PROJECT ABSTRACT

Recommendation System using Collaborative Filtering

SVD matrix-factorization recommender trained on MovieLens 100K (943 users, 1,682 movies, 100,000 ratings), delivered as a built-to-order student project with training notebook, evaluation reports, interactive demo app, report, PPT and viva kit.

01. Title

Recommendation System using Collaborative Filtering

02. Introduction / Background

Recommender systems decide what billions of people watch, buy and read. Collaborative filtering (CF) is the foundational technique: it ignores item content entirely and learns purely from the user-item rating matrix, on the intuition that users who agreed in the past will agree in the future. Matrix factorization — the method that won the Netflix Prize — compresses the sparse rating matrix into low-dimensional user and item latent factors whose dot product predicts unseen ratings. This project implements SVD-based collaborative filtering on the MovieLens 100K benchmark and wraps it in a demo that explains every recommendation.

03. Problem Statement

Students learning recommender systems usually meet CF only as library calls and toy matrices; the leap from a 5x5 example to a real 943x1,682 sparse matrix — with cold start, sparsity and popularity bias — is never demonstrated. The project closes that gap with a complete, honest build: real SVD training on MovieLens 100K with a logged optimization curve, leave-one-out evaluation with ranking metrics, and a demo that shows latent taste factors, nearest users and 'because you liked' explanations for every recommendation.

04. Objectives

- Train an SVD model (50 latent factors, SGD, 30 epochs) on MovieLens 100K using the Surprise library.
- Reach a design target of ~0.86-0.90 test RMSE, with the notebook logging the per-epoch training curve.
- Evaluate with ranking metrics (Precision@5, Recall@10, nDCG@10) under a leave-one-out protocol, all computed by the included evaluation notebook.

- Build an interactive demo: profile picker, predicted ratings with explanations, latent factor viewer, nearest-user panel and evaluation dashboard.
- Deliver the complete student kit: source code, notebooks, report, PPT and viva Q&A.

05. Proposed Methodology

The pipeline has three stages. (1) Data: MovieLens 100K ratings (1-5 stars, every user has >20 ratings) are loaded and split into train/test; user and item means are computed for bias terms. (2) Model: SVD factorizes the 943x1,682 matrix into 50-dimensional user and item factors plus bias terms, optimized with SGD (learning rate 0.005, regularization 0.02) for 30 epochs; RMSE is logged every epoch. (3) Evaluation and serving: the test split yields RMSE, while a leave-one-out protocol (hide each user's latest rating, predict it back, rank all items) yields Precision@K, Recall@K and nDCG@K. The demo loads the trained factors and serves top-N lists with neighbor-based explanations.

06. System Architecture / Working

The user picks a profile and sees their rated movies. On request, the demo computes predicted ratings as user-factor dot item-factor plus biases, ranks unseen movies, and renders the top 6 as cards with predicted stars and a 'because you liked X' line drawn from the highest-overlap rated movie. The 'Inside the model' view visualizes the user's latent factor bars, the three nearest users by cosine similarity, and the logged training-RMSE curve. The evaluation view presents the notebook's offline metrics with a plain-language meaning for each.

07. Hardware / Software Requirements

- Hardware: any 64-bit PC with 4 GB+ RAM; no GPU needed (SVD on 100K ratings trains in minutes on CPU).
- Python 3.10, scikit-surprise (SVD), pandas, NumPy, scikit-learn.
- Matplotlib (RMSE curves, metric charts).
- Flask demo web app (recommend, model-inspection and evaluation views).
- MovieLens 100K dataset (943 users, 1,682 movies, 100,000 ratings).

08. Expected Outcomes

- A trained SVD recommender with test RMSE logged by the notebook (design target ~0.86-0.90; final numbers from the build).
- Offline ranking metrics (Precision@5, Recall@10, nDCG@10, coverage) from the leave-one-out protocol.
- An interactive demo exposing latent factors, nearest users and per-recommendation explanations.

- A report with an honest limitations section (cold start, sparsity, popularity bias) plus PPT and viva Q&A.

09. Applications

- Teaching reference for matrix factorization, latent factors and ranking metrics.
- Baseline recommender for college media-library, canteen-menu or course-elective prototypes.
- Starting point for hybrid extensions (this project's natural follow-up).
- Evaluation harness reusable for comparing CF algorithms (kNN, NMF, autoencoders).

10. Future Scope

- Hybrid extension blending CF scores with content-based similarity.
- Implicit-feedback model (ALS) using watch/click histories instead of stars.
- Serendipity re-ranking to counter popularity bias.
- Online A/B evaluation harness measuring real click-through beyond offline metrics.