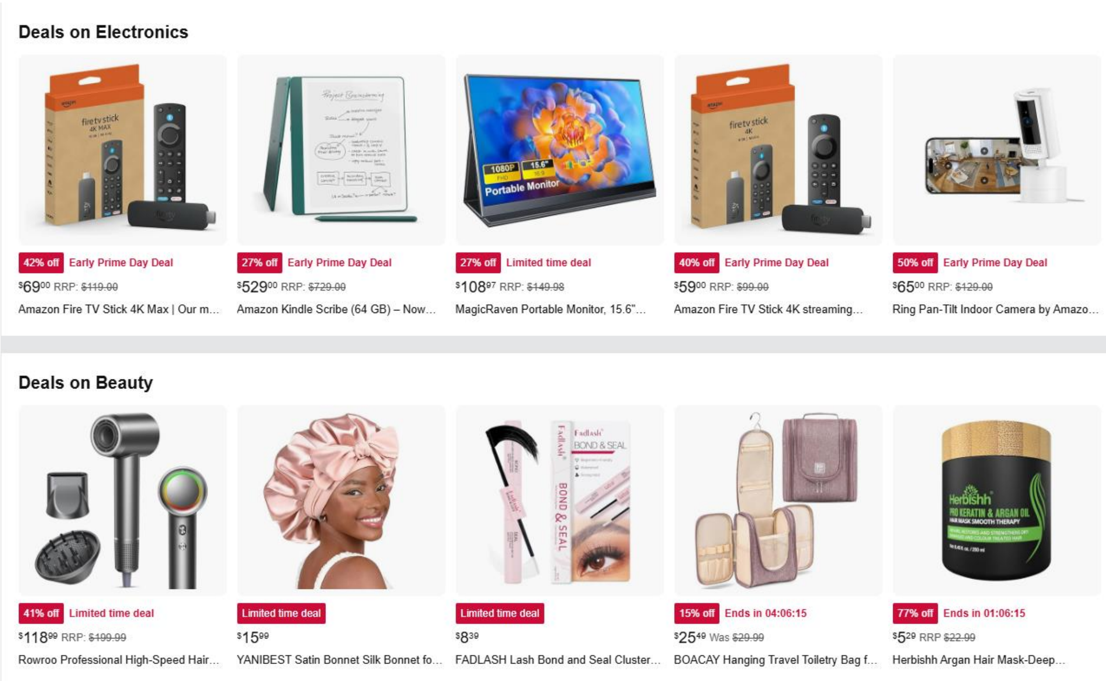
Design and Evaluation of a Hybrid Recommender System for Online Shopping Platforms

Wanting Xue – z5345106

Scope

The proposed recommender system will operate within the domain of online shopping, supporting an e-commerce platform similar to Temu or Amazon. The primary users of the system are general online shoppers, including both new and returning customers with diverse interests and purchasing behaviours. The system will aim to enhance the shopping experience by presenting the top 10 recommended items at a time through a web-based user interface. This interface will display each recommended item with its image, title, price, and action buttons such as “Add to Cart” and “Add to Wishlist.”

To illustrate the envisioned design of our recommender system's user interface, we include mockup image derived from sections of the Amazon website. These mockups serve purely as conceptual references to support the explanation of our system's intended functionality and layout; they do not represent the actual interface of our recommender system. In particular, our design extends typical e-commerce product displays by incorporating additional interactive elements at the bottom of each recommendation: a “Save to Wishlist” button, an “Add to Cart” button, and a “Not Interested” button. These features are intended to enable users to provide immediate feedback on recommendations, facilitating the collection of preference data that can be used to enhance future recommendations through adaptive learning.



User feedback will be collected primarily through implicit signals, such as clicks on recommended items, additions to the shopping cart, wishlist additions, and purchases. This feedback will be continuously logged and used to improve the recommendation model. In dynamic scenarios, the system will support both batch updates and real-time adjustments. Batch updates will retrain the recommendation model periodically (e.g., daily or hourly) based on the latest interaction data. Real-time signals, such as surges in product popularity, will trigger lightweight updates or adjustments to recommendation lists to ensure timely responses to emerging trends. The cold-start problem for new users will be addressed by combining popularity-based recommendations with optional onboarding preference selections, such as preferred categories or price ranges. For new products, the system will use content-based features, including category, brand, and textual description embeddings, to generate initial recommendations while gradually integrating these items into collaborative models as interaction data becomes available. The business model will derive revenue from increased sales conversions, higher average order values through cross-selling and bundle recommendations, the promotion of long-tail items, and sponsored product placements within the recommendation list.

Dataset

For this project, we propose using the Amazon Product Data (2018) dataset (https://nijianmo.github.io/amazon/index.html), specifically focusing on the Clothing, Shoes and Jewelry category. This category offers a rich and diverse set of products with over 11 million reviews in the 5-core subset, where each user and product has at least five reviews. This ensures that our recommendation models can be trained on meaningful, dense interaction data, alleviating issues related to data sparsity that often limit collaborative filtering methods.

The dataset provides detailed user-item interactions, including ratings, review texts, timestamps, helpfulness votes, and user-uploaded images. Additionally, product metadata includes attributes such as brand, category hierarchy, price, style features (e.g., color, size), descriptions, bullet-point features, and co-purchase/co-viewing graphs. These features enable the development of both collaborative and content-based recommendation models.

Our preprocessing plan includes:

a) Restricting data to the Clothing, Shoes and Jewelry 5-core subset, with optional further sampling if computational feasibility requires it, to balance data richness with manageability.
b) Cleaning data by removing reviews with missing critical fields (e.g., ratings, product IDs) and standardizing category and brand labels.
c) Transforming timestamps into structured temporal features (e.g., recency scores).
d) Generating structured features for product metadata (e.g., one-hot encoding of brands, TF-IDF vectors for descriptions).
e) Constructing a user-item rating matrix for collaborative filtering, supplemented by implicit feedback signals where available.
f) (Optional) Extracting image embeddings using pretrained convolutional neural networks if computational resources permit.

While the dataset is large, our use of the 5-core subset at the category level ensures that data volume is realistic yet manageable. However, it is important to note potential limitations:
(i) The dataset may contain sampling bias as it represents only a subset of Amazon users and products.

(ii) Some metadata fields may have missing or inconsistent values.
(iii) Interaction data may not fully capture dynamic user preferences in real-time scenarios.

By addressing these challenges through preprocessing and careful method design, we aim to build and evaluate models that reflect realistic recommendation conditions while remaining computationally tractable.

Methods

To address the recommendation problem for the online shopping domain, we propose implementing and evaluating three classes of methods: collaborative filtering, content-based filtering, and hybrid models that combine both approaches.

For collaborative filtering, we will use matrix factorization techniques (such as SVD or Biased Matrix Factorization) on the user-item rating matrix constructed from the overall ratings field. This method leverages user preferences expressed through explicit ratings to learn latent factors representing user interests and product characteristics. The dense 5-core dataset ensures that collaborative filtering will have sufficient data to identify meaningful patterns.

For content-based filtering, we will build models that recommend products based on their similarity to items the user has rated highly in the past. We will extract product features from metadata, including brand, category, price (normalized), and style attributes (e.g., size, color). Textual features such as product titles, bullet-point descriptions, and full product descriptions will be transformed using TF-IDF vectorization or pretrained word embeddings to capture semantic similarities. This approach is particularly valuable for addressing cold-start scenarios where new products lack sufficient interaction history.

For hybrid recommendation, we propose two strategies. The first is a score-level hybrid, combining collaborative and content-based predictions through weighted averaging or model stacking, where the weight parameters are tuned on a validation set. The second is a feature-level hybrid using neural network models that concatenate latent factors from collaborative filtering with content-based features to learn joint representations that can capture both user preference patterns and item attributes. This approach is expected to provide robust performance across both warm-start and cold-start scenarios.

All methods will be implemented using scalable machine learning libraries such as scikit-learn, surprise, and TensorFlow/Keras. We will carefully balance model complexity and computational efficiency to ensure that training and evaluation can be conducted within the project timeline. Where possible, we will explore incremental or online learning strategies to enable dynamic model updates in response to new user interactions.

Evaluation

The evaluation of the recommender system will encompass both offline model performance metrics and user-centered system assessment to ensure a comprehensive understanding of its effectiveness and usability.

For offline evaluation, we will employ a variety of well-established metrics on historical data. These include Precision, Recall, and F1-score to measure the accuracy of recommended items relative to user preferences. In particular, we will focus on top-N metrics such as Precision@10 to reflect the real-world scenario where users are presented with a limited number of recommendations. Additionally, Root Mean Square Error (RMSE) or Mean Absolute Error (MAE) will be used to assess the accuracy of predicted ratings when applicable. To ensure the recommender provides diverse and novel suggestions, we will also calculate coverage, novelty, and diversity metrics, which measure the breadth and freshness of recommendations.

Beyond model accuracy, we will consider the recommender system's impact on user experience through a small-scale user study. This study will simulate user interactions via a prototype interface, recruiting approximately 20 participants to mimic user browsing, clicking, and purchasing behaviours. Quantitative measures such as click-through rate (CTR) and conversion rate will be tracked during these simulated interactions to provide insight into user engagement and recommendation effectiveness. Furthermore, we will monitor system response times to ensure recommendations can be generated in near real-time, which is crucial for dynamic user scenarios.

The final assessment will balance multiple performance metrics with user behaviour outcomes to select the best-performing recommendation approach. Consideration will also be given to computational efficiency and scalability to confirm the system's practicality for deployment.

Time Estimates and Milestones of Anticipated Progress

Given the project duration of four weeks (Week 6 to Week 10), we propose the following detailed timeline to ensure steady progress and timely completion of the recommender system:

Week 6: Data Preparation and Initial Modeling

The first week will focus on completing data preprocessing, including cleaning, feature engineering, and dataset partitioning. We will also develop baseline recommendation models such as collaborative filtering and content-based filtering to establish initial benchmarks.

Week 7: Advanced Model Development and Hybrid System Integration

In the second week, we will implement and fine-tune more advanced recommendation techniques, including hybrid models that combine collaborative and content-based methods. Hyperparameter tuning and validation will be performed to optimize model performance.

Week 8: Dynamic Updating and Cold Start Solutions

This week will be dedicated to designing and integrating dynamic update mechanisms for the recommender system to handle new data in near real-time. We will also implement strategies to address the cold start problem for new users and items, ensuring robustness.

Week 9: User Study Design and System Evaluation

We will design and conduct a small-scale user study with simulated interfaces to collect user feedback on recommendation quality and usability. Simultaneously, comprehensive offline evaluations using multiple metrics will be carried out to assess model effectiveness.

Week 10: Finalization and Reporting

The final week is reserved for consolidating all results, refining the recommender system based on evaluation feedback, and preparing the final project report and code documentation for submission.