EXPLICATION DU CODE ET DES BIBLIOTHÈQUES

---

1) Pourquoi on utilise des bibliothèques en Data Science

En data science, on utilise des bibliothèques pour :

- lire et manipuler les données,
- faire des calculs,
- visualiser les résultats,
- entraîner et évaluer des modèles.

Chaque bibliothèque a **un rôle précis**. On ne les utilise pas au hasard.

---

2) Explication des bibliothèques utilisées (POURQUOI)

import pandas as pd

Pourquoi ?
pandas sert à manipuler les données sous forme de tableaux (DataFrame), comme un Excel mais en Python.

Utilisation concrète dans ton projet :

- charger les fichiers CSV (read_csv)
- afficher les premières lignes (head)
- sélectionner des colonnes
- créer de nouvelles variables
- nettoyer les données (doublons, valeurs manquantes)

Sans pandas, il est impossible de travailler efficacement sur le dataset.

---

numpy

import numpy as np

Pourquoi ?
numpy est utilisé pour les calculs numériques rapides.

Utilisation concrète :

- calculs mathématiques

- gestion des valeurs numériques
- création de règles (ex : seuils, conditions)

numpy est souvent utilisé en arrière-plan par pandas et les modèles ML.

---

matplotlib

import matplotlib.pyplot as plt

Pourquoi ?
matplotlib permet de créer des graphiques simples.

Utilisation concrète :

- histogrammes (distribution du sommeil, stress, bien-être)
- graphiques en barres
- visualisation des résultats

Les graphiques permettent de **comprendre les données visuellement**, ce qui est essentiel en EDA.

---

seaborn (si présent dans ton notebook)

import seaborn as sns

Pourquoi ?
seaborn est basé sur matplotlib mais permet des graphiques plus lisibles et plus esthétiques.

Utilisation concrète :

- heatmap de corrélation
- boxplots
- comparaison entre groupes

Utilisé pour mieux analyser les relations entre variables.

---

scikit-learn (sklearn)

from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report, confusion_matrix

from sklearn.linear_model import LogisticRegression

Pourquoi ?
scikit-learn est la bibliothèque principale pour le machine learning classique.

Utilisation concrète :

- séparer les données en **train/test**
- entraîner un modèle baseline (régression logistique)
- calculer les métriques (precision, recall, F1)
- afficher la matrice de confusion

Elle est indispensable pour comparer les modèles et mesurer leur performance.

---

3) Explication du notebook Analyse_EDA.ipynb

Objectif de l'EDA

EDA = **Exploratory Data Analysis**
comprendre les données **avant** de faire du machine learning.

On fait l'EDA pour :

- vérifier la qualité des données,
- comprendre les distributions,
- repérer les relations entre variables,
- éviter des erreurs plus tard.

Chargement du dataset

df = pd.read_csv("df_tableau.csv", sep=";")

Pourquoi ?
On charge le dataset pour pouvoir l'analyser.
Le séparateur ; est utilisé car le fichier est au format CSV européen.

Vérification de la structure

df.shape

df.head()

df.info()

Pourquoi ?

- shape : nombre de lignes et colonnes
- head : aperçu des données
- info : types des variables + valeurs manquantes

Permet de comprendre **ce qu'on a vraiment comme données**.

Analyse des valeurs manquantes

df.isna().sum()

Pourquoi ?
Les valeurs manquantes peuvent fausser les résultats du modèle.

On identifie quelles variables posent problème.

Analyse des distributions

plt.hist(df["Sleep_Duration"])

Pourquoi ?
Pour voir :

- si les valeurs sont réalistes,
- s'il y a des valeurs extrêmes,
- la répartition générale.

Cela aide à comprendre le comportement des étudiants.

Corrélations

df.corr()

Pourquoi ?
Pour voir si certaines variables évoluent ensemble (ex : stress ↔ bien-être).

Cela donne des indices utiles pour la modélisation.

---

4) Explication du notebook Exploration_ML.ipynb

Objectif

Tester si on peut **prédire un risque** à partir des variables disponibles.

Séparation X / y

X = df[features]

y = df[target]

Pourquoi ?

- X = variables explicatives (sommeil, stress, etc.)
- y = ce qu'on veut prédire (ex : à risque / pas à risque)

C'est la base de tout modèle ML.

Séparation train / test

X_train, X_test, y_train, y_test = train_test_split(...)

Pourquoi ?

- entraîner le modèle sur une partie des données
- tester sur des données jamais vues

Évite le sur-apprentissage (overfitting).

Modèle baseline : Régression Logistique

model = LogisticRegression()

Pourquoi une baseline ?

- modèle simple
- facile à interpréter
- sert de référence

Si un modèle complexe ne fait pas mieux qu'une baseline, il n'est pas justifié.

Évaluation avec métriques

classification_report(y_test, y_pred)

confusion_matrix(y_test, y_pred)

Pourquoi ces métriques ?

- **Recall** : éviter de rater des étudiants à risque
- **Precision** : éviter trop de fausses alertes
- **F1-score** : compromis
- **Matrice de confusion** : visualisation claire des erreurs

En prévention, le recall est prioritaire.

---

5) Pourquoi tout ce code est nécessaire

Tu peux écrire :

“Le code mis en place permet de vérifier la qualité des données (EDA), de tester la faisabilité du problème via une baseline, puis d’évaluer les performances du modèle à l’aide de métriques adaptées au contexte de prévention. Chaque bibliothèque est utilisée pour un rôle précis : manipulation des données, visualisation, modélisation et évaluation.”