Assignment 3: Association Rule Mining for Market Basket Analysis

Data Analytics 1

Release: October 3, 2025
Deadline: October 12, 2025 (11:55 pm)

Marking Distribution (Total 100 Marks)

- **Part A:** Data Preprocessing **15 Marks**
- **Part B:** Association Rule Mining **35 Marks**
- **Part C:** Recommendation and Evaluation **35 Marks**
- **Report and Code Quality** **15 Marks**

---

1 Objective

The objective of this assignment is to perform market basket analysis to discover associations between products purchased together. You will build a system that can suggest products to a user based on the items already in their shopping cart using association rules.

2 Dataset Description

You will use the "Instacart Market Basket Analysis" dataset from Kaggle. This dataset contains anonymized data on 3 million grocery orders. We will primarily use the following files:

- `orders.csv`: Contains order information for each user.
- `order_products__train.csv`: Contains the products for all orders in the "train" evaluation set. This will be your primary data source.
- `products.csv`: Contains product names.

3 Assignment Tasks

Part A: Data Preprocessing [15 Marks]

1. **Form the Transactional Dataset (10 marks):** From the dataset, identify all orders where `eval_set` is 'train'. Create a list of transactions, where each transaction is an `order_id` associated with the set of `product_ids` in that order.

2. **Split the Data (5 marks):** Randomly divide your transactional dataset into a training set (80% of transactions) and a test set (20% of transactions).

Part B: Association Rule Mining [35 Marks]

1. **Mine for Rules (25 marks):** From your **80% training set**, extract all association rules of the form $\{A\} \rightarrow Y$, where A contains a **single product** and Y can contain **one or more products**. You may use either the **Apriori** or **FP-growth** algorithm.
2. **Experiment with Thresholds (10 marks):** You must experiment with different values for minimum support (*minsup*) and minimum confidence (*minconf*) to find a combination that produces a meaningful number of rules. For example, you could try values for *minsup* in the range [0.001, 0.01] and for *minconf* in the range [0.01, 0.1]. Plot suitable graphs to arrive on an optimum value and justify your final choice in the report.

Part C: Recommendation and Evaluation [35 Marks]

1. **Generate Rule Lists (5 marks):** Create two lists of the top 100 association rules found: one sorted by **support** and the other by **confidence**. Identify and present the rules that appear in **both** lists.
2. **Implement Evaluation Logic (15 marks):** For each transaction in your **20% test set**, create an "input" and a "ground truth". For example, for a basket with more than one item, use the first half of the items as input and the second half as the ground truth to be predicted.
3. **Calculate and Plot Metrics (10 marks):** Compute the **average precision** and **average recall**. Generate plots showing how these metrics change as you vary the number of recommendations per user (from 1 to 10).
4. **Sample Analysis (5 marks):** Show the precision and recall graphs for a few sample users from your test set and briefly discuss their results.

4 Report and Code Quality [15 Marks]

Your submission will also be graded on the quality of your report and code.

- **Report (10 marks):** The report must be clear, well-structured, and provide justifications for your choices (e.g., algorithm, final *minsup*/*minconf*). It should include all plots and a thorough interpretation of your findings and results.
- **Code (5 marks):** The code should be well-commented, readable, and efficiently implemented.

5 Submission Format

Please submit a single zip folder named `<assignment3_teamId>.zip` containing:

- `<TeamId>_report.(pdf)`: Your report should justify your choice of algorithm, explain your data processing steps, and interpret the results and plots.
- `<TeamId>_recommender.(ipynb)`: Your fully commented code.
- `<TeamId>_top100RulesByConf.txt`: Your list of top 100 rules by confidence.
- `<TeamId>_top100RulesBySup.txt`: Your list of top 100 rules by support.