E-commerce Recommendation System: A Statistical Analysis and Implementation

Nandan Keshav Hegde[1], Nirgude Siddhish[1], and Sivagugan Jayachandran[1]

[1]Michigan State University, Department of Statistics and Probability

October 23, 2024

Abstract

This project proposes the development of an advanced E-commerce recommendation system using R and python frameworks. By leveraging statistical methods including Bayesian analysis, conditional probability, and multivariate statistics, we aim to create a dynamic recommendation engine that provides personalized product suggestions based on user behavior patterns. The system will analyze data [5-6 months] of user interaction data to generate insights and recommendations while demonstrating practical applications of advanced statistical concepts.

Contents

1 Introduction

In the rapidly evolving E-commerce landscape, personalized recommendation systems have become crucial for enhancing user experience and driving business growth. This project focuses on developing a recommendation system to know the purchase patterns of customers visiting the website.

1.1 Project Objectives

- Develop product recommendation system
- Implement advanced statistical methods from course curriculum
- Create an interactive dashboard
- Analyze and validate recommendation effectiveness
- Demonstrate practical applications of theoretical concepts

2 Dataset Specifications

2.1 Data Structure

Our analysis utilizes a comprehensive dataset containing the following attributes:

Attribute	Type	Description
event_time	Timestamp	Interaction timestamp
event_type	Categorical	View/cart/purchase
product_id	Numeric	Unique product identifier
category_id	Numeric	Product category ID
category_code	String	Category description
brand	String	Product manufacturer
price	Numeric	Product price
user_id	Numeric	Unique user identifier
user_session	String	Session identifier

Table 1: Dataset Structure

3 Technical Framework

3.1 Mathematical Foundation

3.1.1 Probability Models

For a user u and product p, our recommendation score $R(u, p)$ is computed as:

$$R(u, p) = \alpha P(p|H_u) + \beta P(p|S_u) + \gamma P(p|C_u) \tag{1}$$

Where:

- H_u is the user's purchase history
- S_u is the current session behavior
- C_u is the category preference
- α, β, γ are weights determined through MLE

3.1.2 Time Series Component

We model temporal patterns using an exponential decay function:

$$w(t) = e^{-\lambda(t_{now} - t_{interaction})} \tag{2}$$

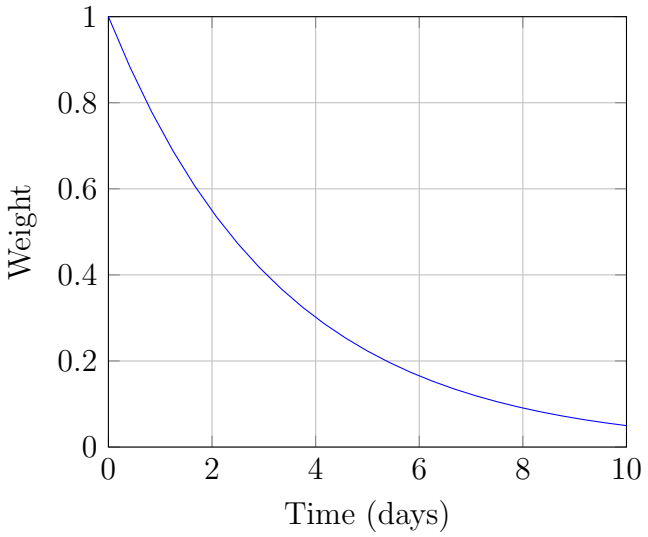

Figure 1: Temporal Decay of Interaction Weights

3.2 Statistical Methods

3.2.1 Collaborative Filtering Model

The user-item similarity matrix S is computed as:

$$S_{ij} = \frac{\sum_{k=1}^{n}(r_{ik} - \bar{r_i})(r_{jk} - \bar{r_j})}{\sqrt{\sum_{k=1}^{n}(r_{ik} - \bar{r_i})^2}\sqrt{\sum_{k=1}^{n}(r_{jk} - \bar{r_j})^2}} \tag{3}$$

3.2.2 Price Sensitivity Analysis

We model price sensitivity using a log-normal distribution:

$$f(x|\mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}} \tag{4}$$

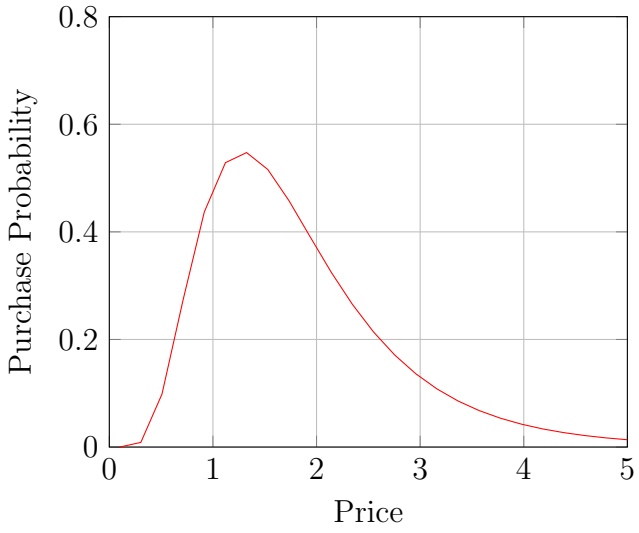

Figure 2: Price Sensitivity Distribution

4 Implementation Methodology

4.1 Phase 1: Data Preprocessing

- **Initial Data Cleaning**
  - Handle missing values in brand and category fields
  - Standardize timestamp formats
  - Remove duplicate entries
- **Feature Engineering**
  - Create time-based features
  - Calculate inter-event times
  - Develop user behavior metrics

4.2 Phase 2: Statistical Analysis

Algorithm 1 Product Recommendation Algorithm

```
procedure GenerateRecommendations(user, n)
    history ← GetUserHistory(user)
    preferences ← ComputePreferences(history)
    candidates ← GetCandidateProducts()
    scores ← ∅
    for each product in candidates do
        p_history ← HistoryProbability(product, history)
        p_session ← SessionProbability(product)
        p_category ← CategoryProbability(product, preferences)
        score ← αp_history + βp_session + γp_category
        scores.append((product, score))
    end for
    return TopN(scores, n)
end procedure
```

5 R Shiny or Python framework Implementation

5.1 Dashboard Components

- User Interface
  - Product selection panel
  - Recommendation display
  - Statistical metrics visualization
- Server Logic
  - recommendation updates
  - Statistical model execution
  - Data processing pipelines

6 Evaluation Framework

6.1 Statistical Metrics

$$\text{Precision@K} = \frac{|\text{relevant items} \cap \text{recommended items}|}{K} \tag{5}$$

$$\text{MAP} = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{1}{n_u} \sum_{k=1}^{n_u} P(k) \times rel(k) \tag{6}$$

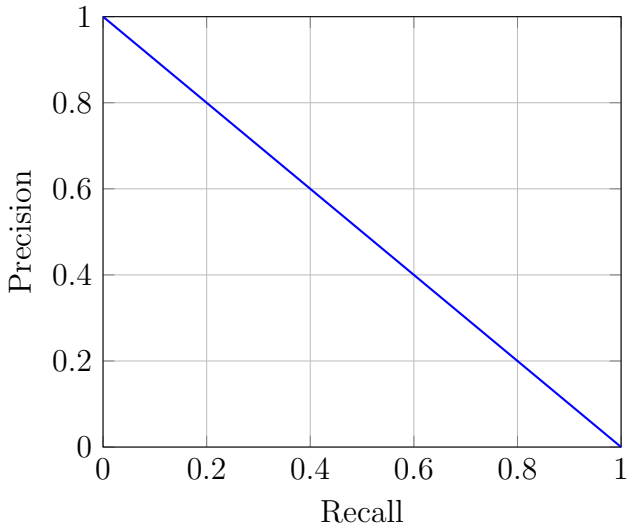

Figure 3: Precision-Recall Curve

7 Timeline and Deliverables

Weeks	Phase	Deliverables
0-1	Data Preprocessing	Cleaned dataset
1-2	Statistical Analysis	Probability models
2-3	Model Development	Initial algorithms
3-4	Implementation	dashboard

Table 2: Project Timeline

8 Challenges

Developing an effective E-commerce recommendation system presents several challenges:

- **Data Sparsity**: Many users interact with only a small portion of available products, leading to sparse interaction matrices, which can limit the effectiveness of collaborative filtering models.
- **Scalability**: As the number of users and products increases, computational efficiency becomes a concern. Efficient algorithms and parallel processing may be required to handle large-scale data in real-time.
- **Cold Start Problem**: Recommending products for new users or products (with no historical data) is a well-known challenge, requiring hybrid models that can leverage both content-based and collaborative filtering techniques.

- **Dynamic Preferences**: User preferences can change over time, necessitating time-sensitive models that can quickly adapt to new patterns of behavior.
- **Bias and Fairness**: Ensuring that the recommendation system remains unbiased and fair across different demographic groups and user segments is crucial, especially in commercial applications where biased recommendations can impact customer trust and sales.

9 Conclusion

This project serves as a comprehensive integration of theoretical knowledge acquired from our Statistics 810 coursework with practical, hands-on implementation in R Shiny, a popular web application framework for R or other frameworks in Python like dash. The primary objective is to bridge the gap between statistical theory and its real-world applications, providing an opportunity to reinforce and deepen our understanding of key concepts learned throughout the course.

Ultimately, the project serves as a capstone for the Statistics 810 class, providing a holistic learning experience that combines the rigor of statistical theory with the practicality of modern data analysis tools. By actively engaging with both the theory and its application, we aim to maximize our understanding and mastery of statistical methodologies in a hands-on manner, preparing us for future real-world challenges in the field of data science and statistics.