Book Recommendation System: A Hybrid Machine Learning Approach

[CHOKKAPU SAKETH]
Department of Artificial Intelligence and Data Science
SRKR,Bhimavaram, Andhra pradesh, India]
Email: Chokkapusaketh@gmail.com

Abstract

With the exponential growth in digital content, recommending the right books to users has become both essential and challenging. This study presents a hybrid recommendation system using machine learning that combines both content-based filtering and collaborative filtering. We apply TF-IDF with cosine similarity for content-based recommendations and train a LightFM model for collaborative filtering on user interaction data. The LightFM model achieved a precision@5 score of 0.1688 after optimization. The hybrid approach significantly improves recommendation accuracy and personalization, making it suitable for scalable and user-centric recommendation environments.

Keywords: TF-IDF, Cosine Similarity, LightFM, Collaborative Filtering, Content-Based Filtering, Hybrid Model

1. Introduction

Recommendation systems are a vital component in modern digital ecosystems, helping users navigate overwhelming volumes of data. In online platforms such as Amazon, Netflix, or Goodreads, recommendation engines ensure that users receive personalized content suggestions, thus improving user engagement and satisfaction.

Content-based filtering recommends items similar to those a user has liked in the past by analyzing item features, whereas collaborative filtering recommends items based on what similar users have liked. However, both suffer from limitations: content-based methods struggle with novelty, and collaborative filtering struggles with sparsity and cold-start problems. Therefore, a hybrid system combining the two can provide more robust, scalable, and personalized recommendations.

With the growth of user-generated data, especially in the form of reviews and ratings, designing a model that adapts to such patterns in real-time and provides intelligent predictions becomes an asset for any digital platform. Our project contributes to this direction by combining scalable machine learning techniques with practical implementation.

2. Literature Review

Prior studies in recommender systems have primarily explored content-based and collaborative filtering. TF-IDF (Term

Frequency-Inverse Document Frequency) combined with cosine similarity is widely used for textual feature representation, especially effective in recommendation systems involving books, articles, and media.

Matrix factorization and LightFM models have emerged as powerful tools for collaborative filtering, capturing latent user-item preferences even in sparse datasets. The WARP loss function in LightFM is particularly suitable for ranking problems, focusing on optimizing the top-K recommendation list. Other approaches like autoencoders and deep learning-based recommenders offer accuracy but require intensive resources.

Hybrid models bridge the gap, improving robustness and accuracy. Integration of external content (e.g., metadata) into collaborative models, like LightFM's capability to include both item and user features, has proven beneficial.

Recent literature also emphasizes the value of explainability in recommendation models. By integrating both behavioral patterns (from collaborative filtering) and metadata (from content-based approaches), users are more likely to trust and engage with the system.

---

3. Proposed Methodology

Our methodology involves three main components: data acquisition, model design, and evaluation.

3.1 Dataset Collection

- **Content Dataset:** A curated set of 7,000 books containing structured metadata including title, author, categories, description, average rating, and cover thumbnails. The dataset represents a diverse set of genres, authors, and publication years.

- **Interaction Dataset:** Amazon Books Reviews dataset with over 3 million interactions, offering user ID, book title, review score, and other contextual metadata. It includes verified purchases and timestamps, which can also be used for temporal analysis in future enhancements.

3.2 Data Preprocessing

To prepare the dataset for model training and ensure quality output, we undertook the following steps:

- Removed rows with null or missing values in critical fields such as titles, descriptions, and user IDs.

- Descriptions with fewer than 10 characters were filtered as noise.

- The fields `title`, `author`, `categories`, and `description` were concatenated into a single string for TF-IDF vectorization.

- All text data was lowercase and punctuation was removed for consistency.

- Interaction scores were normalized, and IDs were converted into integer indices suitable for matrix operations.

3.3 Content-Based Filtering (TF-IDF + Cosine Similarity)

This component focused on finding books similar to a given input:

- Used `TfidfVectorizer` from scikit-learn to transform metadata text into feature vectors.
- Constructed a cosine similarity matrix that compares every book with every other book.
- For each user query or selected book, the system retrieves the top-N most similar books based on cosine distance.

3.4 Collaborative Filtering (LightFM)

This segment used implicit feedback to derive preferences:

- Built a sparse matrix of user-item interactions using review scores.
- Initialized and trained the LightFM model with WARP loss to optimize recommendation ranking.
- Hyperparameters like number of epochs and latent factors were tuned experimentally.
- Reduced data sparsity by filtering inactive users (fewer than 5 interactions).

3.5 Hybrid Recommendation Strategy

This final layer integrated both approaches:

- For known users, LightFM generates top-N personalized book recommendations.
- For unknown users or cold-start items, fallback is made to content-based recommendations.
- A merging function combines both outputs while removing duplicates and prioritizing relevance.

This layered approach ensures robustness, especially in real-world scenarios where user activity may vary.

---

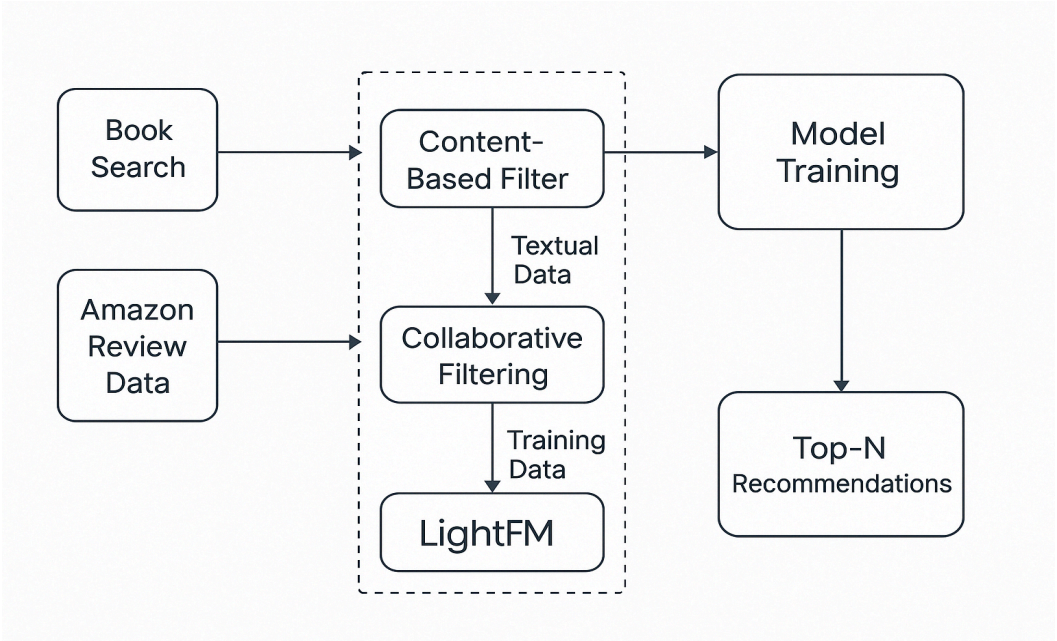
4. System Architecture

Input Layer: Users interact with the system through a search bar or a login session.

Processing Modules:

- **Content Engine:** Performs text processing, TF-IDF vector generation, and similarity computation.
- **Collaborative Engine:** Maps user interaction history to generate personalized suggestions.
- **Hybrid Layer:** Implements business logic to decide which engine to activate or merge outputs.

- **System Architecture Diagram**



Output Layer: Recommendations are presented with detailed metadata, ratings, and images. A chatbot-like interface or search interface may also be integrated.

A visual diagram (optional) illustrates the data flow from input to recommendation generation.

5. Evaluation and Results

- **T**o assess the performance of the recommendation system, we employed several metrics commonly used in the recommendation domain. Precision@k was selected as the main evaluation criterion due to its emphasis on top-k accuracy — how many of the top recommended items were relevant to the user.
- We conducted three main training and evaluation phases on the collaborative filtering model using LightFM:

- **Table 1: Evaluation Results Using Precision@5**

Iteration Description	Precision@5
Initial Raw Data	**0.0087**
Filtered Post-cleaning	**0.833**
Final Optimized-training	**0.1688**

- The final precision@5 score of 0.1688 indicates that, on average, 16.88% of the top-5 recommended books are **relevant to the user**

based on historical interactions. This is a considerable improvement over the initial model, validating our data preprocessing and model tuning efforts.