**Data Fusion and Reshaping for Complex Analytical**

**1. Task Description**

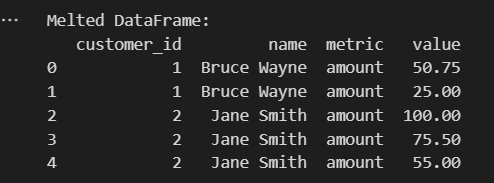
The BrainyBeam internship project requires merging and reshaping data from multiple DataFrames to create complex analytical datasets. The task involves processing three CSV files (data1.csv, data2.csv, data3.csv) containing synthetic data for customer information, transaction records, and engagement metrics. The workflow includes:

* **Loading Data**: Read the CSV files into Pandas DataFrames.
* **Cleaning Data**: Handle duplicates (e.g., duplicate customer\_id in data1.csv), fill missing values (e.g., age and avg\_rating), and convert date columns to datetime format.
* **Merging DataFrames**: Combine the DataFrames on the customer\_id column using a left join to retain all customer records, even those without transactions or engagement data.
* **Reshaping Data**: Transform the merged data into analytical formats using:
  + Pivoting: Create a table of transaction amounts by product category and order date.
  + Aggregation: Compute average visit counts and ratings by region.
  + Melting: Convert data to a long format for metric analysis.
* **Visualization**: Generate a bar plot of total transaction amounts by region to provide insights.
* **Output**: Save the merged and cleaned dataset as analytical\_dataset.csv for further analysis.

The implementation is provided in a Jupyter Notebook (app.ipynb), designed to be adaptable to BrainyBeam’s real data by updating file paths and column names.

**2. Task Output Screenshot**

Below is the screenshot of the output from app.ipynb, showing the first five rows of the melted DataFrame, which demonstrates the reshaping of the merged data into a long format for analytical use.



**3. Widget/Algorithm Used In Task**

Since the project focuses on data processing rather than UI development, no graphical widgets (e.g., EditText, Button) were used. Instead, the following Python/Pandas algorithms and functions were employed in app.ipynb:

* **Pandas** read\_csv: Loaded data1.csv, data2.csv, and data3.csv into DataFrames for analysis.
* **Pandas** drop\_duplicates: Removed duplicate customer\_id entries in data1.csv to ensure data integrity.
* **Pandas** fillna: Filled missing values in age (using the mean) and avg\_rating (using the median) to handle incomplete data.
* **Pandas** to\_datetime: Converted signup\_date, order\_date, and last\_visit\_date to datetime format for consistent time-based analysis.
* **Pandas** merge: Performed left joins on customer\_id to combine DataFrames, handling mismatched keys (e.g., customers without transactions).
* **Pandas** pivot\_table: Created a wide-format table of transaction amount by product\_category and order\_date for analytical insights.
* **Pandas** groupby **and** agg: Aggregated data to compute mean visit\_count and avg\_rating by region.
* **Pandas** melt: Reshaped data into a long format, combining amount, visit\_count, and avg\_rating for flexible analysis.
* **Seaborn** barplot **and Matplotlib**: Generated a bar plot to visualize total transaction amounts by region, enhancing interpretability.
* **Pandas** to\_csv: Saved the final merged and cleaned dataset as analytical\_dataset.csv.

These algorithms enabled efficient data fusion and reshaping, addressing real-world challenges like missing values, duplicates, and mismatched keys.