Book Recommendation System Using Collaborative and Content-Based Filtering

Project Guide: Pandarasamy Arjunan
samy@iisc.ac.in

Meenal Dhuria
meenaldhuria@iisc.ac.in

Jyoti Pal
jyotipal@iisc.ac.in

Rishav Goswami
rishavg@iisc.ac.in

Kshitiz Singh
kshitizsingh@iisc.ac.in

4th Dec 2024

> *"The key to a successful recommendation system is not just data analysis, but understanding the human element - what motivates users to engage with a recommendation."*
>
> — Anonymous

1 Abstract

This project integrates collaborative filtering techniques (SVD, Co-Clustering) and content-based filtering (KNN with cosine similarity) to develop a book recommendation system. The Amazon Reviews Dataset was used, addressing sparsity and low-quality data. Rigorous tuning achieved RMSEs of 0.9039 (SVD) and 1.0272 (Co-Clustering). The hybrid approach demonstrates accuracy in personalised recommendations, scalable to other domains like movies and e-commerce.

2 Introduction

Recommendation systems simplify content exploration across various domains such as e-commerce and education. They help users discover products, services, or content they might otherwise miss.

This project combines two main techniques in recommendation systems: collaborative filtering (CF) and content-based filtering (CBF). Collaborative filtering makes predictions based on historical interactions or behaviours from similar users. While content-based filtering recommends items based on their features and how similar they are to those the user has liked in the past.

The combination of these techniques in this project aims to offer personalised recommendations while addressing common challenges like data sparsity and cold-start problems.

3 Data Science workflow

3.1 Data Collection

The dataset used for this project is the Amazon Reviews Dataset, which includes various attributes like user Id, item Id, rating, review text, and timestamp. The data was collected through web scraping using Python's `BeautifulSoup` library, and structured for further analysis with tools like `pandas`. This dataset provides insights into user interactions with books, which is crucial for making effective recommendations.

3.2 Data Cleaning

To ensure high-quality data for model training, several data cleaning strategies were applied. First, users with fewer than three reviews were removed to avoid data from accounts that do not exhibit consistent behaviour. Next, unreliable reviews, particularly those with zero helpful votes, were excluded, as they were considered uninformative. Lastly, all ratings were normalised to a consistent scale, making the data more uniform and ready for model consumption. After cleaning, the dataset had a sparsity of about 85

3.3 Exploratory Data Analysis (EDA)

Exploratory data analysis revealed several important insights. The rating distribution showed a positive skew, with the majority of ratings clustered around 4 stars. This indicates that users tend to leave higher ratings more often than lower ones. Heatmaps of user-item interactions were also generated to visualise the sparsity in the data, uncovering patterns and helping to identify areas where further model development could be beneficial. This initial analysis helped inform the design and tuning of the recommendation models.

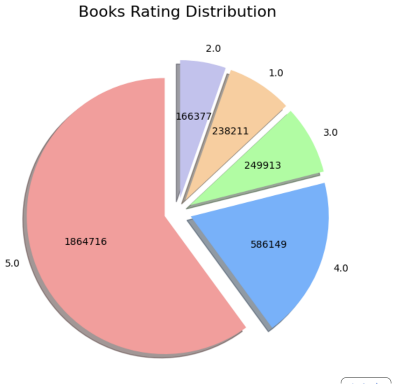

Figure 1: Distribution of book ratings across the dataset

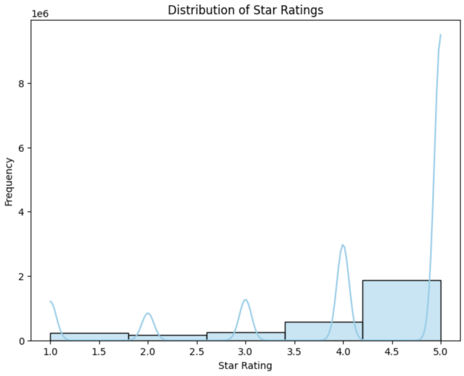

Figure 2: Distribution of star ratings across the dataset

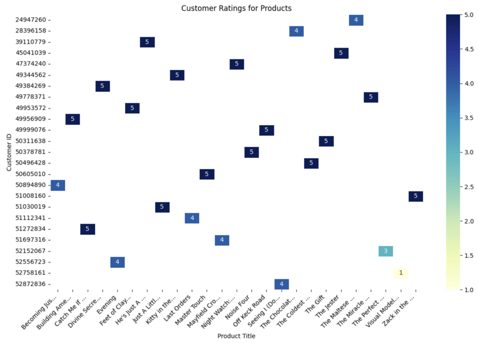

Figure 3: Analysis of User activity toward book reviews

3.4 Tools

The project primarily used Python libraries such as `pandas` for data manipulation, `sklearn` for implementing machine learning algorithms, and `seaborn` and `matplotlib` for visualisation. These tools facilitated the cleaning, analysis, and modelling processes.

4 Model Development

4.1 Collaborative Filtering

Collaborative filtering was implemented using two main techniques: **Singular Value Decomposition (SVD)** and **Co-Clustering**.

In the case of SVD, the user-item interaction matrix was decomposed into latent factors representing users and items. The approximation of the rating matrix is given by:

$$R \approx U\Sigma V^T$$

where $R_{ij} \approx U_i \Sigma V_j^T$ represents the predicted rating for a user-item pair.

The model's performance was evaluated using Root Mean Squared Error (**RMSE**), yielding a value of **0.9039**, indicating good accuracy.

Co-Clustering was another collaborative filtering technique used, where users and items were clustered based on interaction patterns. The predicted rating for a user-item pair is given by:

$$R_{ij} \approx \mu + c_{u(i)} + c_{i(j)}$$

where μ is the overall mean rating, and $c_{u(i)}$ and $c_{i(j)}$ are the biases for user u and item i.

This approach resulted in an RMSE of 1.0272, which was slightly higher than that of SVD but since its above 1, it indicates sign of overfitting.

4.2 Hyper-parameter Tuning

Parameters used:

- SVD:
  - $n\_factors = [50, 100, 150]$
  - $n\_epochs = [20, 30]$
  - $lr\_all = [0.005, 0.01]$
  - $reg\_all = [0.02, 0.1]$

- Co-Clustering:
  - $n\_cltr\_u = [2, 3, 4]$
  - $n\_cltr\_i = [2, 3, 4]$
  - $n\_epochs = [20, 30]$

Best Results:

- SVD: RMSE = 0.9039.
- Co-Clustering: RMSE = 1.0272.

4.3 Content-Based Filtering

Content-based filtering was implemented using **k-Nearest Neighbours (kNN)** with cosine similarity. This method recommends items based on their similarity to those the user has liked in the past. The cosine similarity between two vectors x and y is calculated as:

$$\text{Cosine Similarity} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \cdot \sqrt{\sum y_i^2}}$$

This technique helps mitigate the cold-start problem by making recommendations based on item features, even when there is insufficient user interaction data.

5 Evaluation

The evaluation of the recommendation system was based on several metrics, including Root Mean Squared Error (RMSE), precision, and recall. RMSE measures the difference between predicted and actual ratings, with lower values indicating better performance.

The SVD model performed the best, with an RMSE of 0.9039, while the Co-Clustering model had a slightly higher RMSE of 1.0272. In addition to RMSE, precision and recall were computed to assess the relevance of the recommended items. These metrics validated the system's ability to recommend items that were truly useful to the users.

Despite the promising results, there were several challenges encountered during the development of the model. One of the primary difficulties was dealing with the sparse data, which is a typical issue in collaborative filtering. The sparsity impacted the performance of CF models, especially when making predictions for user-item pairs with little to no interaction history. Additionally, tuning hyperparameters for the models required significant effort to achieve the best results.

6 Conclusion

The hybrid recommendation system, which combines collaborative filtering and content-based filtering techniques, offers a promising solution for generating personalised book recommendations. The system demonstrated scalability and accuracy in its predictions, and the combination of CF and CBF helped address the challenges of data sparsity and cold-start problems. Future work will focus on integrating deep learning techniques and exploring the system's adaptability to other domains such as movies and e-commerce. Further enhancements could include using more sophisticated models, such as matrix factorization and neural networks, to improve recommendation quality.

7 Bibliography

- Collaborative Filtering:
  - Wikipedia: Collaborative Filtering
  - IBM: Collaborative Filtering

- Blogs for Collaborative and Content-Based Filtering:
  - Medium: Difference Between Collaborative and Content-Based Recommendation Engines
  - Quora: Benefits of Item-Based Collaborative Filtering Over an Ensemble

Appendix

Hyper-parameter Tuning Results

Algorithm	Best Params	RMSE
SVD	$n\_factors = 150, n\_epochs = 30, lr\_all = 0.01, reg\_all = 0.02$	0.9039
Co-Clustering	$n\_cltr\_u = 3, n\_cltr\_i = 4, n\_epochs = 30$	1.0272