1. Problems in the Current Dataset

A. Inconsistent User Attributes (The "Identity Crisis")

The primary issue identified in the employee_learning_data.csv file is a severe lack of consistency in user attributes. Upon inspection, we found that individual employees are associated with multiple conflicting values for stable attributes such as role_level, department, and preferred_learning_style.

While it is true that employees can change roles or departments naturally over time, the metadata suggests otherwise in this specific case. An analysis of the module start_date and completion_date columns reveals that these conflicting records were generated within a very short timespan of just a few months. It is highly improbable for an employee to switch departments five times or oscillate between "Manager" and "Associate" roles weekly.

This pattern strongly indicates **data entry issues** or system logging errors rather than genuine organisational changes.

Real-World Context: It is worth noting that this type of "dirty data" is extremely common in real-world scenarios. It often arises from merging data from disparate legacy systems, manual entry errors without validation checks, or logging systems that capture transient states rather than the master record. Cleaning such inconsistencies is a standard and necessary first step before any machine learning pipeline can be built.

Impact on Recommender System Performance: Why is this inconsistency fatal for an AI model? A recommender system relies on finding clear patterns between user attributes and their content preferences. The current volatility breaks this logic in three ways:

1. **Diluted Feature Importance:** Models often learn rules like *"If Department = IT, recommend Python courses."* If the department field randomly flips between 'IT' and 'HR' for the same person, the model sees a weak correlation. It learns to ignore the feature entirely.
2. **Failure of Collaborative Filtering:** Algorithms work by finding "users like you." If your profile (Manager/Visual Learner) keeps changing to (Associate/Auditory Learner), the system cannot consistently identify your peer group.
3. **The "Grey Sheep" Effect:** When a user's data is chaotic, the system fails to personalise for them and defaults to generic "Global Popularity" recommendations.

B. Duplicate Interactions (The "Double-Counting" Problem)

The dataset contains **multiple** where the same employee has interacted with the same training module multiple times.

- **Evidence:** Employee E0001 completed Module 10 in March (Rating: 2) and again in June (Rating: 1).
- **Impact on Recommender Systems:**
    1. **Data Leakage:** In a random Train/Test split, the March entry could go to Training and the June entry to Testing. The model "cheats" by memorizing the user-item pair.
    2. **Conflicting Signals:** Different ratings for the same item confuse the model regarding the user's true preference.
    3. **Bias:** It over-represents active users, optimizing the model for them at the expense of others.

2. Used Solutions

Solution A: Mode Imputation (For Attributes)

To resolve the inconsistent attributes, we implemented the **Mode Imputation Strategy**.

What is the Mode Strategy? The "Mode" in statistics refers to the value that appears most frequently in a data set. In this context, the strategy involves looking at all the records for a single employee, counting how many times each attribute appears, and selecting the winner (the majority vote) as the "true" value. We then overwrite all records for that employee with this single, consistent value.

Example: Imagine Employee E005 has 5 interaction logs with the following preferred_learning_style entries:

1. Visual
2. Visual
3. Auditory (likely an error)
4. Visual
5. Kinesthetic (likely an error)

Using the Mode Strategy, the system identifies **"Visual"** (count = 3) as the mode. It then updates records #3 and #5 so that *all* 5 records for E005 now list "Visual" as the learning style.

Why not use the "Latest Entry"? We could simply use the user attributes from the most recent interaction (based on start_time). However, this approach is typically best when data changes reflect legitimate events, such as a promotion. Since the changes in this dataset appeared random rather than progressive, the "latest" entry is just as likely to be an error as the first entry. The Mode strategy provides a statistically safer consensus for this specific type of data noise.

Solution B: Deduplication (For Interactions)

To resolve the double-counting issue, we identified all duplicate (employee_id, module_id) pairs and kept only the **single best record**. Now there are **2245 rows** in the dataset.

Selection Logic:

1. **Completion Status:** We prioritise records where the status is "Completed" over "Incomplete" or "Dropped."
2. **Recency:** If statuses are equal (e.g., both are Completed), we keep the most recent interaction, assuming the latest attempt reflects the user's current proficiency and sentiment.

Why: This ensures every row in the training data is a unique user-item pair, preventing data leakage and forcing the model to learn from unique preferences.