Report – Association Rule Mining Supermarket System

Course: CAI 4002 – Artificial Intelligence

Semester: Fall 2025

Authors: Eduardo Goncalvez, Alex Waisman, Iván Salazar

1. Introduction

This report presents an interactive supermarket simulation system combined with data preprocessing and association rule mining.

The objective is to generate meaningful product associations using Apriori, Eclat, and an Association Rule Generator.

The system allows users to manually create shopping transactions, import datasets, clean the data, and analyze patterns through a modern interface built with Streamlit.

2. System Overview

The system simulates a supermarket where a user can:

- 1. Create transactions by selecting products.
- 2. Import CSV files containing transaction data.
- 3. Preprocess inconsistent or messy datasets.
- 4. Run Apriori and Eclat to extract frequent itemsets.
- 5. Generate association rules (support, confidence, lift).
- 6. Explore product-to-product recommendations interactively.

The platform integrates data cleaning, mining, and visualization in a single workflow.

3. Technical Stack

- **Programming Language:** Python 3.10+
- **Framework:** Streamlit
- **Libraries:**
  - pandas
  - streamlit
- **Algorithms implemented manually:**
  - Apriori
  - Eclat
  - Association Rule Generator
- **Dataset files:** products.csv, sample_transactions.csv, cleaned_transactions.csv

4. Data Preprocessing

Before mining, the system performs a multi-step cleaning process:

1. **Standardization**
   - Convert item names to lowercase
   - Trim whitespace
2. **Duplicate Removal**
   - Remove repeated items inside the same transaction
3. **Invalid Product Filtering**
   - Remove items not found in products.csv
4. **Transaction Filtering**
   - Remove empty baskets
   - Remove single-item transactions
5. **Statistics computed**
   - Number of valid transactions
   - Total items
   - Number of unique items
   - Duplicate count
   - Invalid product count

A sample output is saved into cleaned_transactions.csv.

5. Algorithm Descriptions

5.1 Apriori

- Uses breadth-first exploration (level-wise).
- Generates candidate itemsets from previous frequent itemsets.
- Prunes combinations with insufficient support (anti-monotonic property).
- Produces itemsets grouped by size (L1, L2, L3, …).

5.2 Eclat

- Uses vertical data representation (TID-sets).
- Intersects transaction ID sets to compute support.
- Depth-first exploration.
- More efficient than Apriori on sparse datasets.

5.3 Association Rule Generator

For each frequent itemset with size ≥ 2:

- Generates all non-empty subsets
- Computes:
  - **Support**
  - **Confidence**
  - **Lift**
- Keeps rules with confidence ≥ minimum confidence threshold

7. Results and Performance

Dataset used: Cleaned dataset (80–100 transactions)

Parameters:

- min_support = 0.2
- min_confidence = 0.5

Performance Summary (Text Version):

- Apriori
  - Runtime: ~1.55ms
  - Memory Usage: ~0.008MB
  - Rules Generated: 11
- Eclat
  - Runtime: ~0.55ms
  - Memory Usage: ~0.011MB
  - Rules Generated: 11

Eclat showed faster execution due to vertical TID-set intersections.

8. Project Structure (Text Version)

```
project-root/
├── src/
│   ├── algorithms/
│   │   ├── apriori.py
│   │   ├── eclat.py
│   │   ├── performance_comparison.py
│   │   └── association_rules.py
│   ├── preprocessing/
│   │   └── preprocessing_utils.py
│   └── frontend/
│       ├── components/
│       │   ├── data_import.py
│       │   ├── home.py
│       │   ├── mining.py
│       │   ├── preprocessing.py
│       │   ├── shopping.py
│       │   └── transactions.py
│       └── app.py
├── data/
│   ├── sample_transactions.csv
│   ├── products.csv
│   └── cleaned_transactions.csv
├── README.md
├── .gitignore
├── requirements.txt
└── REPORT.pdf
```

9. Testing

The following functionality was verified:

- CSV import and parsing
- Duplicate and invalid item detection
- Removal of empty and single-item transactions
- Apriori frequent itemset generation
- Eclat vertical TID-set mining
- Confidence and lift calculations
- Recommendation accuracy in the UI
- Support/confidence threshold changes

10. Limitations

- No CLOSET or FP-Growth implementation.
- Streamlit UI performance depends on session state.

11. Conclusion

This project successfully combines transaction creation, data preprocessing, frequent itemset mining, and association rule generation into a unified interactive Streamlit application.

Both Apriori and Eclat were implemented from scratch and produced consistent, meaningful associations, with Eclat demonstrating faster performance due to its vertical TID-set structure and depth-first exploration.

The interface allows users to explore recommendations in real time, adjust support and confidence thresholds, and review both technical and non-technical outputs.

Overall, the system demonstrates a solid practical understanding of data cleaning, algorithm design, and applied association rule mining.