DA626 Project Report

Report for Implementations in Final Project

Team Members: DEVIKA SINGH, VARSHA, RINSHI KUMARI
Roll Numbers: 210101036, 210101109, 210108040

0.1 Application Analysis, Framework Adaptability, and Dataset Overview

0.1.1 Framework Adaptability

Analyzing the adaptability of the framework presented in the referenced paper is crucial. While it focuses on movie recommendation systems, extending it to different domains like e-commerce requires careful consideration.

In e-commerce, key challenges include balancing fairness across diverse product categories and ensuring visibility for a wide range of sellers. The method's dynamic fairness approach needs to address rapidly changing inventory and seller performance.

0.1.2 Dataset Overview: E-Commerce Purchase Dataset

We utilize the E-Commerce Purchase Dataset from Kaggle. This dataset provides detailed purchase records, capturing insights into customer behavior, product categories, and transaction details. It serves as a robust testbed for evaluating the dynamic fairness approach.

Key Features of the Dataset

- **Customer_ID:** Unique identifier for each customer (mapped to `user_id`).
- **Product_Category:** Category of the purchased product (mapped to `item_id`).
- **Purchase Details:** Includes transaction details such as amount spent, transaction date, and payment method.

0.1.3 Mapping of Variables

- `customer_id` corresponds to `user_id`.
- `product_category` corresponds to `item_id`.
- `rating` is derived from `clicks_in_site / time_on_site`.

0.1.4 Methodology

For our experiments, we used a history length $N = 1$ due to compatibility issues with larger N. Only 80 dataset entries supported $N = 2$, with fewer entries available for higher N. Training data consisted of 660 elements for $N = 1$. Although smaller, this dataset was chosen as it included both date and ratings, unlike most others.

The maximum number of recommendations (`max_k`) was set to 15, aligning with the 15 unique product categories in the dataset. The model was thus configured to recommend all categories at most once.

0.1.5 Results and Analysis

The model performed well within the constraints imposed by the dataset size and configuration.

Results indicate that the optimal performance for Precision is achieved when max_k = 5.

The trends for Recall, Hit Rate, and Precision closely mirror those observed in the movie dataset. While Gini and NDCG metrics show slight variations in their curves, their overall directional trends remain consistent with the movie dataset example.

- The similarity in trends for Recall, Hit Rate, and Precision across the two datasets suggests that the model generalizes well across domains (movies and e-commerce). This indicates that the core recommendation algorithm effectively captures user-item interactions regardless of the specific dataset context.
- Slight differences in Gini and NDCG curves could be attributed to the structural differences between the datasets.They highlight how user behavior and dataset diversity influence fairness and relevance metrics.

0.2 Individual Fairness

0.2.1 Objective

The primary objective of this improvement is to adapt and add user-based fairness to FCPO's already present group based fairness. While the original FCPO model incorporates an item cost indicator to ensure fairness in item recommendations, this project aims to extend this concept by introducing a user cost indicator.

The user cost indicator is designed to measure and account for disparities in user interactions and preferences, promoting fairness at the individual user level. By leveraging user interaction data, such as browsing time and engagement metrics, the model will dynamically balance the recommendations, ensuring both fairness and relevance.

This enhancement is expected to provide a more equitable recommendation system that addresses user-specific needs while maintaining the robustness of the original FCPO framework.

0.2.2 Results and Analysis

Figure 2 presents the evaluation metrics for individual fairness, measured across different values of max_k (the maximum number of recommendations).

Recall and Hit Rate

Both Recall and Hit Rate show an upward trend as max_k increases. This indicates that the model's ability to retrieve relevant items improves when more items are recommended, which is typical in recommendation systems. A higher max_k allows the model to cover a broader range of potentially relevant items, resulting in better Recall and Hit Rate values.

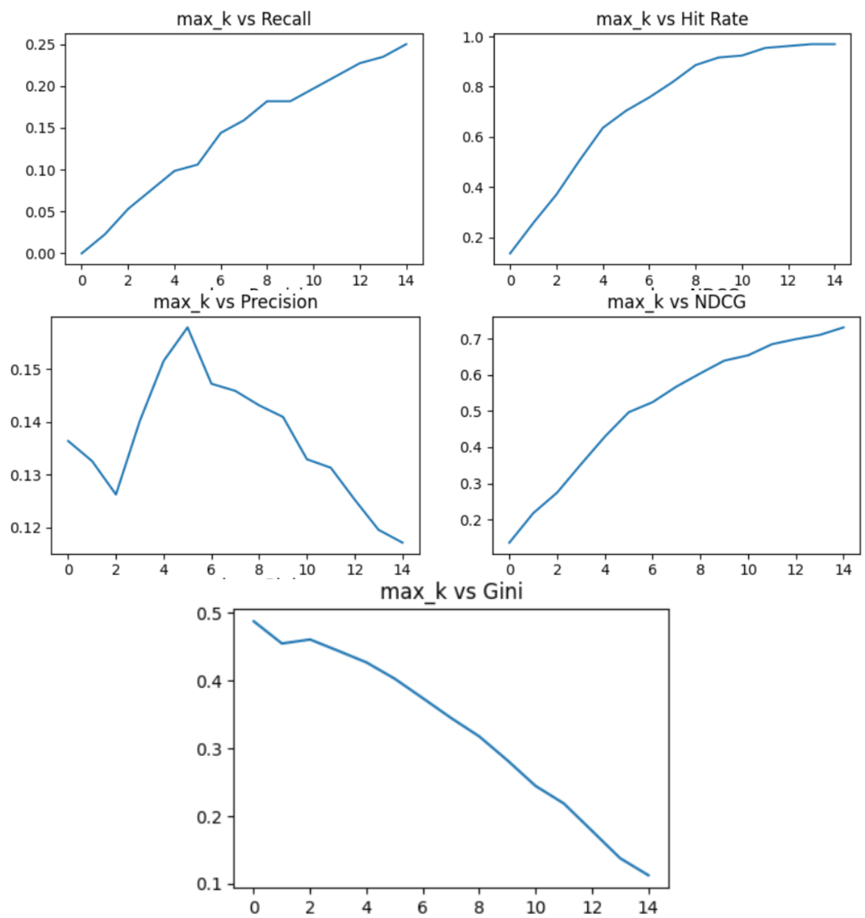

Figure 1: Evaluation metrics for application analysis.

Precision

Precision, however, decreases as max_k increases. This trend suggests that while the model retrieves more items with a larger max_k, the relevance of each individual recommendation decreases. This is expected, as increasing the number of recommendations often introduces less relevant items, thus lowering the proportion of correct recommendations among the total recommended items.

NDCG (Normalized Discounted Cumulative Gain)

The NDCG metric shows a consistent increase as max_k grows, indicating that the ranked relevance of recommended items improves slightly with a larger recommendation set. This improvement suggests that the model effectively ranks more relevant items toward the top of the recommendation list, even as max_k increases.

Gini Coefficient

The Gini coefficient decreases with an increase in max_k, reflecting improved fairness in item exposure. A lower Gini coefficient indicates a more equitable distribution, meaning the model distributes recommendations more evenly across different items. This decrease implies that the model avoids concentrating recommendations on a limited set of popular items, which is crucial for ensuring diversity and fairness.

Overall Analysis

The results indicate that the model effectively balances relevance and fairness. By adjusting max_k, it's possible to tune the model's behavior between maximizing user satisfaction (higher Recall and Hit Rate) and maintaining individual fairness (lower Gini coefficient). While a higher max_k reduces precision, the model successfully maintains an equitable recommendation distribution, which is critical in applications prioritizing fairness.

These metrics collectively illustrate the model's adaptability and highlight the trade-offs between accuracy and fairness, offering valuable insights for configuring the recommendation system based on specific application needs.

0.3 Code Enhancement Project

0.3.1 Code Extraction

The code was extracted from the original paper's GitHub repository: **FCPO GitHub Repository**.

0.3.2 Goals

1. **Code Simplification:**
    - Eliminate redundant classes and functions.
    - Streamline the code for better readability and maintainability.

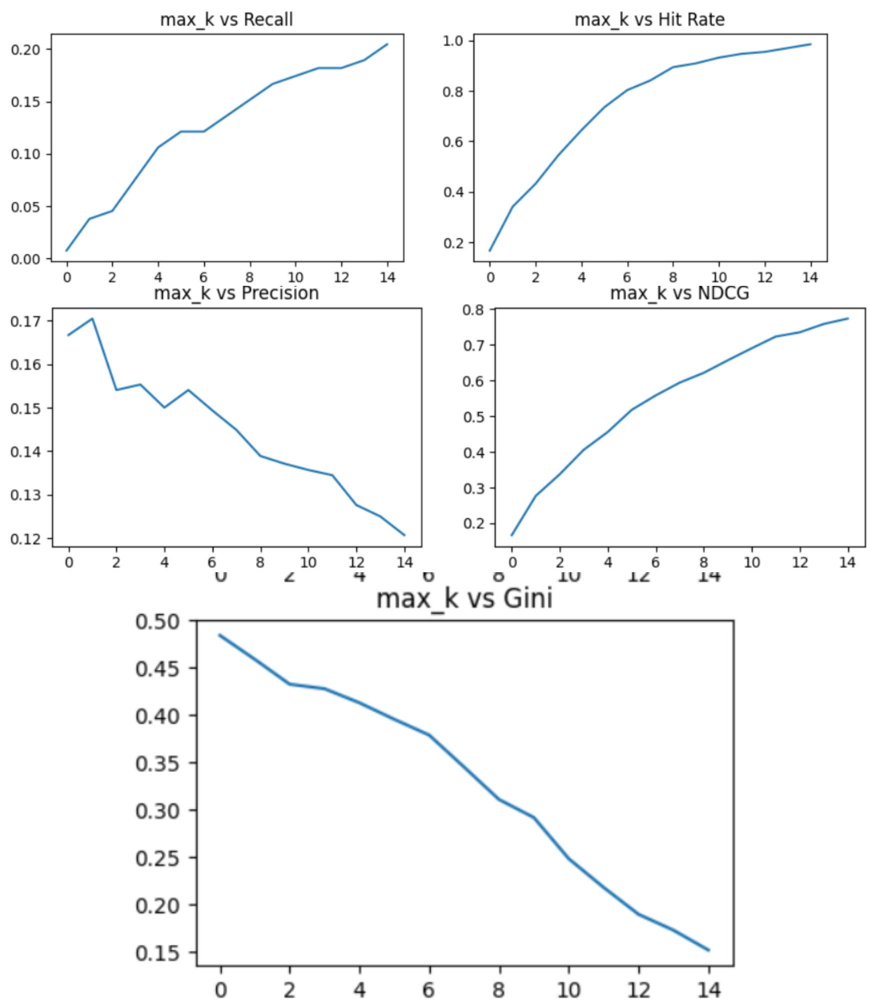

Figure 2: Evaluation metrics for Individual Fairness.

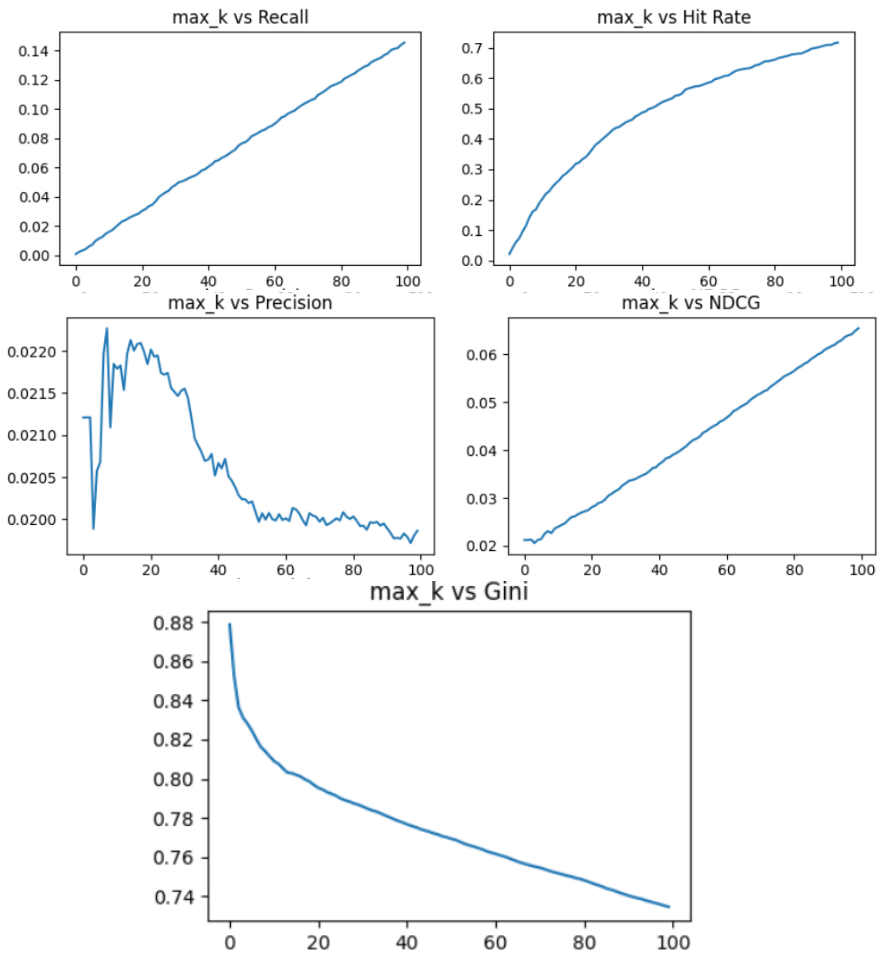

Figure 3: Evaluation metrics for Code Enhancement.

2. **Library Updates:**
   - Upgrade libraries to their latest versions.
   - Modify the code for compatibility with updated libraries.

0.3.3 Environment-Specific Issues

Python and library versions in Kaggle are periodically updated by the Kaggle team. Hence we updated the code to the latest version of libraries available in Kaggle. We also tested in Google Collab and successfully updated the code to globally latest libraries.

0.3.4 Kaggle Implementation

The updated notebook has been made available on Kaggle. Link to Kaggle Notebook.