

# Recommender Systems & Pattern Discovery in E-Commerce

---

## 🔍 Project Title

### SmartCart Recommender System & Pattern Mining for E-Commerce Analytics

---

## 🎯 Objectives

In this group project, your team will simulate an e-commerce platform and implement both **Collaborative Filtering** and **Association Rule Mining** to:

- Recommend products to users based on past interactions and behaviors.
  - Identify frequent purchase patterns.
  - Visualize insights and evaluate recommendation quality.
  - Benchmark your solution on performance and scalability.
- 

## 📦 Provided Dataset: ecommerce\_user\_data.csv

Each row in this CSV file represents a user's rating or interaction with a product:

UserID	ProductID	Rating	ProductCategory	Timestamp
U001	P123	5	Electronics	2024-09-01
U002	P876	3	Home	2024-09-03
...	...	...	...	...

You'll also receive a companion CSV file: **product\_details.csv**:

ProductID	ProductName	Category
P123	Wireless Headphones	Electronics
P876	Blender	Home
...	...	...

---

## Part 1: Data Preprocessing

- Load and clean the data using pandas.
  - Convert ratings data into a **user-item matrix**.
  - Handle missing data by filling unrated items appropriately.
  - Group and aggregate purchase behaviors per user and category.
- 

## Part 2: User-Based Collaborative Filtering (Cosine Similarity)

- Implement cosine similarity to measure user similarity.
  - Recommend products to users based on the preferences of their most similar users.
  - Evaluate the **accuracy** of your recommendations using one or more:
    - Precision / Recall @ K
    - Mean Average Precision (MAP)
    - Coverage and diversity metrics
- 

## Part 3: Association Rule Mining (Apriori)

- Convert user-product purchase data into **transactions**.
  - Implement the Apriori algorithm to find **frequent itemsets**.
  - Generate **association rules** and evaluate them based on:
    - Support
    - Confidence
    - Lift
  - Visualize frequent itemsets using bar charts or network diagrams.
- 

## Part 4: Analysis & Visualization

Create a dashboard or notebook that shows:

- User similarity heatmaps.
  - Top-5 product recommendations for each user group.
- 

## Conceptual Questions (To be Answered in Your Report)

1. How does the sparsity of the data affect your recommender system's performance?
2. What kinds of product bundles were discovered in the association rules?
3. What improvements would you recommend for a real e-commerce system using your approach?

---

## ☑ Deliverables

- notebook\_or\_script.py Or project\_notebook.ipynb with all code.
  - report.pdf (5–8 pages) summarizing your methods, visualizations, findings, and challenges.
  - A brief README.md explaining how to run the code and where to find outputs.
- 



## Evaluation Rubric – SmartCart Project (100 points)

Category	Criteria	Max Points	Notes / Expectations
<b>1. Data Preprocessing</b>	<ul style="list-style-type: none"><li>- Clean and load datasets correctly</li><li>- Create user-item matrix</li><li>- Handle missing values</li><li>- Aggregate user behavior by category</li></ul>	<b>15</b>	Code should show correct use of pandas and good data organization.
<b>2. Collaborative Filtering</b>	<ul style="list-style-type: none"><li>- Cosine similarity implementation</li><li>- Top-N recommendations per user</li><li>- Evaluation using Precision@K / Recall@K / MAP</li><li>- Coverage or diversity included</li></ul>	<b>15</b>	Recommendations must be based on most similar users and evaluated properly.
<b>3. Association Rule Mining</b>	<ul style="list-style-type: none"><li>- Convert data into transaction format</li><li>- Use Apriori correctly</li><li>- Generate and evaluate rules with support, confidence, lift</li><li>- Visualize results</li></ul>	<b>15</b>	Frequent itemsets and rules must be interpretable and meet thresholds.
<b>4. Analysis &amp; Visualization</b>	<ul style="list-style-type: none"><li>- User similarity heatmap</li><li>- Frequent itemsets chart</li><li>- Recommendation outputs (e.g., table or CSV)</li><li>- Insightful visual storytelling</li></ul>	<b>15</b>	Visuals should help explain the findings and add analytical value.
<b>5. Report (report.pdf)</b>	<ul style="list-style-type: none"><li>- Structure: intro, methods, results, insights, challenges</li><li>- Addresses conceptual questions thoughtfully</li><li>- Includes screenshots of visuals, if applicable</li><li>- Shows understanding of tradeoffs and improvements</li></ul>	<b>30</b>	Should reflect critical thinking, not just descriptive summary.
<b>6. Code Quality &amp; Execution</b>	<ul style="list-style-type: none"><li>- Code runs without errors</li><li>- Modular, reusable functions</li><li>- Commented and readable</li><li>- Organized (e.g., consistent filenames, clear entry point)</li></ul>	<b>5</b>	Notebook or .py file should be easy to follow and maintain.
<b>7. README File</b>	<ul style="list-style-type: none"><li>- Clear instructions to run the project</li><li>- Explains dependencies, file structure, and outputs</li><li>- Concise and useful</li></ul>	<b>5</b>	Must help reviewers/instructors reproduce the results easily.

---

## **Provided Dataset:**

You are provided with two offline datasets:

-  `ecommerce_user_data.csv` – contains user-product interactions and ratings
  -  `product_details.csv` – contains product metadata including category
- 

## **Starter Notebook (Optional)**

To help you get started, we've provided a **starter Jupyter notebook**.

◊ **Using this notebook is completely optional.** You are encouraged to use your own implementation and structure if preferred.