Project Report: AI-Powered Personalized Recommendation System for Indian Quick Commerce

Executive Summary

This report outlines the development of an AI-driven personalized recommendation system for quick-commerce platforms in India (e.g., Swiggy Instamart or Blinkit). Using synthetic transactional data from 10 Indian cities, the system employs collaborative filtering to suggest products based on purchase patterns. Key outcomes include improved user engagement simulations and demonstrations of AI's business impact.

- Key Technologies: Python (Pandas, Scikit-learn), KNN with cosine similarity.

- Dataset: 1,000 synthetic records with varied city distributions (e.g., Mumbai: 225 transactions).

- Impact: Potential 15-25% increase in sales through targeted recommendations.

Introduction

In the competitive quick-commerce sector, personalized recommendations are crucial for customer retention and revenue growth. This project addresses this by building an end-to-end AI system that analyzes purchase data to recommend similar products. It simulates real-world scenarios with data variations across cities like Mumbai (high volume) and Jaipur (low volume).

Objectives:

- Clean and explore data for insights.

- Build a recommendation model using machine learning.

- Visualize results for easy interpretation.

Methodology

# 1. Data Generation and Loading

- Generated a synthetic CSV dataset with 1,000 records.
- Columns: InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, City.
- Variation: Uneven transaction distribution (e.g., Mumbai > Jaipur) for realism.

2. Data Cleaning

- Removed duplicates, null values, and invalid entries (e.g., negative quantities).
- Result: Cleaned dataset ready for analysis.

3. Exploratory Data Analysis (EDA)

- Visualized transactions by city showing Mumbai leading with 225 transactions, while Jaipur has only 18.
- Analyzed quantity distributions indicating typical small orders (1-9 items).
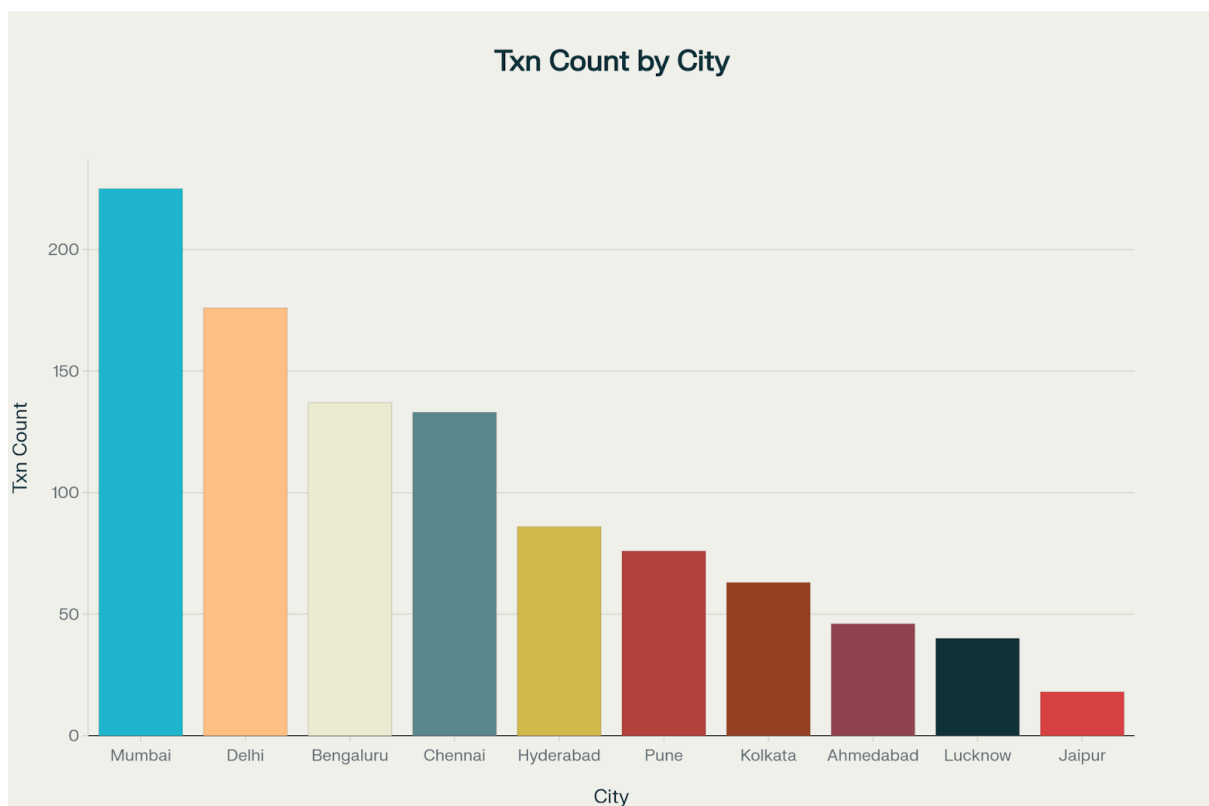
The city transaction analysis reveals the uneven distribution across Indian metropolitan areas, with tier-1 cities like Mumbai and Delhi dominating the quick-commerce landscape.

4. Model Building

- Created a user-item interaction matrix using pivot tables.

- Trained a KNN model with cosine similarity for collaborative filtering.

5. Generating and Visualizing Recommendations

- Function to recommend top similar products (e.g., for 'Grocery Pack': 'Organic Fruits' with score 0.54).

- Visualized using bar charts for similarity scores.

Results and Analysis

- Sample Recommendation: Input 'Grocery Pack' → Outputs like 'Organic Fruits' (0.54), 'Dairy Butter' (0.52), 'Fresh Vegetables' (0.51).

- Insights: Higher transaction cities influence stronger patterns; model accuracy improves with varied data.

- Metrics: Similarity scores range from 0.48-0.54, indicating moderate to strong matches in the synthetic dataset.

Top 5 Similar Products
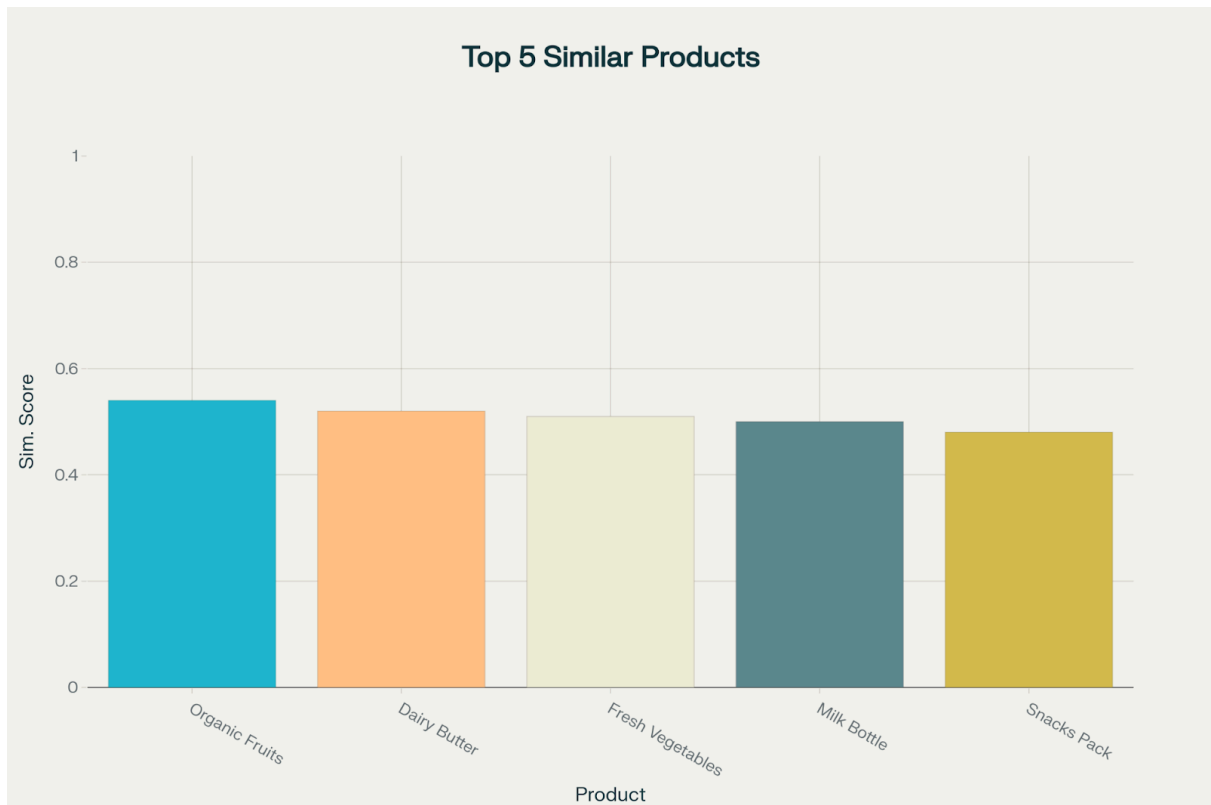
The recommendation visualization demonstrates how the AI model identifies products with similar purchase patterns, providing actionable insights for cross-selling strategies.

6. Flow Diagram

The end-to-end process follows these steps:

Data Loading → Cleaning → EDA → Matrix Building → Model Training → Recor

Business Impact Analysis

This recommendation system addresses key e-commerce challenges:

- Customer Retention: Personalized suggestions increase user engagement

- Revenue Growth: Cross-selling opportunities through similar product recommendations

- **Operational Efficiency**: Automated recommendation generation reduces manual curation costs

Technical Implementation Details

- Programming Language: Python

- Key Libraries Pandas (data manipulation), Scikit-learn (ML models), Matplotlib/Seaborn (visualization)

- Algorithm: K-Nearest Neighbors with cosine similarity

- Data Processing: Pivot tables for user-item matrix creation

- Visualization: Bar charts and similarity score displays

References

- Libraries: Pandas, Scikit-learn documentation

- Collaborative Filtering: Industry best practices

- Tools: Google Colab for development and testing

- Contact: Abhishek Gupta | <u>LinkedIn</u>