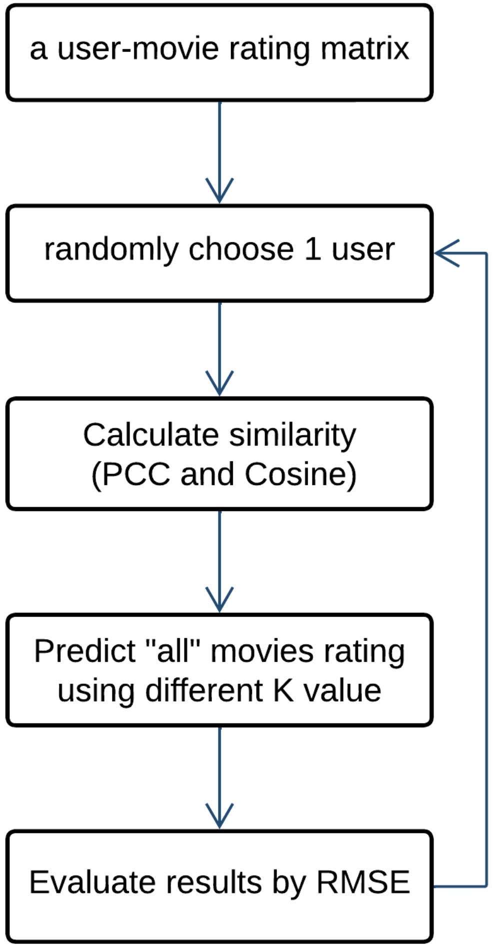
Practical Data Science with Python
(COSC2670)

- Assignment 3

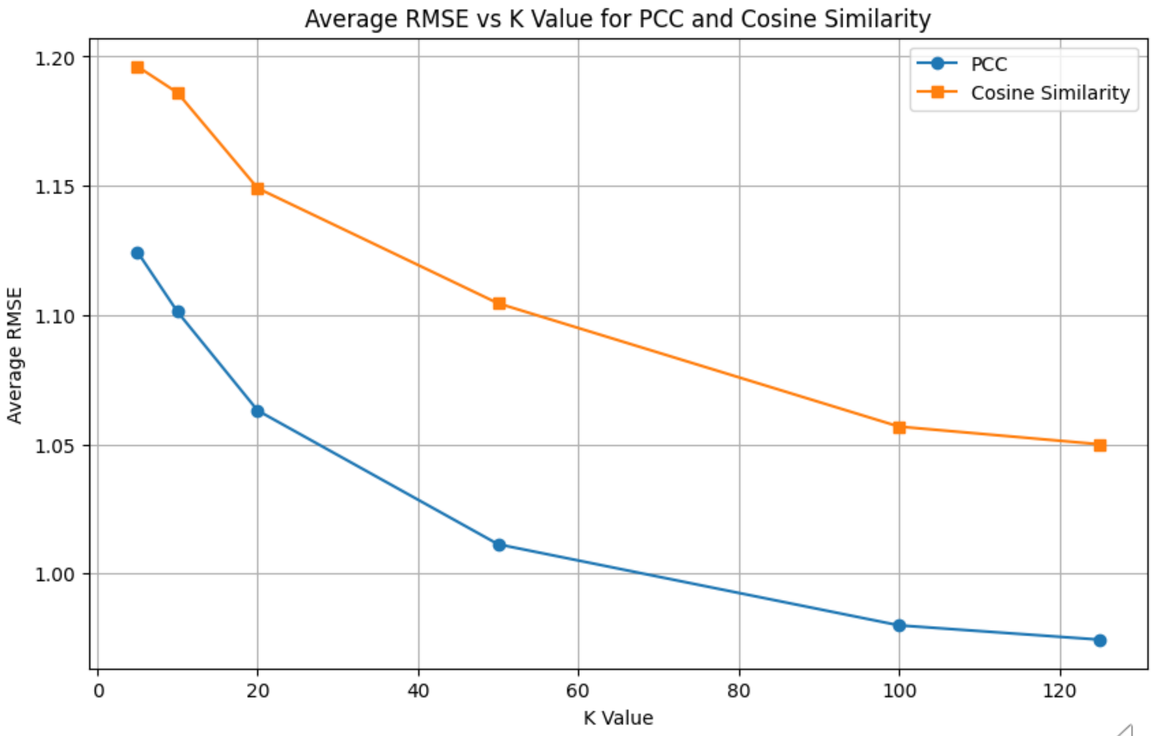
S4068959
Jung-De Chiou

Task1: kNN-based Collaborative Filtering

Original Dataset:
6040 Users, 3883 Movies, 1000209 Ratings



- PCC has better performance
- **K = 125 has lower RMSE**



▶ Run 100 times and plot the average for each K value

※ Detailed calculation can be found in ipynb file

Task2: Matrix Factorization-based Recommendation

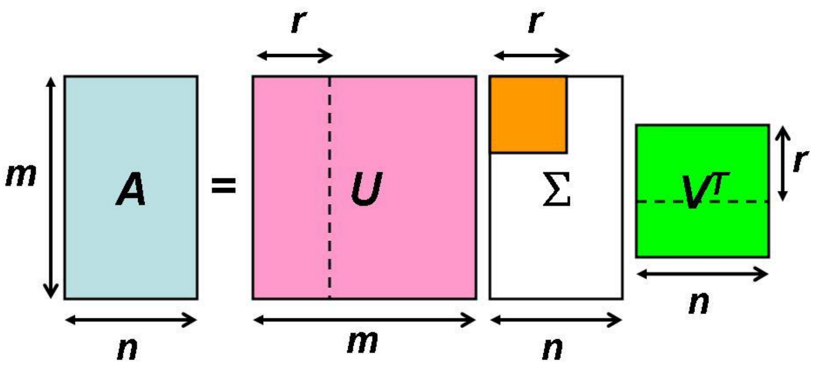
SVD（Singular Value Decomposition）:

Singular Value Decomposition (SVD) is a mathematical technique used in linear algebra to decompose a matrix into three other matrices. Specifically, it factorizes a given matrix A into three matrices: U, Σ, and V^T

U = It represents the relationships between the rows (e.g., users).

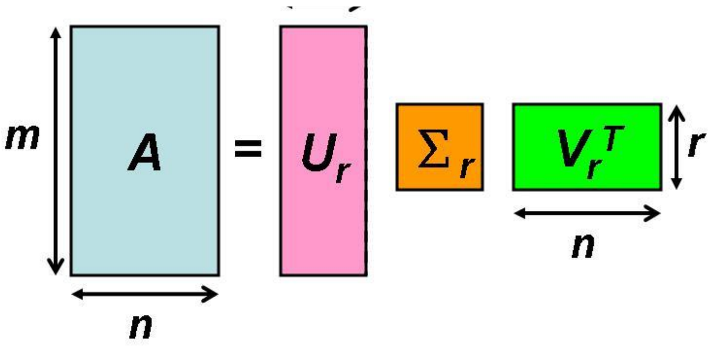
V^T = It represents the relationships between the columns (e.g., items).

Σ = These values help identify the dominant latent features in the data.



Truncated SVD:

Truncated SVD is a variant of SVD that is used specifically to reduce the dimensionality of the data by keeping only the most significant components. Instead of decomposing a matrix completely into UΣVT, Truncated SVD only retains the top k singular values and their corresponding vectors. This is particularly useful in handling large, sparse matrices, which are common in applications like natural language processing and recommendation systems.

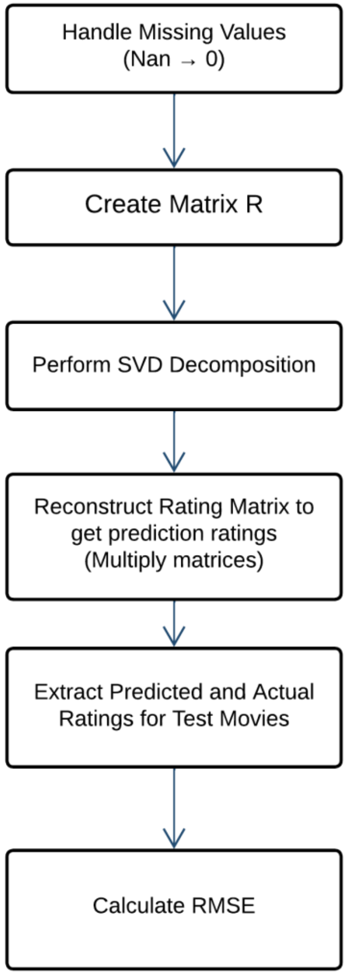


Phadnis, Neelam & Gadge, Jayant. (2014). Framework for Document Retrieval using Latent Semantic Indexing. International Journal of Computer Applications. 94. 37-41. 10.5120/16414-6065.

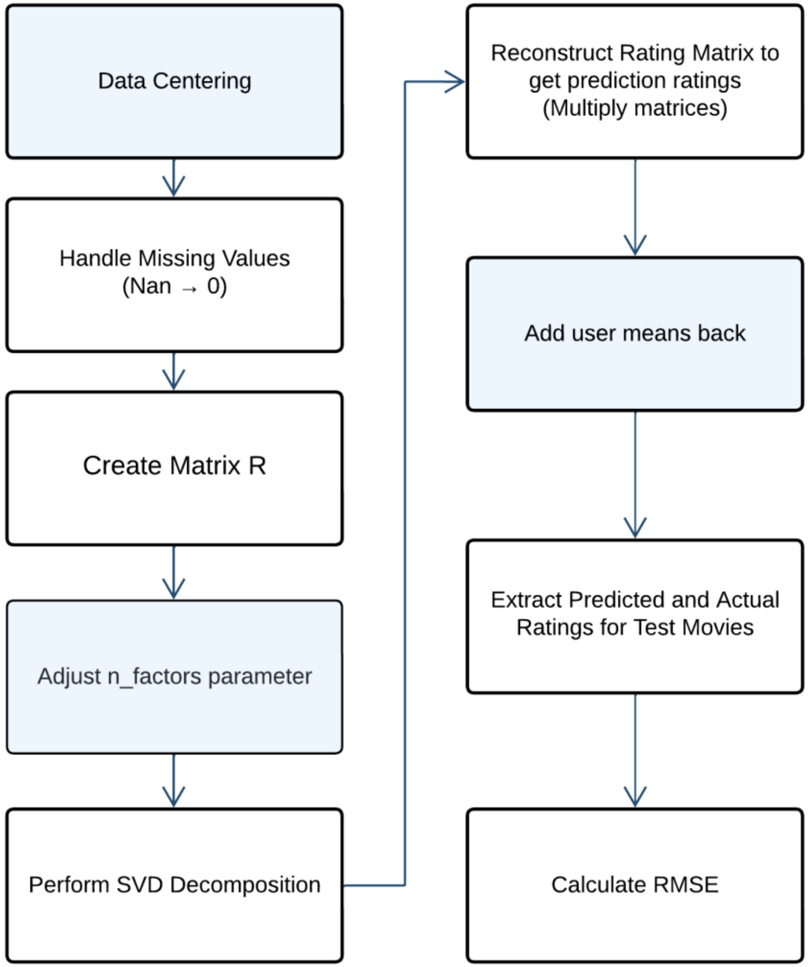
Task2

- Randomly choose 5 items and predict **all** users' ratings on these movies.

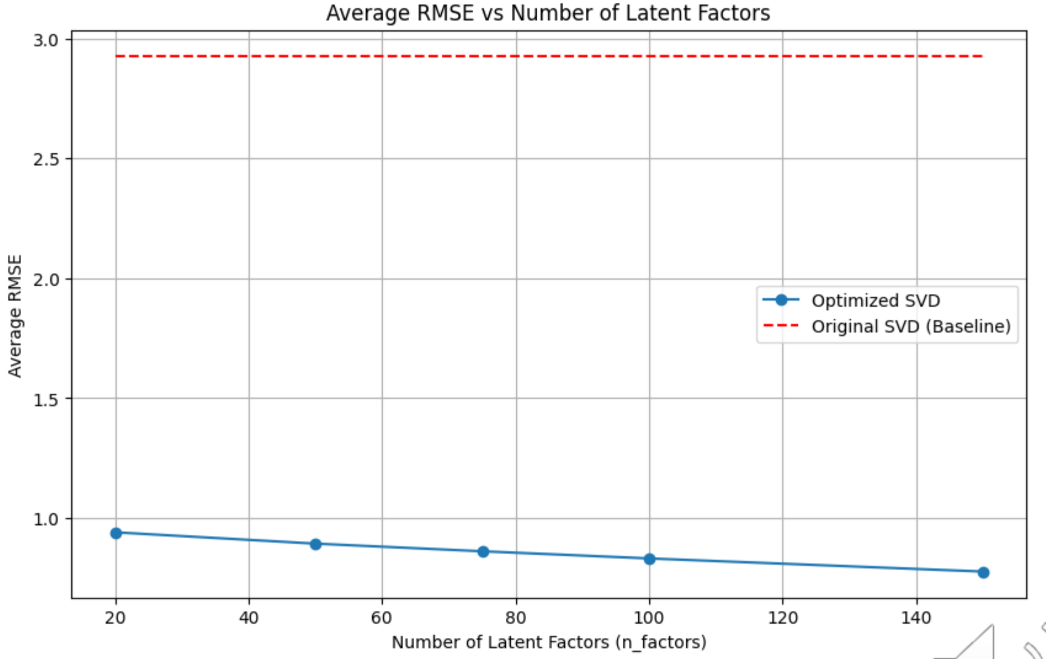
Truncated SVD



Modified Truncated SVD



- **Data centering** and **Adjusting n_factors** parameters
- Modified SVD performance better than the original one
- **N_factors = 150** has lower RSME
- Restrict n_factors to 150

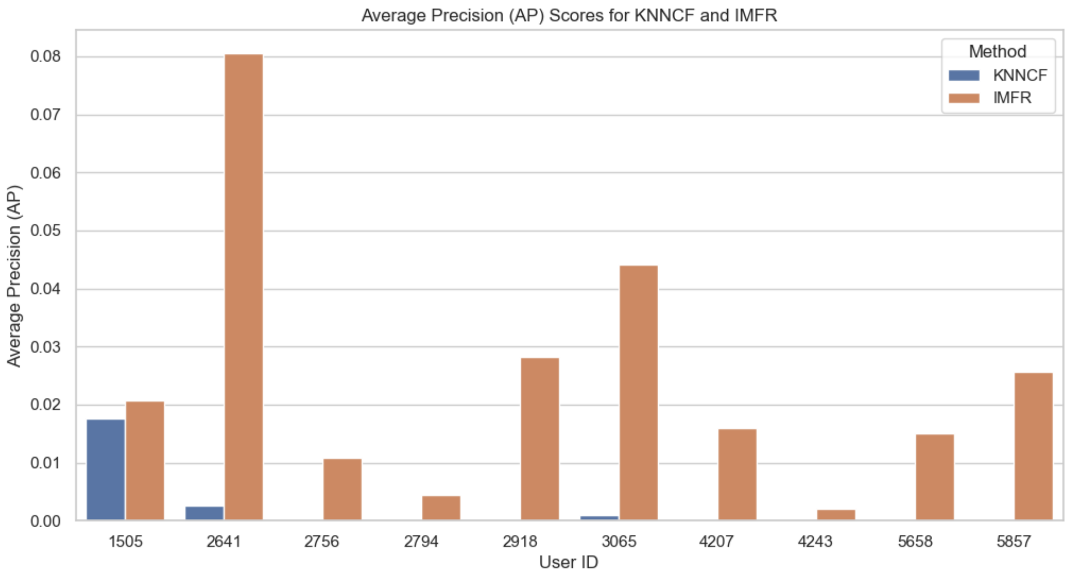


- Run 100 times and plot the average for each n_factors value

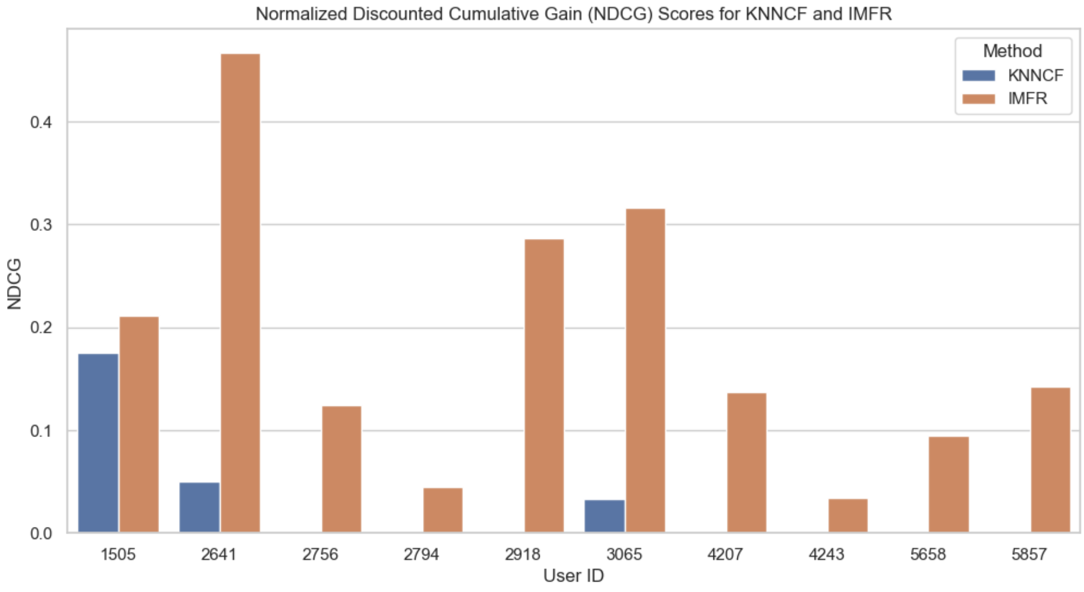
Task3: Ranking-based Evaluation and Comparison

Randomly choose 10 users and recommend Top-20 movies to each of them

AP (Average Precision)



NDCG (Normalized Discounted Cumulative Gain)



- **IMFR consistently outperforms KNNCF**
- **KNNCF's performance is significantly lower**, with many instances where the AP and NDCG scores are close to zero.

Task3

Limitations of KNNCF

1. **Data Sparsity Issues**:

   KNNCF relies on finding users with similar tastes based on co-rated items. With few overlapping ratings, it's challenging to compute reliable similarities.

2. **Ineffective Similarity Measures**:

   The Pearson correlation coefficient may not be effective when the number of co-rated items is small. Also, Ratings can be influenced by outliers or users with unusual rating behaviors, affecting similarity calculations.

How to Improve

1. **Include Side Information**: Use user demographics, movie genres, and other metadata to enhance the recommendation process.
2. **Address Data Sparsity**: Implement techniques like data imputation or clustering to reduce sparsity.
3. **Significance Weighting**: Adjusting significance weighting (GAMMA parameter) to mitigate the effect of users with few co-rated items.

Why IMFR Delivers Better Performance

1. **Capturing Underlying Preferences:** IMFR uses Singular Value Decomposition (SVD) to uncover latent factors that represent hidden patterns in user preferences and item characteristics.
2. **Reduced Dependence on Co-Rated Items**: IMFR does not rely solely on direct co-rated items between users, making it more robust in sparse datasets.
3. **Better Generalization**: By reducing dimensionality, IMFR can generalize from observed ratings to predict unseen ratings more effectively.
4. **Data Compression:** SVD compresses the user-item matrix, mitigating the impact of missing values.

Reference

1. scikit-learn. (n.d.). *sklearn.decomposition.TruncatedSVD*. Retrieved from https://scikit-learn.org/dev/modules/generated/sklearn.decomposition.TruncatedSVD.html
2. Data Aspirant. (n.d.). *Truncated SVD*. Retrieved from https://dataaspirant.com/truncated-svd/
3. V7 Labs. (n.d.). *Mean Average Precision*. Retrieved from https://www.v7labs.com/blog/mean-average-precision
4. Towards Data Science. (2022, October 15). *What is Average Precision in Object Detection & Localization Algorithms and How to Calculate It*. Retrieved from https://towardsdatascience.com/what-is-average-precision-in-object-detection-localization-algorithms-and-how-to-calculate-it-3f330efe697b
5. Evidently AI. (n.d.). *NDCG Metric*. Retrieved from https://www.evidentlyai.com/ranking-metrics/ndcg-metric