 : Installation des bibliothèques nécessaires

Avant d'installer les bibliothèques, mettez à jour `pip` : pip install --upgrade pip

Installez les bibliothèques avec une seule commande : pip install pandas scikit-learn matplotlib

Testez l'installation en important les bibliothèques dans un script Python :

```
import pandas as pd
import sklearn
import matplotlib.pyplot as plt
print("Pandas version:", pd.__version__)
print("Scikit-learn version:", sklearn.__version__)
print("Matplotlib fonctionne correctement !")
```

Étape 5 : Configuration dans VS Code

1. Créez un fichier `test_env.py` et ajoutez :

   ```
   import pandas as pd
   import matplotlib.pyplot as plt
   from sklearn.datasets import load_iris
   # Chargement des données
   iris = load_iris()
   df = pd.DataFrame(iris.data, columns=iris.feature_names)
   # Visualisation rapide
   df.plot(kind='scatter', x='sepal length (cm)', y='sepal width (cm)')
   plt.title("Exemple de visualisation avec Matplotlib")
   plt.show()
   ```

2. Lancez le script avec **Run Python File** ou **Code Runner**.

Étape 6 : Test avec un projet simple

Objectif : Vérifier que tout fonctionne correctement en utilisant Pandas, Scikit-learn et Matplotlib en exécutant un simple projet ***Classification avec Iris***. Ajoutez ce code dans `test_env.py` pour tester :

```
from sklearn.datasets import load_iris  # Importer la fonction pour charger le dataset
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
# Chargement du dataset Iris
iris = load_iris()
# Données et cibles
X = iris.data
y = iris.target
# Division en données d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Modèle
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)
# Prédictions
y_pred = model.predict(X_test)
print("Précision du modèle :", accuracy_score(y_test, y_pred))
```

Vous devrez avoir un résultat comme :
“Précision du modèle : 1.0”
