Book Recommendation using Collaborative Filtering (Book-Crossing)

An item-based collaborative-filtering recommender on the Book-Crossing benchmark — 'readers who liked X also liked Y', with every suggestion citing its evidence book so the recommendation is explainable.

---

1. Introduction / Background

'Readers who liked X also liked Y' is the oldest trick in recommendation — and still the backbone of how bookstores, libraries and reading apps suggest your next book. The idea is collaborative filtering: find items similar to the ones you liked, where similarity is learned from the rating behaviour of thousands of readers, not from genre tags.

This project builds item-based collaborative filtering properly on the Book-Crossing dataset (1.15 million ratings): a user–item rating matrix, item–item cosine similarity, score aggregation, top-N ranking, and evaluation with Precision@5 and Recall@10 against user-based CF and popularity baselines.

2. Problem Statement

Choice overload paralyzes readers — 271,379 books and no idea what to read next. Black-box recommenders give scores without reasons, which users distrust. Item-based CF is both effective and explainable: every suggestion cites its evidence book. Students need a project that teaches ranking evaluation (not classification accuracy) and the honest limits of collaborative filtering — cold start, sparsity, popularity bias.

3. Objectives

- Build item-based collaborative filtering with item–item cosine similarity on Book-Crossing.
- Evaluate with Precision@5 and Recall@10 on held-out ratings, against user-based CF and popularity baselines.
- Make every recommendation explainable with its evidence book.
- Build an interactive demo: pick liked books, get ranked suggestions.

4. Proposed Methodology

Book-Crossing ratings (1–10 scale) are cleaned and assembled into a user–item matrix. Item–item cosine similarity is computed across rating vectors. For a user's liked books, candidate scores are aggregated as Σ similarity × rating and ranked top-N, each citing its evidence book. A popularity baseline and user-based CF variant are built for comparison on the same held-out ratings, and all three are scored with Precision@5 and Recall@10.

5. System Architecture / Working

- Input: user's liked books.
- Similarity: precomputed item–item cosine matrix from the rating matrix.

- Scoring: Σ similarity × rating per candidate → top-N ranking.
- Demo: book picks → instant ranked, evidence-cited recommendations.

6. Hardware / Software Requirements

- Python 3, Pandas, NumPy, scikit-learn, Matplotlib, Jupyter Notebook.
- Any laptop — similarity computation is the heaviest step and runs in minutes.
- The web demo runs offline in any modern browser after download.

7. Expected Outcomes

- An item-based CF model reaching the design targets of ≈ 0.31 Precision@5 and ≈ 0.42 Recall@10, beating the baselines, reported honestly after the run.
- An interactive recommender demo with explainable suggestions.
- A complete report, PPT and viva Q&A; covering CF theory and ranking metrics.

8. Applications

- Library and bookstore recommendation demonstrations.
- Teaching recommender systems: ranking metrics, baselines, cold start.
- A template for recommenders in other domains (movies, courses).

9. Future Scope

- Hybrid content + collaborative models to ease cold start.
- Matrix factorization (SVD) comparison.
- Diversification and serendipity in ranking.

10. Conclusion

The project shows recommendation done right: a canonical algorithm, ranking-native evaluation, baselines that keep it honest, and explanations for every suggestion — plus a clear-eyed account of what collaborative filtering cannot do.