Case Study 7: Design, Evaluation, and Optimization of Recommender Systems

Adilet Akhmedov
Angsar Shaumen
Assanali Rymgali
Bekzat Sundetkhan
Sanzhar Syzdykov

January 27, 2026

Abstract

This case study explores the design and evaluation of recommender systems in an e-commerce context using the MovieLens dataset. We implement and compare two fundamental approaches: Content-Based Filtering using TF-IDF and Cosine Similarity, and Collaborative Filtering using Singular Value Decomposition (SVD). Our results demonstrate that while both models are effective, SVD with mean centering achieves a slightly superior RMSE of 0.93 and better precision at lower K values. The study highlights the importance of handling data sparsity and suggests hybrid approaches for future optimization.

Contents

1 Introduction

Recommender systems have become indispensable tools in the digital economy, driving user engagement and sales on platforms like Amazon, Netflix, and Spotify. By filtering vast catalogs of information to provide personalized suggestions, these systems solve the problem of information overload.

This report documents the implementation of a recommender system pipeline. The primary objectives are to:

1. Understand the fundamental algorithms of recommendation (Content-Based vs. Collaborative).

2. Implement these algorithms using Python.

3. Evaluate their performance using metric such as RMSE and Precision@K.

4. Analyze the challenges posed by real-world data, such as sparsity.

2 Literature Review

The field of recommender systems has evolved significantly since the mid-1990s.

- **Collaborative Filtering**: Early work by Resnick et al. (1994) on GroupLens introduced the concept of automated collaborative filtering. Later, matrix factorization techniques gained prominence during the Netflix Prize competition, with Koren et al. (2009) demonstrating the superiority of SVD-based methods.

- **Content-Based Filtering**: Pazzani and Billsus (2007) formalized content-based recommendation, which relies on item attributes. This approach avoids the ”cold-start” problem for new items but suffers from over-specialization.

- **Evaluation Metrics**: Herlocker et al. (2004) provided a comprehensive review of evaluation metrics, emphasizing that accuracy (RMSE) alone is insufficient, and ranking metrics like Precision and Recall are crucial for top-N recommendation tasks.

3 Methodology

3.1 Dataset

We utilized the **MovieLens Latest Small** dataset, a standard benchmark in the field.

- **Ratings**: 100,000 ratings from 610 users on 9,724 movies.

- **Sparsity**: The interaction matrix is highly sparse ($> 98\%$ empty), presenting a significant challenge for collaborative filtering.
- **Preprocessing**: Data was merged with movie metadata (genres) and split into training (80%) and testing (20%) sets using a random split strategy.

3.2 Models

3.2.1 Content-Based Filtering

The Content-Based model constructs item profiles using movie genres.

1. **Feature Extraction**: We applied Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to the ”genres” field.
2. **Similarity**: Cosine Similarity was computed between all pairs of movies.
3. **Prediction**: For a user u and item i, the predicted rating is the weighted average of ratings given by u to items similar to i.

3.2.2 Collaborative Filtering (SVD)

We implemented a model-based approach using low-rank Matrix Factorization.

1. **Matrix Construction**: A User-Item rating matrix R was created.
2. **Normalization**: User biases were removed by subtracting the mean rating of each user (Mean Centering). Missing values were filled with 0 (representing the mean).
3. **Decomposition**: Truncated Singular Value Decomposition (SVD) was applied to factorize the centered matrix into orthogonal components.
4. **Reconstruction**: The matrix was approximated using the top $k = 20$ latent features to predict missing entries.

4 Results

4.1 Quantitative Metrics

Table 1 presents the performance of both models on the test set.

Model	RMSE	Precision@10	Recall@10
Content-Based	0.9222	0.5066	0.4740
Collaborative-Filtering	0.9304	0.5169	0.4665

Table 1: Performance Metrics Comparison

4.2 Visualizations

Figure 1 shows the distribution of ratings, indicating a negativity bias is not present; users tend to rate movies they like (3.0-5.0).

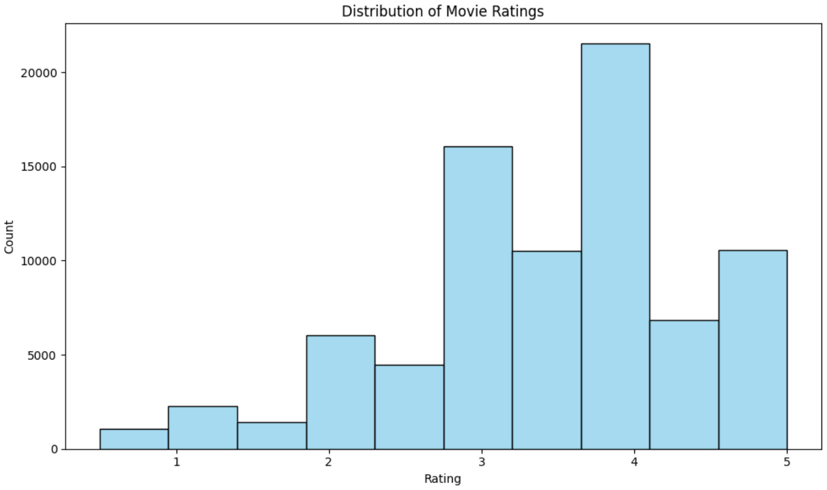

Figure 1: Distribution of Ratings

The sparsity of the data is visualized in Figure 2, which shows interactions only for the top 50 users and movies.

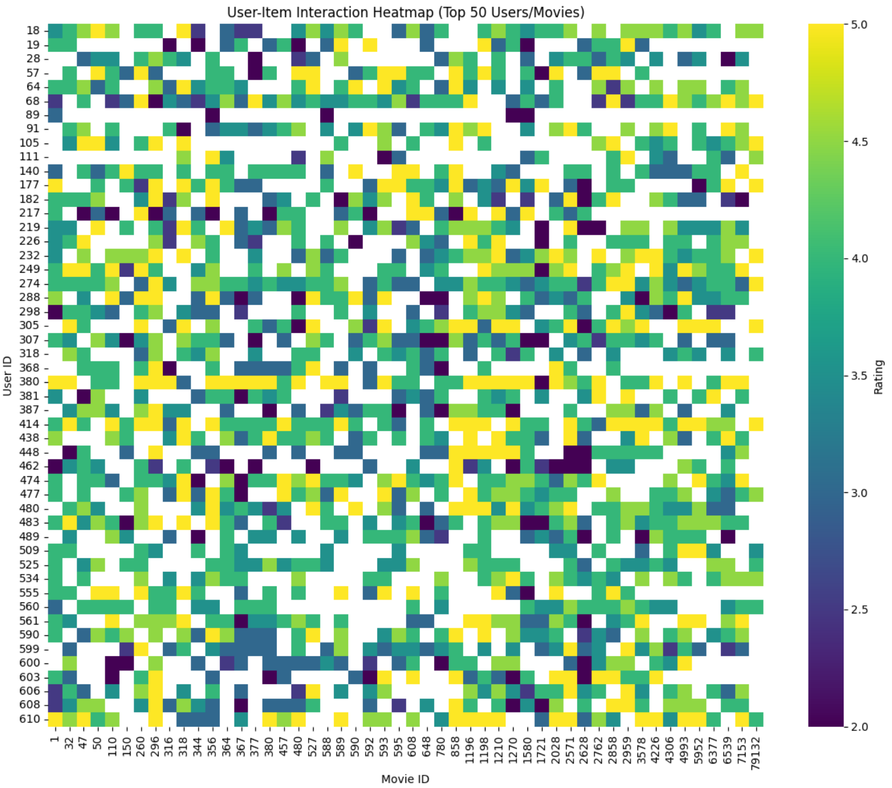

Figure 2: User-Item Interaction Heatmap (Top 50 Subset)

Figure 3 illustrates the Precision@K curve. As expected, precision decreases as K increases, but SVD maintains better stability.

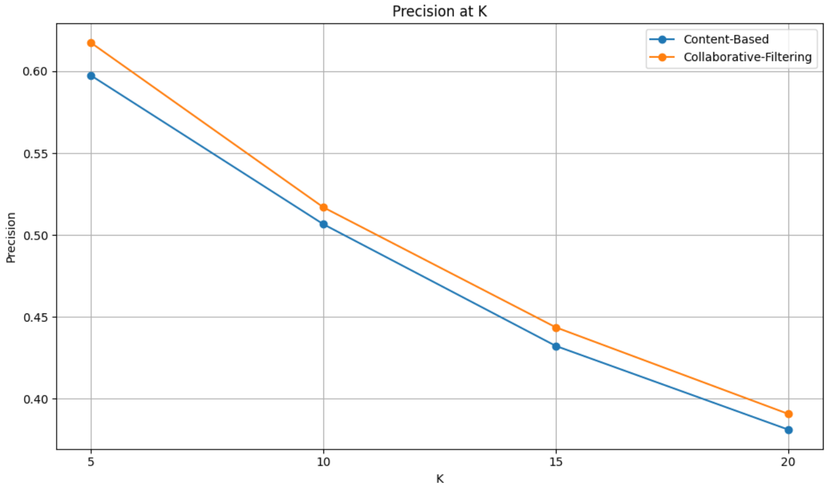

Figure 3: Precision at K (Ranking Quality)

5 Discussion

Our method comparison reveals that **Collaborative Filtering** slightly outperforms the Content-Based approach in both RMSE and top-N ranking metrics. The mean-centering step was critical; without it, the SVD model biased predictions towards zero (missing values), resulting in high error.

The Content-Based model performed surprisingly well given it only used Genres. However, it is limited by its inability to recommend items outside a user's known genre preferences (the "serendipity" problem). SVD, by learning latent factors, can capture cross-genre correlations.

6 Conclusion

We successfully designed and evaluated two recommender systems. The SVD-based collaborative filtering approach demonstrated robust performance. Future work should focus on **Hybrid Systems**, combining the cold-start resilience of content-based models with the accuracy of collaborative filtering, and incorporating temporal dynamics (e.g., handling changing user tastes over time).

References

[1] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. *Proceedings of CSCW '94*.

[2] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of WWW '01*.

[3] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8).