Student dataset

Dataset Description

This dataset, titled "Intro to Data Cleaning, EDA, and Machine Learning," is designed to help learners practice essential data science techniques such as data cleaning, exploratory data analysis (EDA), and machine learning. It contains information on students, including their demographic and academic data, such as age, gender, country of origin, study hours, and scores in Python and Database (DB) courses.
The dataset was initially raw and required significant cleaning to handle inconsistencies, missing values, and outliers, providing an excellent opportunity for hands-on data cleaning and preprocessing.

Data Challenges

- **Inconsistencies**: Mixed formats in `gender`, `country`, and `prevEducation` fields, such as "Male" vs. "M" or "Rsa" vs. "RSA," leading to unreliable analysis.
- **Missing Values**: Incomplete data, particularly in the `Python` and `DB` scores, could bias the results.
- **Outliers**: Extreme or unrealistic values in performance scores can skew predictions or analyses.

Tasks – Data Cleaning, Missing Data, Outliers

Part 1 – Data Cleaning

1. **Check dataset structure**

   - Use `df.shape`, `df.info()`, and `df.head()` to understand the number of rows, columns, and data types.

   - *Question:* Which columns should be categorical and which should be numerical?

2. **Detect inconsistent categories**

   - Run `df['gender'].unique()`, `df['country'].unique()`, and `df['prevEducation'].unique()`.

   - Find issues such as `"Male"` vs `"M"` or `"Barrrchelors"` vs `"Bachelor"`.

   - **Task:** Fix them using `.replace()` or string methods like `.str.upper().strip()`.

3. **Handle duplicates**

   - Check for duplicates with `df.duplicated().sum()`.

   - **Task:** Drop them using `df.drop_duplicates()`.

Part 2 – Missing Data

1. **Identify missing values**

   - Use `df.isnull().sum()` to see which columns have missing data.

   - *Question:* Which columns are most affected by missing values?

2. **Impute missing values**

   - **Option 1 (Numerical):** Fill missing scores in `Python` or `DB` using `mean` or `median`.

   - **Option 2 (Categorical):** Fill missing categories in `country` or `gender` with `mode`.

   - **Task:** Try both methods and compare results.

Part 3 – Outliers

1. **Detect outliers**
    - Use boxplots (`sns.boxplot`) or summary statistics (`df['Python'].describe()`).
    - *Question:* Which values in `studyHOURS`, `Python`, or `DB` look unrealistic?
2. **Handle outliers**
    - **Option 1:** Remove rows where scores are outside a reasonable range (e.g., <0 or >100).
    - **Option 2:** Apply IQR method

Deliverables in GitHub

- A cleaned version of the dataset: `cleaned_students.csv`.
- A short notebook/report explaining:
    - What inconsistencies you found and how you fixed them**. As markdown**
    - How missing values were imputed and why. **As markdown**
    - How outliers were detected and treated. **As markdown**

Data Dictionary

Column Name	Description
`fNAME`	First name of the student.
`lNAME`	Last name of the student.
`Age`	Age of the student in years.
`gender`	Gender of the student (Male/Female).
`country`	The country of origin of the student.
`residence`	The current residence or type of residence (e.g., BI Residence, Private).
`entryEXAM`	The score the student obtained in their entry exam (out of 100).
`prevEducation`	The highest level of education the student had completed (High School, Diploma, Bachelor, Masters, Doctorate).
`studyHOURS`	The number of hours the student spends studying weekly.
`Python`	The score the student achieved in the Python programming course (out of 100).
`DB`	The score the student achieved in the Database (DB) course (out of 100).