
3.a. Comprehensive Student Performance Analysis Using Pandas

Problem Statement

Perform a thorough analysis of student performance across multiple subjects while handling data issues such as missing marks, duplicate entries, and inconsistent formatting. Use various Pandas functionalities to clean, organize, and derive insights from the data.

Objective Scenario

A college stores internal assessment scores of students in an Excel/CSV file for three subjects: **Math**, **Physics**, and **Chemistry**. However, the dataset contains:

- Missing marks due to absent students.
- Duplicate entries for some students.
- Inconsistent string formatting in names.
- No unique ID assigned to students.

The academic head wants to:

1. Clean and prepare the data.
2. Analyze individual and subject-wise performance.
3. Identify high-performing students and subjects with the most variability.

Dataset

The data is stored in a CSV file named `student_scores.csv`, with the following columns:

Example data:

Name,Math,Physics,Chemistry

Alice,78,85,

Bob,82,,88

Charlie,75,79,91

Alice,78,85,

David,,92,87

Eve,85,84,88

Bob,82,,88

Tasks to be Performed

- **Data Loading and Inspection**
  - Load the dataset using Pandas from the CSV file.
  - Display basic information about the dataset using `.info()` and `.describe()`.
  - Count the number of missing entries in each subject column.
  - Detect and display duplicate rows based on student names.
- **Data Cleaning**
  - Remove duplicate entries, keeping only the first occurrence of each student.
  - Replace missing subject scores with the **median** of that subject using `.fillna()`.
  - Assign a new unique identifier column `StudentID` starting from 1.
- **Data Transformation and Normalization**
  - Add a new column `TotalMarks` which sums the marks of all three subjects for each student.
  - Create another column `AverageMarks` as total marks divided by the number of subjects.
  - Apply **min-max normalization** on all three subject scores to scale them between 0 and 1, storing them in new columns:
    - `Math_Norm`
    - `Physics_Norm`
    - `Chemistry_Norm`
- **Statistical and Performance Analysis**
  - Calculate the **mean**, **median**, and **standard deviation** for each subject using normalized scores.
  - Identify the subject with the highest standard deviation (most variable scores).
  - Identify the **top 3 students** based on their average marks.
  - Display subject-wise average performance using a pivot table.

3.b. Movie Ratings Aggregation and Review Analysis

Problem Statement

Analyze movie ratings and review text data to find average ratings, detect duplicates, and extract insights using string operations.

Objective Scenario

A streaming platform wants to analyze user reviews. The dataset contains **User IDs**, **Movie Titles**, **Ratings**, and **Review Comments**. Some reviews are duplicated and some comments contain extra whitespace or punctuation noise.

Dataset

Stored in **`movie_reviews.csv`** with the columns:

Sample data:

UserID,MovieTitle,Rating,ReviewText

U001,Inception,4.5,"Amazing movie!"

U002,Inception,4.0," brilliant "

U001,Inception,4.5,"Amazing movie!"

U003,Interstellar,5.0,"Mind-blowing!"

U004,Interstellar,,"Outstanding visuals."

U005,Inception,4.2,"Good plot but confusing"

Tasks to be Performed

- **Cleaning and Preprocessing**
  - Remove duplicate reviews based on `UserID` and `MovieTitle`.
  - Handle missing ratings by filling them with the **average rating per movie**.
  - Strip and lowercase all review text using vectorized string operations.
- **Aggregation**
  - Group by `MovieTitle` and calculate **mean**, **median**, and **count** of ratings.
  - Identify the **top-rated movie**.
- **Text Analysis**
  - Count how many times the word `"amazing"` appears in all reviews.
  - Filter reviews where the rating is **above 4.0** and the review contains the word `"plot"`.