

Assignment 3:

Recommender Systems

Name: Pranit Santosh Hande

Student number: 3971628

User Based Collaborative Filtering

1. Implements User-Based Collaborative Filtering to make movie recommendations for a specific user (User 1) by finding K nearest similar users based on cosine similarity.
2. Predicts movie ratings for User 1 by considering the preferences and ratings of similar users.
3. Displays the predicted ratings and the top 5 movie recommendations for User 1 at different values of K (neighborhood size) and includes the movie titles for clarity

```
For user 1, K=5, Predicted Ratings: [3.22762671 1.93922462 1.47574514 ... 2.76041667 2.
Recommendations for K=5:
1: "Newton Boys, The (1998)" - Predicted Rating: 4.48
2: "Forbidden Planet (1956)" - Predicted Rating: 4.20
3: "Savage Nights (Nuits fauves, Les) (1992)" - Predicted Rating: 3.99
4: "Favor, The (1994)" - Predicted Rating: 3.94
5: "Sliver (1993)" - Predicted Rating: 3.94
-----
```

```
For user 1, K=10, Predicted Ratings: [2.72379695 2.49868296 1.47574514 ... 2.76041667 2
Recommendations for K=10:
1: "Forbidden Planet (1956)" - Predicted Rating: 4.20
2: "Up Close and Personal (1996)" - Predicted Rating: 4.13
3: "Threesome (1994)" - Predicted Rating: 4.13
4: "Promise, The (Versprechen, Das) (1994)" - Predicted Rating: 4.13
5: "Casablanca (1942)" - Predicted Rating: 4.13
-----
```

```
For user 1, K=15, Predicted Ratings: [3.14530285 2.38299363 1.79007801 ... 2.76041667 2
Recommendations for K=15:
1: "Small Wonders (1996)" - Predicted Rating: 4.14
2: "Little Women (1994)" - Predicted Rating: 4.13
3: "It's a Wonderful Life (1946)" - Predicted Rating: 4.13
4: "Cat People (1982)" - Predicted Rating: 4.13
5: "Montana (1998)" - Predicted Rating: 4.13
-----
```

```
For user 1, K=20, Predicted Ratings: [3.01540775 1.9988516 1.79007801 ... 2.76041667 2
Recommendations for K=20:
1: "Small Wonders (1996)" - Predicted Rating: 4.14
2: "Little Women (1994)" - Predicted Rating: 4.13
3: "Cat People (1982)" - Predicted Rating: 4.13
4: "Ready to Rumble (2000)" - Predicted Rating: 4.13
5: "Promise, The (Versprechen, Das) (1994)" - Predicted Rating: 4.13
-----
```

AP	NDCG	RMSE
0.1825	0.0526	1.01173747 914736



Item Based Collaborative Filtering with KNN

- Utilizes both Cosine Similarity and Jaccard Similarity to measure the similarity between items (movies) based on user interactions
- Predicts ratings for a selected movie (Movie ID 4) for all users by considering the top-K (K=10) most similar movies using both similarity metrics
- Calculates the average rating for the chosen movie and similar items, then iterates through users to provide personalized movie ratings based on item similarity.
- RMSE (Root Mean Square Error) values for Item-Based CF using Jaccard and Cosine Similarity are close, indicating that both similarity metrics perform similarly in predicting movie ratings.

	Cosine Similarity	Jaccard Similarity
RMSE	1.4431863969462515	1.4436061620760676

Matrix Factorization (Singular Value Decomposition)

- 1.Sortes user predictions using Matrix Factorization (SVD), making it easier to recommend movies the user is likely to enjoy based on their ratings.
- 2. Combines user data with movie information, presenting what movies the user has already rated and providing recommendations for unrated movies.
- 3. It utilizes SVD-based Matrix Factorization to recommend the highest-rated movies that the user has not seen yet, enhancing the user's movie-watching experience.
- 4. SVD is an algorithm that decomposes a matrix into the best lower rank (i.e. smaller/simpler) approximation of the original matrix . Mathematically, it decomposes A into a two unitary matrices and a diagonal matrix, where is the input data matrix (users's ratings), is the left singular vectors (user "features" matrix), is the diagonal matrix of singular, and is the right singlar vectors and are column orthonomal, and represent different things. represents how much users "like" each feature and represents how relevant each feature is to each movie.**(Refer to the First image)**

$$\mathbf{A}_{m \times n} = \mathbf{U}_{m \times m} \times \Sigma_{m \times n} \times \mathbf{V}_{n \times n}^T$$

$(m < n)$

$$\mathbf{A}_{m \times n} = \mathbf{U}_{m \times m} \times \Sigma_{m \times n} \times \mathbf{V}_{n \times n}^T$$

$(m > n)$

AP	NDCG	RMSE
0.99	0.99	0.87

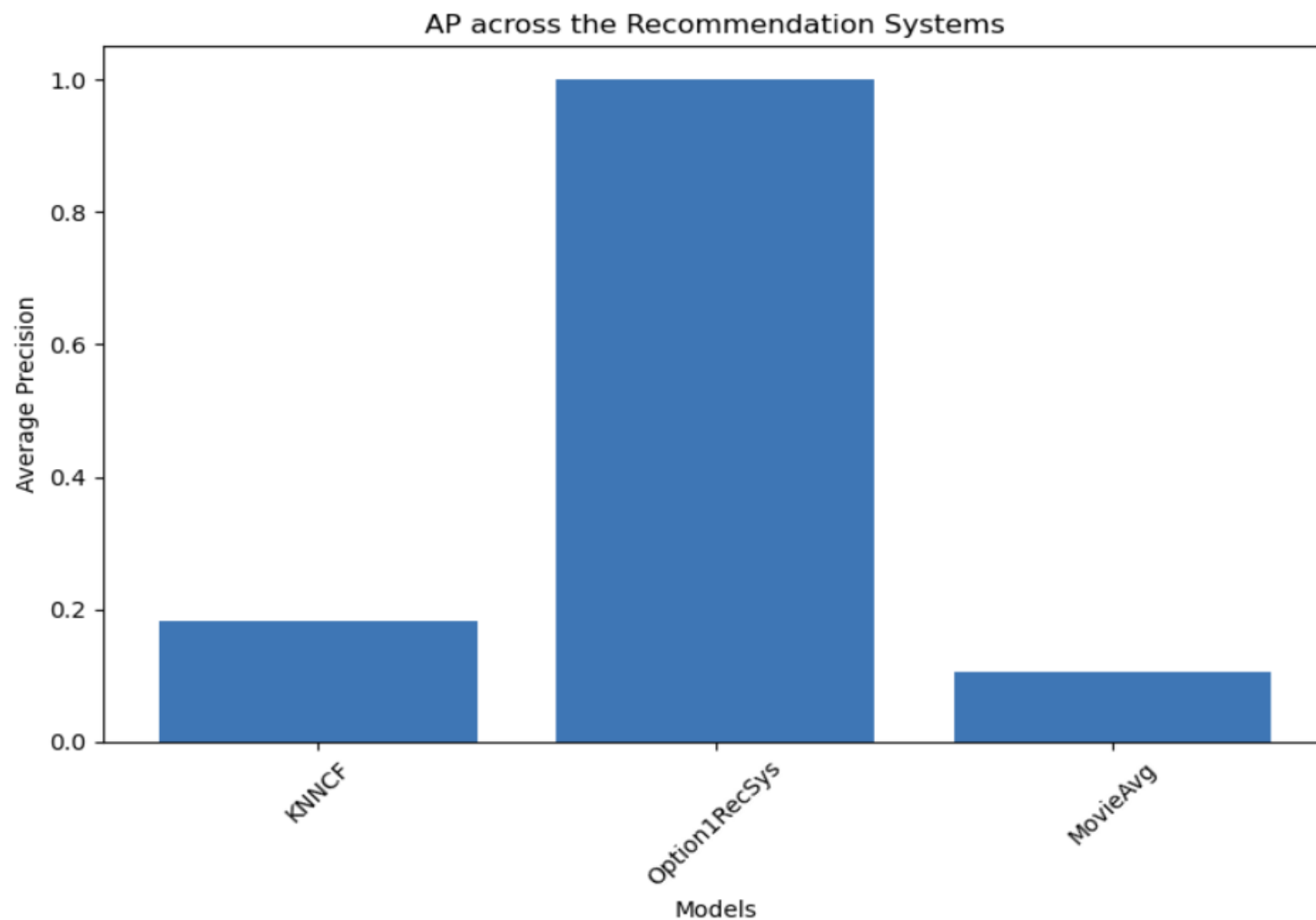
- This information was referenced from :
- Patra, S., & Ganguly, B. (2019). Improvising Singular Value Decomposition by KNN for Use in Movie Recommender Systems. Journal of Operations and Strategic Planning, 2(1), 22-34.
<https://doi.org/10.1177/2516600X19848956>.
- Guan, Xin & Li, Chang-Tsun & Guan, Yu. (2017). Matrix Factorization With Rating Completion: An Enhanced SVD Model for Collaborative Filtering Recommender Systems. IEEE Access. PP. 1-1.
10.1109/ACCESS.2017.2772226.

Movie Average: Recommendation with the highest average ratings

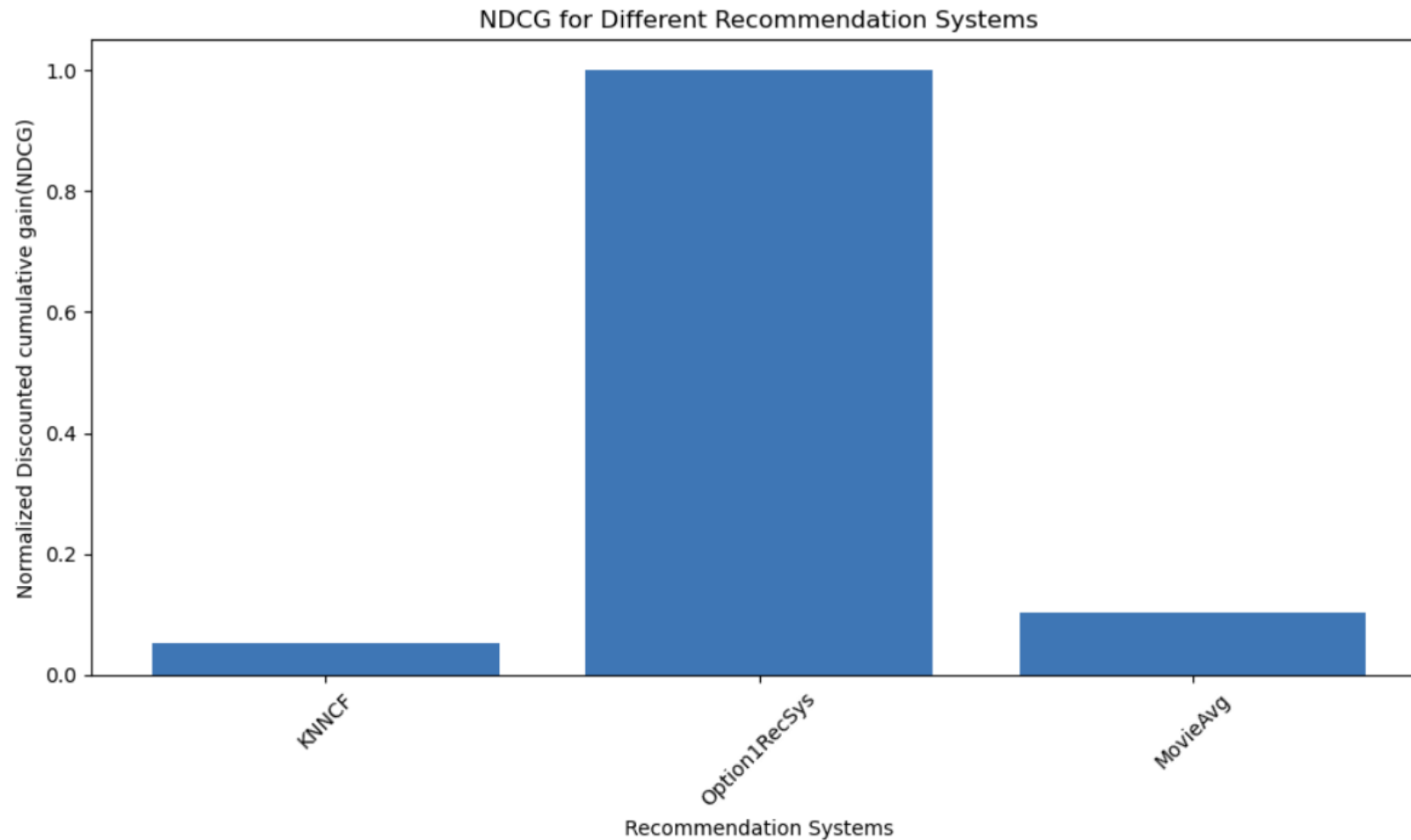
1. Calculates the average rating for each movie, enabling the assessment of movie popularity and quality.
2. It forms a user-movie matrix, revealing user preferences and interactions with different movies.
3. Identifies users who have rated more than 100 movies and recommends the top 30 movies with the highest average ratings for five randomly selected users, enhancing their movie-watching experience based on popular and highly-rated films.

AP	NDCG	RMSE
0.105316	0.105316	0.845258

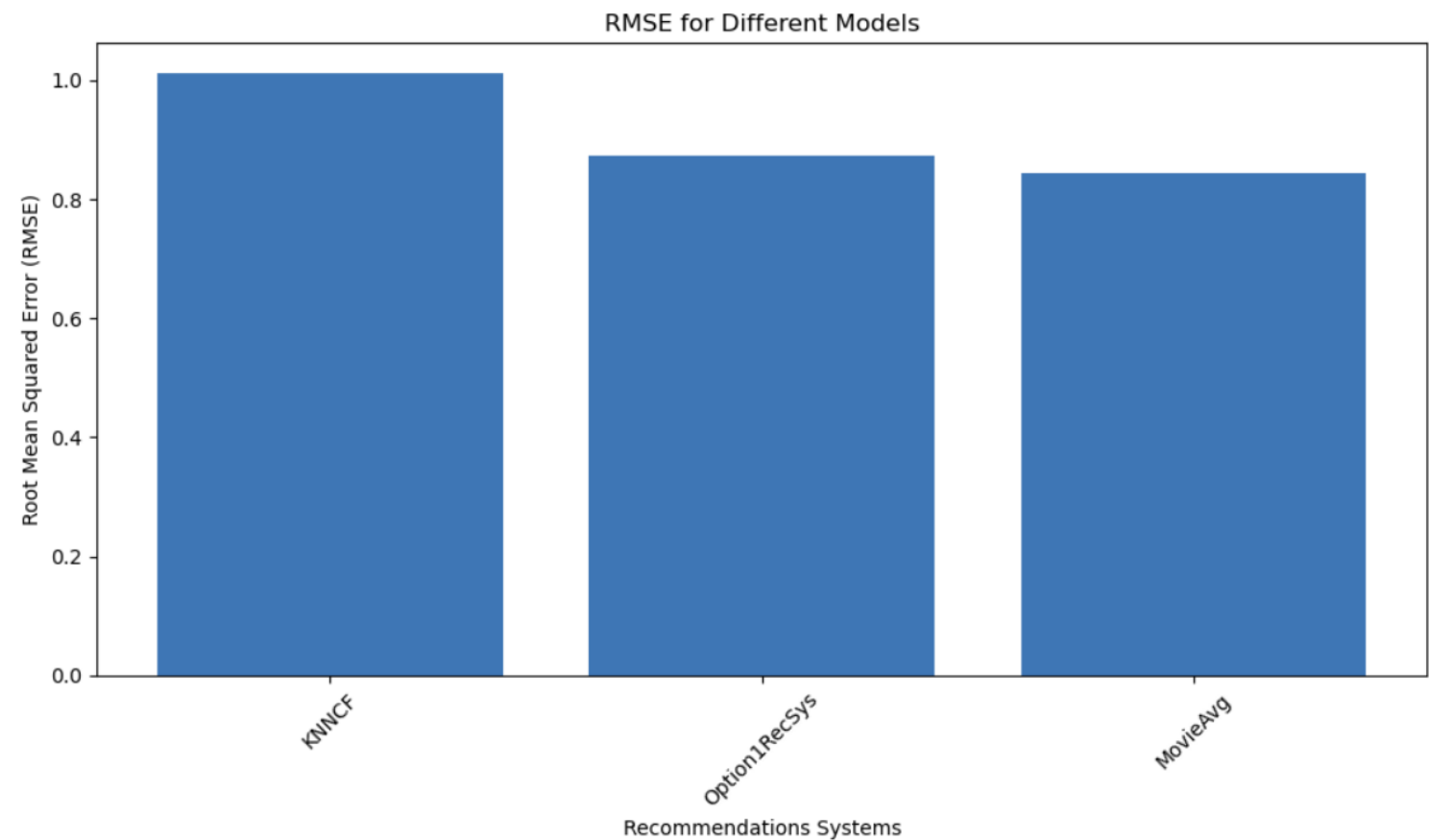
Average Precision across KKNCF, OPTION1RECSYS AND MOVIEAVG



Normalized Discounted cumulative gain(NDCG) across KKNCF, OPTION1RECSYS AND MOVIEAVG



Root Mean Squared Error (RMSE) across KKNCF, OPTION1RECSYS AND MOVIEAVG



References

- Tasks 1 and 2, were referenced from the materials given by the tutor and were based on the Practical Data Science with Python COSC2670 lab.
- Le, J. (2018, April 28). *The 4 Recommendation Engines That Can Predict Your Movie Tastes*. Medium. <https://le-james94.medium.com/the-4-recommendation-engines-that-can-predict-your-movie-tastes-bbec857b8223>
- Team, G. L. (2020, August 4). *Matrix Factorization Explained | What is Matrix Factorization?* Great Learning Blog: Free Resources What Matters to Shape Your Career! <https://www.mygreatlearning.com/blog/matrix-factorization-explained/>
- *movielens/SVD_Model.ipynb at master · khanhnamle1994/movielens*. (n.d.). GitHub. https://github.com/khanhnamle1994/movielens/blob/master/SVD_Model.ipynb
- Python, R. (n.d.). *Split Your Dataset With scikit-learn's train_test_split()* – Real Python. Realpython.com. <https://realpython.com/train-test-split-python-data/>
- *Evaluation Measures in Information Retrieval*. (n.d.). Pinecone. <https://www.pinecone.io/learn/offline-evaluation/>
- Zach. (2020, September 3). *How to Calculate RMSE in Python*. Statology. <https://www.statology.org/rmse-python/>