Market Basket Analysis using Apriori Algorithm (Online Retail Dataset)

Project Abstract · AI & Machine Learning · Built-to-order project

1. Introduction / Background

Retailers have long asked a deceptively simple question: which products do customers buy together? The answer drives shelf placement, cross-sell bundles, and the "frequently bought together" panels of every e-commerce site. Market basket analysis formalises the question as *association rule mining* over transaction data, and the Apriori algorithm — introduced by Agrawal and Srikant in 1994 — remains the canonical method taught for it. Its central insight, the Apriori property (every subset of a frequent itemset is itself frequent), lets the search prune vast numbers of candidate itemsets without counting them. This project applies genuine Apriori mining to the UCI Online Retail data set, real transactions from a UK-based online gift retailer recorded between December 2009 and December 2011, and delivers the results through a live, single-file web demonstration that mines the data in the visitor's own browser.

2. Problem Statement

Students typically meet Apriori through ten-row toy examples, which hide every difficulty that makes the algorithm interesting: real transaction data is noisy (cancelled orders, returns, missing descriptions), product catalogues run to thousands of items, and the choice of minimum-support and minimum-confidence thresholds completely reshapes the rule set. Without running the algorithm on real data, concepts like support, confidence and lift remain definitions to memorise rather than measures to reason with. The problem addressed here is to build an end-to-end, honest market-basket analysis pipeline — data cleaning, sampling, a faithful hand-implemented Apriori engine, threshold exploration, and a recommender-style application — on a recognised public retail data set.

3. Objectives

- Clean and prepare the UCI Online Retail data set into transaction baskets with a reproducible sampling procedure.
- Implement the Apriori algorithm from first principles (candidate generation, subset pruning, support counting) without relying on a data-mining library.

- Discover association rules with support, confidence and lift at documented thresholds, reporting only values actually computed by the run.
- Build an interactive web demo with three views: a rule explorer with tunable thresholds, a basket builder that recommends co-purchased products, and a data-insights dashboard.
- Document the methodology, the interestingness measures with worked examples, and the effect of threshold choices, at a standard suitable for a final-year project report and viva.

4. Proposed Methodology

The pipeline has four stages. **(i) Data preparation (Python):** cancelled invoices, non-positive quantities and blank descriptions are removed; invoice numbers are grouped into baskets; baskets with fewer than two items are dropped, leaving 18,338 clean transactions; a reproducible 4,000-transaction sample (seed 42) is drawn. **(ii) Encoding:** products are mapped to integer IDs and embedded in the demo as compact arrays. **(iii) Mining (JavaScript, in-browser):** tid-lists (sorted transaction-ID lists) are built per product; frequent 1-itemsets seed level-wise candidate generation — candidates are formed by joining frequent (k−1)-itemsets that share a prefix, and any candidate possessing an infrequent (k−1)-subset is pruned before counting; support is counted by tid-list intersection with early exit. **(iv) Rule generation:** every frequent itemset of size ≥ 2 is split into all antecedent/consequent partitions; partitions meeting the confidence threshold become rules, each annotated with support, confidence and lift.

5. System Architecture / Working

The system is a single self-contained HTML file. On load it parses the embedded transaction sample, builds product frequencies and tid-lists, and renders the data-insights dashboard (top-10 products, basket-size distribution, sample statistics). The rule explorer runs the Apriori engine on demand with user-set thresholds and tabulates the top 100 rules sortable by lift, confidence or support. The basket builder lets the user search the full 3,637-product catalogue, assemble a basket, and receive ranked "frequently bought together" recommendations computed live from pairwise support, confidence and lift against the basket contents. The offline Python script performs cleaning, sampling and a reference mining run used to validate the browser engine.

6. Hardware / Software Requirements

Component	Requirement

Processor / RAM	Any dual-core machine, 4 GB RAM (demo mines 4,000 transactions in ~0.4 s)
Browser	Any modern Chromium / Firefox / Safari release (demo is plain HTML + JS, no build step)
Data preparation	Python 3 with pandas and openpyxl
Data set	UCI Online Retail (public, ~45 MB xlsx)
Optional	Git for version control

7. Expected Outcomes

The reference run — minimum support 1%, minimum confidence 50% — discovers **2,972 frequent itemsets** and **1,940 association rules** from the 4,000-transaction sample in approximately 0.4 seconds in a desktop browser. The strongest rule found is {HERB MARKER BASIL + HERB MARKER ROSEMARY} → {HERB MARKER THYME}, with support 1.10%, confidence 97.8% and lift 78.2×. The high-lift rules consistently describe product families bought as sets (herb-marker sets, cake-decorating bundles, matching bag ranges), which is precisely the cross-sell signal retailers act on. All figures above are values computed by the actual reference run, and the browser demo reproduces them exactly. As an unsupervised method, the project reports interestingness measures rather than accuracy, which does not apply to association mining.

8. Applications

- Cross-sell and "frequently bought together" recommendation panels for e-commerce.
- Bundle design and promotional offer construction from high-confidence rules.
- Shelf and catalogue adjacency planning for brick-and-mortar retail.
- Teaching aid for data-mining courses: thresholds, pruning and interestingness measures become explorable instead of abstract.
- Template pipeline reusable on any invoice/item data set (the cleaning script accepts the same two-column shape).

9. Future Scope

- Constraint-based mining: restrict consequents to high-margin categories to surface profit-relevant rules.
- Sequential pattern mining (e.g., GSP) to capture *order* of purchase across repeat invoices.
- Scaling the miner with vertical bitmap representations or FP-Growth for million-transaction catalogues.
- Online learning: incrementally updating tid-lists as new transactions stream in.
- A/B evaluation of the basket recommender against popularity baselines on held-out transactions.

Reference run (computed, not claimed): 4,000 transactions · 3,637 products · min-support 1% · min-confidence 50% → 2,972 frequent itemsets · 1,940 rules · ~0.4 s in-browser. Top rule lift 78.2×.

Keywords: market basket analysis, Apriori algorithm, association rule mining, support, confidence, lift, UCI Online Retail, recommender systems, unsupervised learning.