Graph-Based E-Commerce Recommender System: Neo4j Implementation with Jaccard Similarity

Intelligent Product Recommendations Through Network Analysis

Problem Statement

E-commerce platforms generate massive transaction data, yet traditional recommender systems using matrix-based collaborative filtering struggle with sparse datasets (98%+ empty cells), lack transparency in recommendation logic, and require complete model retraining for updates. The challenge is to build a recommendation engine that efficiently handles sparse data, provides explainable suggestions, enables real-time updates, and maintains accuracy comparable to established methods while supporting complex relationship queries beyond simple similarity matching.

Business Value

Effective product recommendations drive 15-30% increases in average order value, reduce customer search time by 40%, improve conversion rates by 10-20%, and enhance user satisfaction through personalized experiences. For online retailers processing thousands of transactions daily, intelligent recommendations directly translate to millions in additional revenue while reducing cart abandonment. The explainability of graph-based recommendations also builds customer trust and enables data-driven inventory management decisions.

Methodology

Dataset: UCI Machine Learning Repository's Online Retail Dataset from UK-based gift wholesaler. Original: 541,909 transactions, cleaned to 346,087 valid transactions representing 3,776 unique customers purchasing 3,461 unique products over 12 months (Dec 2010 - Dec 2011). Dataset includes invoice details, stock codes, product descriptions, quantities, prices, timestamps, and customer IDs.

Data Cleaning Pipeline (10-Step Process):

- Removed 135,080 transactions with missing customer IDs (can't recommend to unknown users)
- Eliminated cancelled orders (invoices starting with 'C')
- Filtered negative/zero quantities and prices
- Removed extreme outliers (unrealistic bulk quantities)
- Deduplicated transactions
- Focused on UK market only (346,453 transactions)
- Retained customers with 3+ purchases (minimum data for patterns)
- Eliminated products purchased by only 1 customer (not generalizable)
- Result: 36% reduction yielding high-quality, analysis-ready dataset

Graph Database Architecture (Neo4j):

- Node types: Customer nodes (3,776), Product nodes (3,461)
- Relationship type: PURCHASED edges with properties (quantity, price, date)
- Total relationships: 346,087 customer-product connections

- Optimized for sparse data: only stores actual connections (not empty cells)
- Cypher query language for natural graph traversal
- Bolt protocol for Python integration via neo4j driver

Similarity Computation: Jaccard Index

- Formula: $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$ where A and B are product sets
- Measures overlap between two customers' purchase histories
- Score range: 0.0 (no common products) to 1.0 (identical purchases)
- Threshold filtering: similarity ¿ 0.1 for relevant recommendations
- Computationally efficient on graph structure
- Naturally handles varying purchase frequencies

Recommendation Generation Algorithm:

- Step 1: Identify target customer's purchased products
- Step 2: Find similar customers using Jaccard similarity calculation
- Step 3: Extract products purchased by similar customers
- Step 4: Filter out products target customer already owns
- Step 5: Rank by frequency among similar customers and average similarity
- Step 6: Return top-N recommendations with scores and explanations

Python Implementation Stack:

- pandas: Data manipulation, cleaning, transformation
- neo4j driver: Database connectivity, transaction management
- NetworkX: Graph visualization and analysis
- Matplotlib/Seaborn: Statistical visualizations, distribution analysis
- Surprise library: Validation benchmarking (SVD algorithm)
- Batch processing: UNWIND clause for efficient bulk operations (1,000 records/batch)

Validation Methodology:

- Comparative analysis with Surprise library's SVD collaborative filtering
- Jaccard similarity scores correlated with SVD predicted ratings
- Sample validation: Customer 17850 and 5 similar customers
- Pearson correlation coefficient calculation
- Scatter plot visualization of method agreement
- Academic validation following Oprea & Bâra (2025) methodology

Results

Dataset Characteristics:

- Average purchases per customer: 91.7 transactions
- Average unique products per customer: 62.7 items
- Most active customer: 7,667 purchases (wholesaler/retailer)
- Most popular product: White Hanging Heart T-Light Holder (1,986 purchases)
- Matrix sparsity: 98.19% (only 1.81% of possible customer-product pairs exist)
- Customer engagement: 42% low activity (11-50 purchases), 2% very high (500+)

Recommendation System Performance:

- Validation correlation: r = 0.892 (very strong agreement with traditional CF)
- Perfect match detection: Jaccard = 1.0 correctly identifies identical purchase patterns
- Ranking consistency: Both graph-based and matrix-based methods identify same top similar customers
- Query performance: ¡ 2 seconds for similarity calculation and top-10 recommendations
- Coverage: 90%+ of customers receive recommendations (high system utility)

Example Recommendation Case (Customer 17850):

- Customer's purchases: 175 unique products (home decor, kitchenware)
- Top similar customer: 100% Jaccard similarity (identical purchase subset)
- Second similar: 14.4% Jaccard similarity (89 common products)
- Top recommendations: Party bunting, lunch bags, bird ornaments
- Recommendation explanation: "Purchased by 8 similar customers with avg 0.12 similarity"

Graph Database Advantages Demonstrated:

- Sparse data efficiency: 98.19% sparsity handled without memory waste
- Explainability: Visual graph shows why products are recommended
- Real-time updates: Add new purchases instantly without retraining
- Complex queries: "Find similar UK customers who bought red products in December"
- Scalability: Batch processing handles 346K relationships efficiently
- Transparency: Cypher queries readable by non-technical stakeholders

Key Statistical Insights:

- Product popularity follows power law: few super-popular, many niche items
- Customer purchase frequency: median 42, mean 91.7 (right-skewed distribution)
- Seasonal patterns: November-December peak (holiday shopping surge)
- Price distribution: median GBP 1.95, mean GBP 3.12 (affordable gift items)
- Typical order: 1-10 items, GBP 1-5 per item

Validation Against Industry Standard:

- Surprise library SVD predictions: 1.39-5.0 score range
- Graph-based Jaccard scores: 0.097-1.0 range
- Strong linear correlation: validates graph approach accuracy
- Academic citation: Oprea & Bâra (2025) IEEE Access publication
- Comparable accuracy with added interpretability benefits

Production Deployment Considerations:

- Infrastructure: Neo4j Community Edition (free) or Enterprise (clustering)
- Caching strategy: Pre-compute top-N similarities for frequent queries
- Batch updates: Nightly refresh for new transactions
- Monitoring: Track recommendation CTR, conversion rate, diversity
- Scalability: Tested on 346K edges, scales to millions with indexing
- API integration: RESTful endpoint for real-time recommendations

Business Impact Projections:

- Expected sales increase: 15-30% from personalized recommendations
- Conversion rate improvement: 10-20% on recommended products
- Customer satisfaction: Reduced search time, relevant suggestions
- Inventory optimization: Data-driven stocking decisions
- Marketing efficiency: Targeted campaigns based on purchase patterns

---

This project demonstrates production-ready graph-based recommendation technology, validated against industry standards and optimized for real-world e-commerce deployment with explainable AI principles.