Book Recommendation System Using Collaborative And Content-Based Filtering

Subject: Data Science and Practice

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

4^{th} Dec 2024

Contents

Chapter 1

Abstract

"In the digital age, the right information at the right time can make all the difference."

— Anonymous

This project presents a comprehensive book recommendation system by integrating collaborative filtering techniques (SVD and Co-Clustering) and content-based filtering (using kNN with cosine similarity). The system utilises the Amazon Reviews Dataset, cleaned and preprocessed to address low-quality reviews and user sparsity issues. Exploratory data analysis (EDA) uncovered significant insights to inform feature engineering.

Rigorous hyper-parameter tuning was applied, achieving an RMSE of 0.9039 with SVD and 1.0272 with Co-Clustering.

The kNN model complements collaborative filtering by analysing item similarity, mitigating cold-start challenges. Applications extend beyond books to other domains like movies and e-commerce. The results highlight the effectiveness of hybrid approaches in delivering accurate and personalised recommendations.

Chapter 2

Introduction

"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

Recommendation systems have become indispensable in the digital era, enabling users to efficiently explore vast content across various domains. By forecasting user preferences and suggesting pertinent items, these systems have reshaped industries such as e-commerce, entertainment, education, and healthcare. Developing effective recommendation systems requires integrating machine learning, data science, and domain expertise to accurately model and predict user behaviours.

This project leverages a hybrid approach, combining collaborative filtering (CF) for personalised recommendations based on user behaviour and content-based filtering (CBF) for item similarity. Together, these methods enhance recommendation accuracy, even in the presence of sparse data and limited interactions.

Chapter 3

Data Science Workflow

3.1 Data Collection

Source: The dataset used for this project was sourced from the following repository:

- Dataset Link: https://amazon-reviews-2023.github.io/
- This dataset comprises detailed Amazon product reviews and metadata, providing a robust foundation for building a recommendation system.
- The dataset's structure includes information such as user IDs, item IDs, ratings, review texts, and timestamps, which were crucial for both collaborative filtering and content-based filtering approaches.

Tools and Libraries:

- Web APIs and web scraping techniques
- requests and BeautifulSoup

3.2 Data Cleaning and Preparation

Strategies:

- Removed users with fewer than three reviews to avoid outlier noise.
- Excluded reviews with zero or negligible "helpful votes" for data reliability.
- Imputed missing ratings using user or item averages.
- Normalised ratings for consistent scaling across datasets.

Tools and Libraries:

- Pandas
- Python
- Sklearn

Outcome: Cleaned data with 85% sparsity, validated for consistent user-item mapping.

3.3 Exploratory Data Analysis (EDA)

The following action items were performed during the EDA phase:

1. **Statistical Summary**: Generated descriptive statistics for numerical features to understand data distributions.
2. **Correlation Analysis**: Investigated relationships between numerical columns to identify possible associations.
3. **Visualization**: Created various plots such as histograms, bar charts, and heat-maps to visualise distributions and correlations between key variables.
4. **Outlier Detection**: Identified and handled outliers in numerical data using box plots.
5. **Feature Engineering**: Created new features (e.g., rating counts, helpful votes) to aid in model performance.

Tools and Libraries:

- Language: Python
- Libraries: Pandas, Sklearn etc
- Visualisation tool: matplotlib, seaborn etc

Key Findings:

- Rating distribution showed a positive bias with a peak at 4.
- Sparsity heat-maps confirmed high data sparsity but revealed actionable patterns.

Visualizations:
To better understand the dataset, several visualisations were created. These provide insights into the data distribution, user activity, item ratings, and other key patterns.

3.3.1 Distribution of Book Ratings

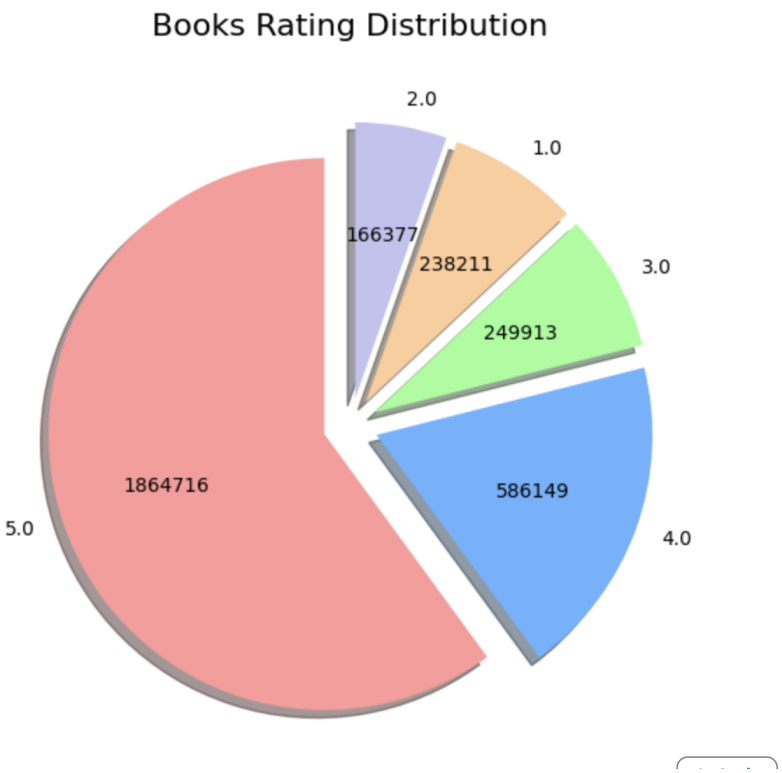

Figure 3.1: Distribution of book ratings across the dataset

3.3.2 Star Rating Distribution

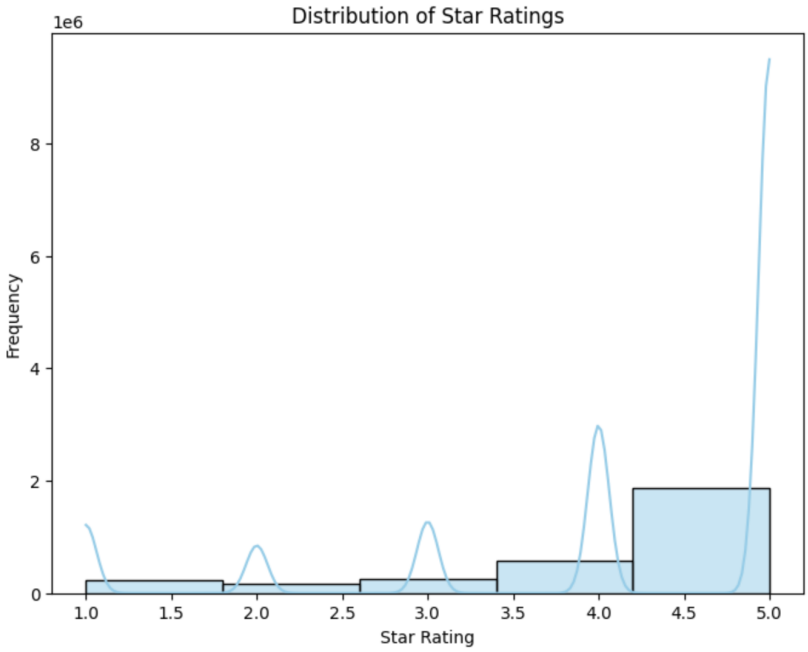

Figure 3.2: Distribution of star ratings across the dataset

3.3.3 User Activity Analysis

Figure 3.3: Analysis of User activity toward book reviews

Chapter 4

Model Development

4.1 Collaborative Filtering

4.1.1 Singular Value Decomposition (SVD)

Mechanism: SVD reduces the dimensionality of the user-item matrix by identifying latent factors representing users and items.

Mathematics:

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

where R: User-item matrix, U, V: Latent user and item matrices, Σ: Singular values. Predicted rating:

$$\hat{R}_{ij} = U_i \Sigma V_j^T$$

Why SVD?

- Effectively handles sparsity.
- Captures nuanced user-item relationships through latent factors.

4.1.2 Co-Clustering

Mechanism: Co-Clustering groups users and items into clusters based on shared patterns. Recommendations derive from cluster-level interactions.

Mathematics:

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

where $c_{u(i)}$ and $c_{i(j)}$ denote user and item cluster effects.

Why Co-Clustering?

- Provides interpretable cluster-level recommendations.
- Handles small datasets effectively.

4.1.3 Hyper-parameter Tuning and Results

Parameters:

- 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.

4.2 Content-Based Filtering with kNN

Mechanism: kNN recommends items based on their similarity in feature space, leveraging metadata like genres and authors.

Similarity Measure: Cosine Similarity:

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}$$

Why kNN?

- Effective for cold-start problems.
- Provides interpretable results based on explicit features.

Implementation:

- Items represented as vectors in a high-dimensional space.
- Nearest neighbours identified for each item based on cosine similarity.
- Recommendations generated from top neighbours.

4.3 Tools and Libraries:

- Pandas
- Python
- Sklearn
- Suprise
- Scipy
- Ipywidgets
- Random

Chapter 5

Evaluation and Observations

5.1 Metrics

The evaluation of the models was conducted using the following metrics:

- **Root Mean Square Error (RMSE):** Used to measure prediction accuracy.
  - **SVD:** RMSE = **0.9039** with optimal parameters:
    - `n_factors`: 150
    - `n_epochs`: 30
    - `lr_all`: 0.01
    - `reg_all`: 0.02
  - **Co-Clustering:** RMSE = **1.0272** with optimal parameters:
    - `n_cltr_u`: 3
    - `n_cltr_i`: 4
    - `n_epochs`: 30
- **Precision and Recall:** Used for ranking evaluation to assess the relevance of recommendations.

The metrics indicate that SVD outperformed Co-Clustering in prediction accuracy. Additionally, precision and recall analyses validated the system's ability to provide relevant and personalised recommendations.

5.2 Challenges and Observations

Challenges:

- Sparse data in collaborative filtering.
- Uneven distribution of ratings and metadata.
- High computational cost for hyper-parameter tuning.

Observations:

- SVD effectively identified hidden patterns.
- Co-Clustering provided meaningful cluster-level insights.
- kNN tackled cold-start problems using metadata effectively.

Chapter 6

Conclusion

This project successfully demonstrates the value of combining collaborative and content-based filtering for building an effective recommendation system. The SVD model emerged as the most effective for accurate predictions, while kNN ensured reliable recommendations for content-driven queries. The findings suggest that future work could focus on exploring hybrid models to mitigate challenges such as sparsity and cold-start problems, paving the way for more robust and scalable recommendation systems.

"The best way to predict the future is to create it."

— Abraham Lincoln

Chapter 7

Bibliography

- Collaborative Filtering:
  - Wikipedia: https://en.wikipedia.org/wiki/Collaborative_filtering
  - IBM Topic Page: https://www.ibm.com/topics/collaborative-filtering
- Blogs for Collaborative and Content-Based Filtering:
  - Medium Blog: https://muvi-com.medium.com/the-difference-between-collaborative-and-content-based-recommendation-engines
  - Quora Discussion: https://www.quora.com/unanswered/What-are-the-benefits-of-using-a-single-item-based-collaborative-filtering

Appendices

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