# DSCI-D 351 Project 4: Collaborative Filtering on MovieLens 100K

Pete Gerdsen • December 13, 2025

## 1) Data representation and split strategy

**Dataset:** MovieLens 100K (943 users, 1682 movies, 100,000 ratings, scale 1-5). **Representation:** sparse dictionaries: user_ratings[u][m]=r and movie_ratings[m][u]=r.

**Split (hidden ratings):** randomly select 20% of users as test users (189/943). For each test user, hide 20% of their ratings (3780 hidden targets total). Training includes all ratings from non-test users plus the remaining 80% visible ratings for test users.

**Safeguards:** ensure each test user retains at least 5 visible ratings; if a target movie has no usable neighbors in training (or denominator=0), fall back to user mean (or global mean). Predictions are clipped to [1,5].

## 2) Similarity, KNN prediction, and hyperparameters

**User-user CF (required):** for each hidden (u,m), consider users v who rated m in training. Compute similarity s(u,v) on co-rated movies, select top-k neighbors, and predict with mean-centering:

*r_hat(u,m)=mu_u + (sum_v s(u,v)*(r(v,m)-mu_v)) / (sum_v |s(u,v)|).*

**Similarities:** cosine similarity and Pearson correlation; Pearson neighbors restricted to non-negative similarity (>=0). **Normalization impact:** mean-centering improved accuracy vs a non-centered weighted average (cosine k=40 RMSE 0.9513 vs 1.0084; Pearson>=0 k=40 RMSE 0.9499 vs 0.9999).

**Hyperparameter sweep:** k in {5,10,20,40}. Larger k reduced RMSE but increases compute; similarities are cached during evaluation.

## 3) Results, interpretation, limitations, and bonus

### User-user results (hidden-rating evaluation)

| Model | k | MSE | RMSE |
|---|---|---|---|
| User-User (Cosine) | 5 | 1.0428 | 1.0212 |
| User-User (Cosine) | 10 | 0.9725 | 0.9861 |
| User-User (Cosine) | 20 | 0.9221 | 0.9603 |
| User-User (Cosine) | 40 | 0.9050 | 0.9513 |
| User-User (Pearson>=0) | 5 | 1.0486 | 1.0240 |
| User-User (Pearson>=0) | 10 | 0.9678 | 0.9838 |
| User-User (Pearson>=0) | 20 | 0.9244 | 0.9614 |
| User-User (Pearson>=0) | 40 | 0.9023 | 0.9499 |

**Best user-user:** Pearson>=0, k=40 (MSE 0.9023, RMSE 0.9499). **Accurate:** user 90, movie 515 (Das Boot (1981)) true 5.0, pred 5.000. **Far off:** user 239, movie 318 (Schindler's List (1993)) true 1.0, pred 4.903. Large errors typically occur when a user is an outlier relative to neighbors for a broadly liked title.

## Bonus) Item-item collaborative filtering

Item-item CF uses adjusted cosine similarity between movies (each user's ratings centered by their mean), then predicts a user's rating from the k most similar movies they rated (similarity-weighted average).

| Model (Bonus) | k | MSE | RMSE |
|---|---|---|---|
| Item-Item (Adj. Cosine) | 5 | 1.0485 | 1.0239 |
| Item-Item (Adj. Cosine) | 10 | 0.9522 | 0.9758 |
| Item-Item (Adj. Cosine) | 20 | 0.9049 | 0.9513 |
| Item-Item (Adj. Cosine) | 40 | 0.8914 | 0.9441 |

**Best overall:** item-item k=40 (MSE 0.8914, RMSE 0.9441), slightly better than best user-user (RMSE 0.9499). **Accurate:** user 644, movie 276 (Leaving Las Vegas (1995)) true 4.0, pred 4.000. **Far off:** user 739, movie 465 (The Jungle Book (1994)) true 1.0, pred 4.940.

## Strengths, limitations, and improvement

**Strengths:** interpretable neighborhood models; good performance on frequently-rated movies; controllable bias-variance via k. **Limitations:** sparsity for low-support movies, cold-start users/movies, and popularity bias (overpredicting popular items for outlier users). **Improvement:** add baseline biases (global+user+movie) and predict residuals with kNN, or use matrix factorization for better generalization. **When item-based is preferable:** very large user base with a relatively stable item catalog (item similarities can be cached).

## Reference

Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS), 5(4), Article 19.