PMAI LAB • CASE STUDY 02

Streaming Recommendation System: Vector Similarity and Latent Structure

Inner Products • Cosine Similarity • SVD • Python

Application field
Recommender systems and digital platforms

Mathematical focus
Rating vectors, cosine similarity, missing-value estimation, low-rank SVD

Computational tool
Python with NumPy and Matplotlib

Instructor
Dr. Hira Benish

Use similarity for a local prediction and SVD
to reveal the platform’s global preference structure.

1. Scenario and Rating Data

A streaming platform wants to estimate a missing rating and compress user-preference information

Six users rate six content categories on a scale from 1 to 5. User 1 has not rated Adventure:

	Action	Sci-Fi	Adventure	Documentary	History	Nature
U_1	5	4	?	1	1	1
U_2	4	5	4	1	1	1
U_3	5	4	5	1	1	1
U_4	1	1	1	5	4	4
U_5	1	1	1	4	5	4
U_6	1	1	1	5	4	5

Investigation questions

1. Which users have rating patterns most similar to User 1?
2. What rating should be predicted for User 1 on Adventure?
3. Does the completed rating matrix contain low-rank structure?
4. How much information is retained by a rank-two SVD approximation?
5. What are the benefits and risks of using this model for recommendations?

Learning outcomes

Students will represent users as vectors, calculate cosine similarity, make a similarity-weighted prediction, perform SVD, choose a rank using retained energy, quantify reconstruction error and storage, implement the workflow in Python, and discuss limitations.

Two mathematical lenses
Cosine similarity gives a local comparison between users. SVD gives a global decomposition of the complete user-category matrix and reveals latent preference groups.

Missing-data rule

The unknown entry is excluded when similarities are calculated. It is not replaced by zero, because zero would incorrectly represent a strong negative preference rather than an unobserved rating.

2. Vector Similarity and Rating Prediction

Compare User 1 with users who have already rated Adventure

Use only the categories jointly rated by Users 1, 2, and 3:

$$\boldsymbol{u}_1 = \begin{pmatrix} 5 \\ 4 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad \boldsymbol{u}_2 = \begin{pmatrix} 4 \\ 5 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad \boldsymbol{u}_3 = \begin{pmatrix} 5 \\ 4 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

Cosine similarity is

$$\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u}^\top \boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|}.$$

User 1 compared with User 2

$$\boldsymbol{u}_1^\top \boldsymbol{u}_2 = 43, \qquad \|\boldsymbol{u}_1\| = \|\boldsymbol{u}_2\| = \sqrt{44},$$

so

$$\text{sim}(U_1, U_2) = \frac{43}{44} \approx 0.9773.$$

User 1 compared with User 3

The two observed vectors are identical:

$$\text{sim}(U_1, U_3) = 1.$$

Similarity-weighted prediction

Users 2 and 3 rated Adventure as 4 and 5. Predict

$$\widehat{r}_{1,3} = \frac{0.9773(4) + 1(5)}{0.9773 + 1} \approx \boxed{4.5057}.$$

Recommendation decision
Predict approximately 4.51 for User 1 on Adventure. Because the platform uses a five-point scale, it may display the expected rating or round it according to product policy.

Interpretation caution
High cosine similarity means similar rating direction, not identical behaviour. The prediction uses only two neighbours and may be unstable when few co-rated items are available.

3. SVD and Latent Preference Structure

Complete the matrix and retain the dominant information layers

After inserting the predicted rating, the completed matrix is

$$\mathbf{R} = \begin{bmatrix} 5 & 4 & 4.5057 & 1 & 1 & 1 \\ 4 & 5 & 4 & 1 & 1 & 1 \\ 5 & 4 & 5 & 1 & 1 & 1 \\ 1 & 1 & 1 & 5 & 4 & 4 \\ 1 & 1 & 1 & 4 & 5 & 4 \\ 1 & 1 & 1 & 5 & 4 & 5 \end{bmatrix}.$$

Its SVD is

$$\mathbf{R} = \mathbf{U}\Sigma\mathbf{V}^{\top}.$$

The singular values are approximately

$$16.4310, \quad 10.4363, \quad 1.2525, \quad 1.2073, \quad 0.5586, \quad 0.2627.$$

Choose the rank

The first two components retain

$$\frac{\sigma_1^2 + \sigma_2^2}{\sum_{i=1}^{6} \sigma_i^2} \approx \boxed{99.11\%}.$$

Thus rank 2 captures the dominant preference structure.

Rank-two reconstruction

$$\mathbf{R}_2 \approx \begin{bmatrix} 4.69 & 4.32 & 4.52 & 1.00 & 1.00 & 1.00 \\ 4.48 & 4.14 & 4.32 & 1.00 & 1.01 & 1.00 \\ 4.86 & 4.48 & 4.69 & 0.99 & 1.00 & 1.00 \\ 1.00 & 1.00 & 1.00 & 4.57 & 4.22 & 4.25 \\ 1.00 & 1.01 & 1.00 & 4.54 & 4.19 & 4.22 \\ 0.99 & 1.01 & 1.00 & 4.91 & 4.54 & 4.57 \end{bmatrix}.$$

The Frobenius reconstruction error is

$$\|\mathbf{R} - \mathbf{R}_2\|_F \approx 1.8459.$$

Storage comparison

The original stores $6 \times 6 = 36$ values. Rank 2 stores

$$2(6 + 6 + 1) = 26$$

values, a reduction of approximately 27.78%.

> **Latent interpretation**
> One dominant direction captures overall rating level; another separates action-oriented content from documentary and knowledge-oriented content.

4. Recommendation Decision and Model Evaluation

Combine local similarity with global low-rank evidence

Local result

User 1 is highly similar to Users 2 and 3 on co-rated categories:

$$\text{sim}(U_1, U_2) \approx 0.9773, \qquad \text{sim}(U_1, U_3) = 1.$$

The predicted Adventure rating is

$$\widehat{r}_{1,3} \approx 4.5057.$$

Global result

The rank-two SVD retains approximately 99.11% of matrix energy. Its reconstructed User 1 Adventure entry is approximately 4.52, which is close to the neighbour-based estimate.

> **Platform recommendation**
> Adventure is a strong recommendation candidate for User 1. The agreement between similarity and low-rank reconstruction provides additional confidence for this small example.

What the model learns

- Users 1–3 form an action, science-fiction, and adventure preference group.
- Users 4–6 form a documentary, history, and nature preference group.
- Rank-two structure summarizes the two main preference patterns.

What the model does not learn

- why a user gave a rating;
- whether preferences changed over time;
- whether a category is unavailable in a user's region;
- whether the recommendation promotes useful discovery or creates a narrow filter bubble.

> **Responsible limitation**
> Real recommender systems must address sparse data, cold-start users, popularity bias, privacy, fairness, feedback loops, and diversity. High numerical similarity is not sufficient evidence of user satisfaction.

Evaluation beyond reconstruction

In practice, hold out known ratings and measure prediction error, ranking quality, coverage, novelty, diversity, and user outcomes.

5. Python Laboratory

Estimate the missing rating, perform SVD, select rank, and visualize the reconstruction

```python
import numpy as np
import matplotlib.pyplot as plt

R = np.array([
    [5., 4., np.nan, 1., 1., 1.],
    [4., 5., 4.,     1., 1., 1.],
    [5., 4., 5.,     1., 1., 1.],
    [1., 1., 1.,     5., 4., 4.],
    [1., 1., 1.,     4., 5., 4.],
    [1., 1., 1.,     5., 4., 5.]
])

def cosine(a, b):
    return (a @ b) / (np.linalg.norm(a) * np.linalg.norm(b))

# Exclude User 1's missing Adventure rating
observed = ~np.isnan(R[0])
u1 = R[0, observed]
u2 = R[1, observed]
u3 = R[2, observed]

sim12 = cosine(u1, u2)
sim13 = cosine(u1, u3)
prediction = (
    sim12 * R[1, 2] + sim13 * R[2, 2]
) / (sim12 + sim13)

R_complete = R.copy()
R_complete[0, 2] = prediction

# Global low-rank model
U, s, Vt = np.linalg.svd(R_complete, full_matrices=False)
energy = np.cumsum(s**2) / np.sum(s**2)
k = 2
R_k = (U[:, :k] * s[:k]) @ Vt[:k, :]
error = np.linalg.norm(R_complete - R_k, ord="fro")
stored_original = R_complete.size
stored_rank_k = k * (R.shape[0] + R.shape[1] + 1)

print("Similarity U1-U2:", round(float(sim12), 4))
print("Similarity U1-U3:", round(float(sim13), 4))
print("Predicted Adventure rating:", round(float(prediction), 4))
print("Singular values:", np.round(s, 4))
print("Cumulative energy:", np.round(energy, 4))
print("Rank-2 error:", round(float(error), 4))
print("Stored values:", stored_original, "->", stored_rank_k)
print("Rank-2 U1 Adventure:", round(float(R_k[0, 2]), 4))

fig, axes = plt.subplots(1, 2, figsize=(9, 3.6))
im0 = axes[0].imshow(R_complete, cmap="YlGnBu", vmin=1, vmax=5)
axes[0].set_title("Completed ratings")
im1 = axes[1].imshow(R_k, cmap="YlGnBu", vmin=1, vmax=5)
axes[1].set_title("Rank-2 reconstruction")
for ax in axes:
    ax.set_xlabel("Content category")
    ax.set_ylabel("User")
fig.colorbar(im1, ax=axes, fraction=0.025)
plt.tight_layout()
plt.show()
```

Selected output

```
Similarity U1-U2: 0.9773    Similarity U1-U3: 1.0
Predicted Adventure rating: 4.5057
Singular values: [16.431 10.4363 1.2525 1.2073 0.5586 0.2627]
Cumulative energy: [0.7062 0.9911 0.9952 0.999 0.9998 1.]
Rank-2 error: 1.8459    Stored values: 36 -> 26
Rank-2 U1 Adventure: 4.5217
```

6. Student Investigation Tasks

Extend the recommendation analysis and test different modelling choices

Part A: similarity

1. Recalculate sim(U_1, U_2) and sim(U_1, U_3) by hand.
2. Compute cosine similarity between Users 4 and 6 using all six categories.
3. Explain why the missing rating must be excluded rather than replaced by zero.

Part B: prediction sensitivity

4. Suppose User 2 changes the Adventure rating from 4 to 3. Recalculate User 1's predicted rating using the same similarities.
5. Compare the new result with the original prediction 4.5057.

Part C: rank selection

6. Confirm that rank 2 retains at least 99% of squared singular-value energy.
7. Find the smallest rank retaining at least 99.5% energy.
8. Compare the storage count for ranks 2 and 3 with the original 36 entries.

Part D: responsible design

9. Identify two ways popularity bias could affect recommendations.
10. Suggest one method for increasing recommendation diversity.
11. Explain why reconstruction error alone is not a complete evaluation metric.

Submission expectation
Include similarity calculations, the predicted missing rating, singular values, energy calculations, a rank decision, Python output, one heatmap, and a short responsible-recommendation discussion.

Advanced extension
Create a train-test experiment: hide several known ratings, predict them, and report RMSE. Compare neighbour-based prediction with rank-k SVD reconstruction.

7. Answers, Takeaways, and Extension

Brief answers

Task	Answer
1	sim$(U_1, U_2) = 43/44 \approx 0.9773$ and sim$(U_1, U_3) = 1$.
2	sim$(U_4, U_6) = 64/\sqrt{60 \cdot 69} \approx 0.9947$.
3	Zero would be interpreted as a numerical rating and would distort angles and norms.
4	New prediction ≈ 4.0115.
5	The prediction decreases by approximately 0.4942 rating points.
6	Rank 2 retains approximately 99.11% energy.
7	Rank 3 is the smallest rank exceeding 99.5% energy.
8	Rank 2 stores 26 values; rank 3 stores 39; original matrix stores 36.

Why rank three is not automatically better

Rank 3 slightly increases retained energy, but its factorized representation stores more values than the original tiny matrix. Rank choice must balance accuracy, storage, speed, interpretability, and application quality.

Key takeaways

- Ratings are vectors, and inner products support similarity measures.
- Missing values must be handled differently from genuine low ratings.
- Neighbour-based prediction is transparent but sensitive to the available neighbours.
- SVD reveals latent groups and produces controlled low-rank approximations.
- Energy retention and reconstruction error do not measure fairness, diversity, or satisfaction.

Final conclusion
Similarity predicts a rating of approximately 4.51 for User 1 on Adventure. Rank-two SVD retains about 99.11% of matrix energy and reveals two major content-preference groups.

Key terms: rating vector • inner product • cosine similarity • missing value • weighted prediction • SVD • singular value • latent factor • low-rank approximation • reconstruction error