A Movie Recommendation Systems Using K-Nearest Neighbors and Singular Value Decomposition

Angel Salges
Electrical and Computer Engineering Department
Florida State University
Tallahassee,Florida,USA
ads19n@fsu.edu

Simon Foo, Ph.D
Electrical and Computer Engineering Department
Florida State University
Tallahassee,Florida,USA
foo@eng.famu.fsu.edu

Ming Yu, Ph.D
Electrical and Computer Engineering Department
Florida State University
Tallahassee,Florida,USA
mingyu@eng.famu.fsu.edu

***Abstract*—Recommendation systems have become essential in the entertainment industry, helping users discover content tailored to their preferences. This paper presents a movie recommendation system utilizing two distinct approaches: K-Nearest Neighbors (KNN) and Singular Value Decomposition (SVD), both enhanced by cosine similarity. The system leverages the MovieLens (small) dataset, which includes user ratings for various movies, to generate personalized movie recommendations. KNN computes recommendations by finding the most similar movies based on user ratings, while SVD reduces dimensionality, capturing latent features for improved recommendations. This paper evaluates their effectiveness in producing relevant suggestions. The results demonstrate that while KNN provides highly personalized recommendations, SVD offers better scalability and handles sparse data more effectively. This paper discusses the strengths, limitations, and potential future improvements of both methods in building more robust recommendation systems. The findings suggest that hybrid approaches combining KNN and SVD could further enhance the quality of movie recommendations for diverse user preferences.**

Keywords—Movie Recommendation System, KNN, SVD, Cosine Similarity, Collaborative Filtering, Machine Learning, Data Science, Recommender Systems, MovieLens.

I. Introduction

Recommendation systems play a vital role in enhancing user experience by providing personalized suggestions tailored to individual preferences. These systems are widely adopted in various domains such as e-commerce, social media, and entertainment platforms. In the film industry, movie recommendation systems have become essential tools for platforms like Netflix, Amazon Prime, and other streaming services, helping users discover content that aligns with their tastes. By analyzing user behavior and historical ratings, recommendation systems can predict and recommend movies that a user is likely to enjoy.

There are two major approaches for implementing recommendation systems: collaborative filtering and content-based filtering. Collaborative filtering focuses on user behavior, relying on similarities between users or items, whereas content-based filtering analyzes movie attributes or features. Among collaborative filtering techniques, K-Nearest Neighbors (KNN) and Matrix Factorization-based methods like Singular Value Decomposition (SVD) are widely employed due to their simplicity, scalability, and effectiveness.

The KNN algorithm identifies relationships between items (movies) or users by measuring similarity, such as cosine similarity, and uses the ratings of similar neighbors to recommend movies. On the other hand, SVD is a matrix factorization method that reduces the user-item interaction matrix into latent factors, enabling the system to predict ratings for unseen user-movie pairs. Both approaches offer unique strengths: KNN is interpretable and easy to implement, while SVD is efficient for handling sparse datasets and uncovering hidden patterns in data.

This paper presents an analysis of KNN with cosine similarity and SVD for a movie recommendation system using the MovieLens dataset. The performance of both models is evaluated based on their ability to recommend relevant movies.

II. Methodology

A. Cosine Similarity

Cosine similarity is a widely used metric to measure the similarity between two vectors based on their angle in a multi-dimensional space. It is particularly effective for high-dimensional and sparse data, such as the user-movie rating matrix.

The cosine similarity between two vectors A and B is defined as:

$$\text{Cosine Similarity} = A \cdot B / \|A\|\|B\|$$

Where:

- $A \cdot B$ is the dot product of the two vectors.
- $\|A\|$ and $\|B\|$ are the magnitudes (Euclidean norms) of vectors A and B. [1]

For a user-movie rating matrix, the vectors AA and BB represent rows (users) or columns (movies). The similarity score ranges from -1 to 1:

- 1 indicates perfect similarity.
- 0 indicates no similarity.
- -1 indicates complete dissimilarity.

B. K-Nearest Neighbors (KNN) with Cosine Similarity

The K-Nearest Neighbors (KNN) algorithm is a non-parametric method that finds the "k" most similar items (or users) based on a similarity measure. In this project, we apply cosine similarity to identify similar movies. It first constructs a matrix where rows represent movies and columns represent users. Each cell contains a rating given by a user to a movie. Then, it calculates pairwise cosine similarity between movies using the user-movie rating matrix. This generates a similarity matrix. Finally, for given

target movie, it identifies the k-nearest movies with the highest cosine similarity scores.

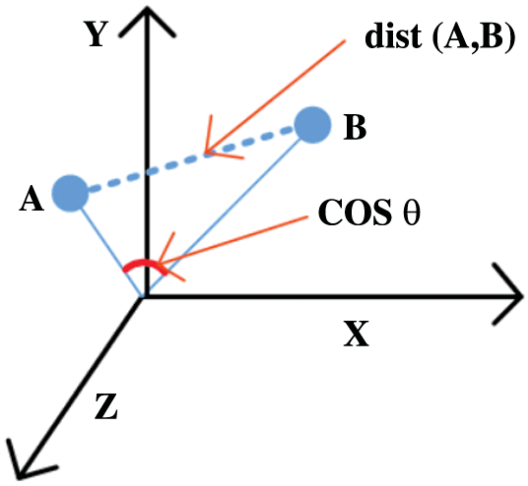

Figure 1: K-Nearest Neightbor with Cosine Similarity [2]

C. *Singular Value Decomposition (SVD) with Cosine Similarity*

Singular Value Decomposition (SVD) is a matrix factorization technique used in collaborative filtering to predict missing ratings. It decomposes the user-movie rating matrix RR into three matrices:

$$R \approx U \Sigma V^T$$

Where:

- U is the user matrix (latent features for users).
- Σ is a diagonal matrix of singular values (importance of features).
- V^T is the item matrix (latent features for movies). [3]

To do this process we first ecompose the user-movie rating matrix using SVD into its latent components U, Σ, and V^T. then use the decomposed matrices to reconstruct the rating matrix and predict ratings for missing user-movie pairs. After that, compute cosine similarity between the latent vectors of movies from V^T to determine relationships between movies. Finally, use the similarity to rank movies and recommend the top-N movies to the user

III. Dataset Analysis

The MovieLens dataset is a widely-used benchmark dataset for building and evaluating recommendation systems. It is provided by the GroupLens Research Group at the University of Minnesota and has been instrumental in advancing collaborative filtering techniques. The dataset contains user ratings for movies and associated metadata, making it ideal for testing both memory-based and model-based recommendation algorithms.

This dataset was chosen as it had many good qualities that would help the development of this project. It is well structured, clean and labeled which allows for clear identification for the parameters this project will use. It has a size that is sufficient for this research project and it will not be computationally expensive which can happen with larger datasets. It is a very popular dataset for benchmarking recommendation systems and allows for comparison with existing research. It has high amounts of metadata which helps the recommendation algorithm and allows extensions of the system in the future.

A. *MovieLens Starting Set*

The MovieLens dataset is available in multiple versions, varying in size to suit different levels of experimentation. In this paper, the MovieLens Latest Dataset (small) is used, which contains a manageable amount of data for analysis and testing. Last updated September, 2018 [5]

Key characteristics of the MovieLens Latest Dataset:

- Number of Users: 610
- Number of Movies: 9,742
- Number of Genres: 20
- Number of Ratings: 100 838
- Rating Scale: 1 to 5 (whole-star ratings)
- Sparsity: The dataset is sparse, as most users have rated only a small subset of the movies.

The dataset consists of four main *.csv* files: *ratings, movies, tags, and links*. This paper focuses on the first two. The *ratings* file contains the ratings of each user for any movie.

Column	Description
userId	Unique identifier for the user
movieId	Unique identifier for the movie
rating	Rating given by the user (1-5)
timestamp	Timestamp of the rating

The *movies* file contains movie metadata, such as titles and genres.

Column	Description
movieId	Unique identifier for the movie
title	Title of the movie (includes release year)
genres	Pipe-separated list of genres

The *tags* file contains user-generated tags for movies, which can be used for content-based filtering. The *links* file provides links to external movie databases like IMDb and TMDB.

B. *Modified Dataset*

This dataset was modified for the purpose of this paper to address some of the issues such as data sparsity, formatting, and rating biases. Data sparsity is a problem for most recommendation algorithms. In this dataset, it is caused as many users have only rated a handful of movies. This was adressed by removing users that had less than 50 reviews. The genre tags for each movie were on a single cell divided by a pipe " | ". This causes problems for reading genres unless each tag is adressed individually. The ratings are biased to somewhere between 3 and 4 as users tend to prefer decent quality movies.

Key characteristics of the MovieLens Latest Dataset:

- Number of Users: 385
- Number of Movies: 9,742
- Number of Genres: 20
- Number of Ratings: 93 812
- Rating Scale: 1 to 5 (whole-star ratings)

C. *Data Preprocessing*

The dataset was also checked for any duplicates for user ratings and any found were removed. A user-movie matrix was constructed, where rows represent movies, columns represent users, and the values correspond to ratings. Any missing ratings are treated as zeros for matrix factorization techniques like SVD. Movies were indexed to facilitate fast lookup for similarity calculations.

D. *Modified Dataset Analysis*

The dataset created was taken into MATLAB to characterize the information within it and analize its contents. We validated that there were no duplicate rows within our dataset.

```
Duplicate rows:
    Var1    userId    movieId    rating    timestamp    title    genres
    ____    ______    _______    ______    _________    _____    ______
```

Figure 2: No Duplicate Rows Validation

The ratings were distributed the following way:

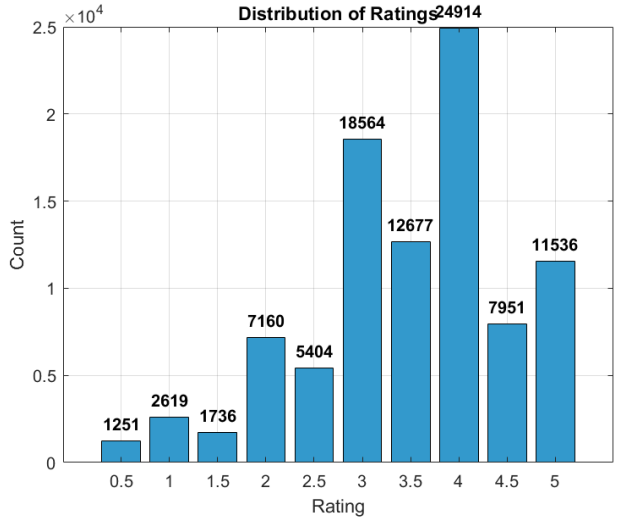

Figure 3: Rating Distribution

These skewness towards movies rated at a 3 or a 4 within the dataset still persists in this project's data. We believe it is essential to recommend highly rated movies to improve the likelihood of user satisfaction.

The genres were divided the following way:

```
Top 5 Genres:
    uniqueGenres      Var2
    ____________      ____

    {'Drama'    }     38944
    {'Comedy'   }     36698
    {'Action'   }     28302
    {'Thriller' }     24415
    {'Adventure'}     22202
```

Figure 4: Most Present Genres

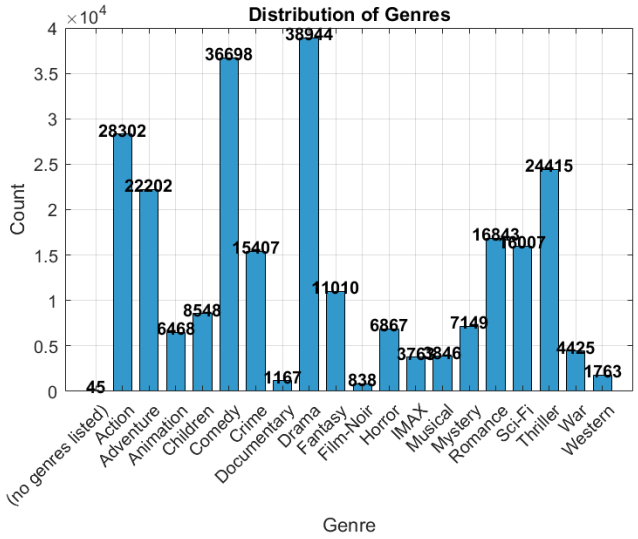

Figure 5: Genre Distribution

The genres with the most ammount of ratings are in order: *"Drama"*, *"Comedy"*, *"Action"*, *"Thriller"*, and "Adventure". These correlates to popularity and users are more likely to watch these genre of movies than genres like *"Documentary"* or "*Film-Noir"*.

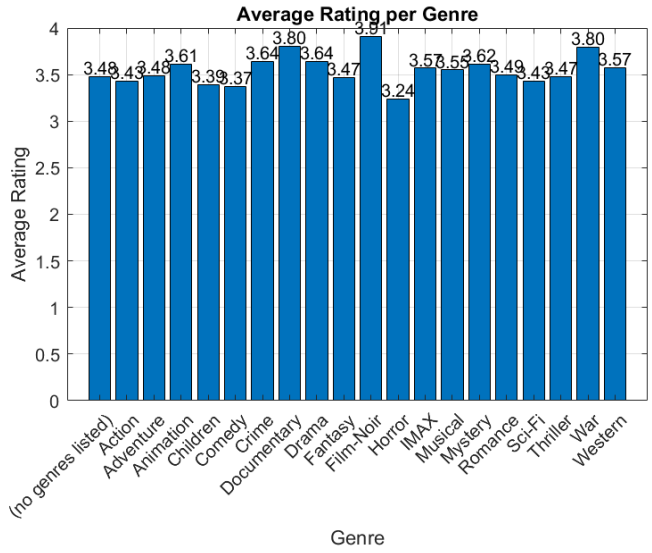

Figure 6: Average Rating by Genre

We see that films from certain genres are highly rates despite being fewer. This can be because of several factors. Genre prestige, lower popularity making them niche

films who a handful of users that are dedicated to film watching rate decreasing the rating variation, and incomplete data.

IV. Results

To test the recommendation system we tried inputing several movies and seeing the recommendations and simliarity score based on the cosine similarity. This paper evaluated the modified dataset with and without the cut for users at a minimum of 50 ratings.

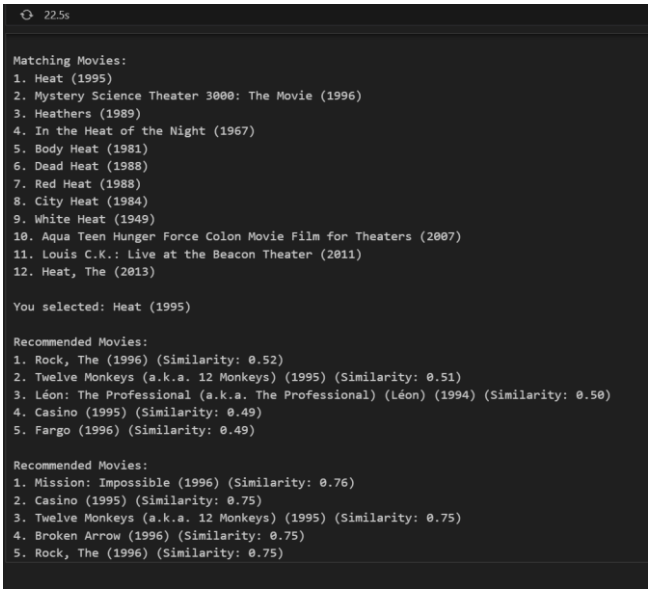

Figure 7: "Heat (1995)" Recomendation Full

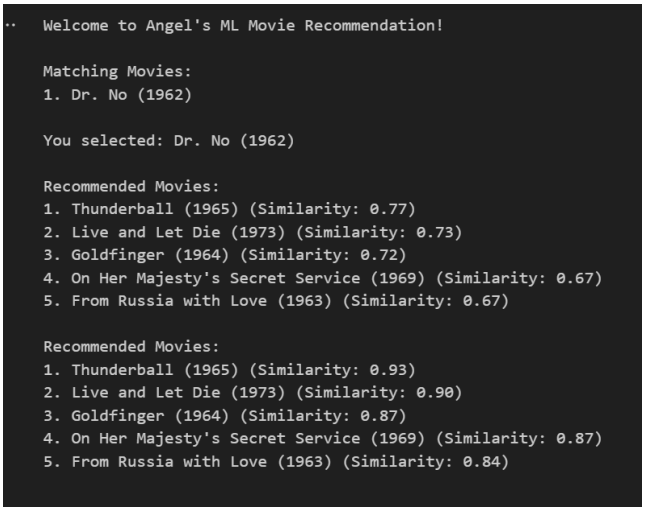

Figure 8:"Dr. No (1962)" Recomendation Full

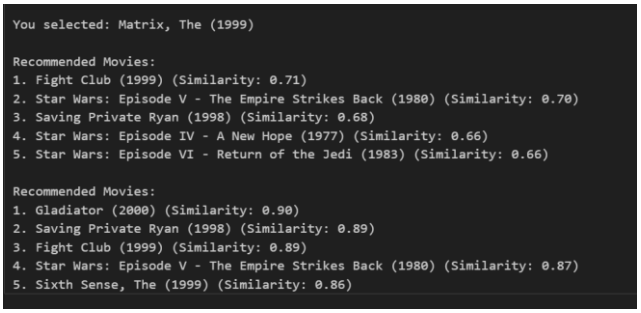

Figure 9: "The Matrix (1999)" Recomendation Full

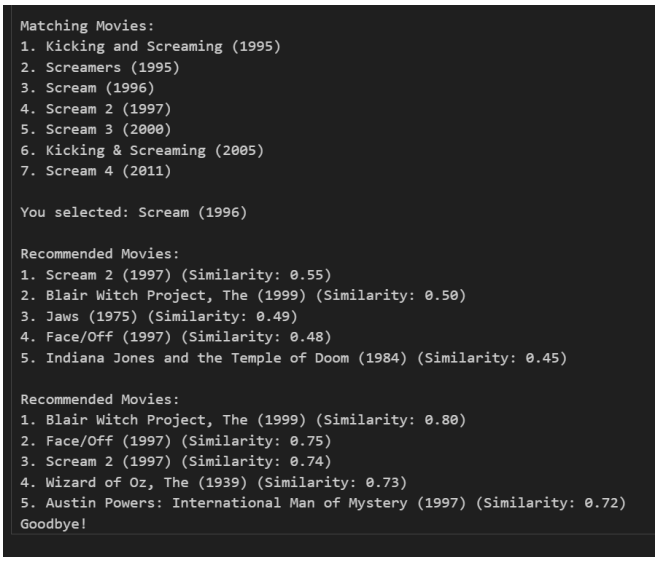

Figure 10: "Scream(1996)" Recomendation Full

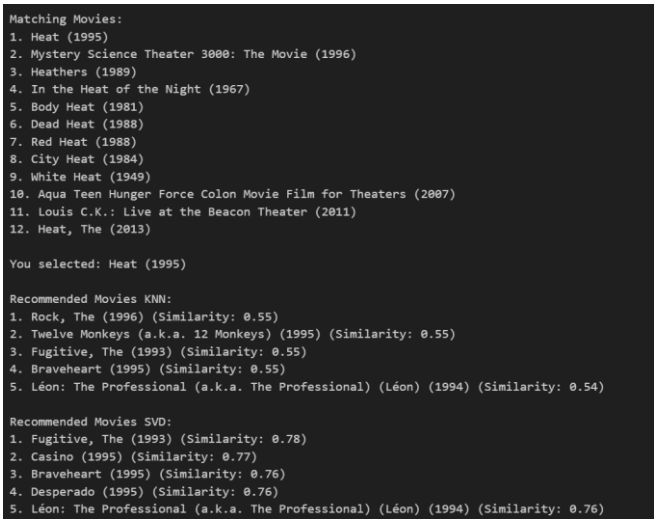

Figure 11: "Heat (1995)" Recomendation Cut

```
Matching Movies:
1. Dr. No (1962)

You selected: Dr. No (1962)

Recommended Movies KNN:
1. Thunderball (1965) (Similarity: 0.78)
2. Goldfinger (1964) (Similarity: 0.74)
3. Live and Let Die (1973) (Similarity: 0.72)
4. Man with the Golden Gun, The (1974) (Similarity: 0.70)
5. From Russia with Love (1963) (Similarity: 0.69)

Recommended Movies SVD:
1. Thunderball (1965) (Similarity: 0.92)
2. Live and Let Die (1973) (Similarity: 0.89)
3. Goldfinger (1964) (Similarity: 0.87)
4. On Her Majesty's Secret Service (1969) (Similarity: 0.86)
5. Spy Who Loved Me, The (1977) (Similarity: 0.84)
```

Figure 12: "Dr. No (1962)" Recomendation Cut

```
You selected: Matrix, The (1999)

Recommended Movies KNN:
1. Fight Club (1999) (Similarity: 0.77)
2. Star Wars: Episode V - The Empire Strikes Back (1980) (Similarity: 0.75)
3. Star Wars: Episode VI - Return of the Jedi (1983) (Similarity: 0.72)
4. Star Wars: Episode IV - A New Hope (1977) (Similarity: 0.72)
5. Saving Private Ryan (1998) (Similarity: 0.71)

Recommended Movies SVD:
1. Gladiator (2000) (Similarity: 0.91)
2. Fight Club (1999) (Similarity: 0.90)
3. Saving Private Ryan (1998) (Similarity: 0.90)
4. Star Wars: Episode V - The Empire Strikes Back (1980) (Similarity: 0.88)
5. Sixth Sense, The (1999) (Similarity: 0.88)
```

Figure 13:"The Matrix (1999)" Recomendation Cut

```
Matching Movies:
1. Kicking and Screaming (1995)
2. Screamers (1995)
3. Scream (1996)
4. Scream 2 (1997)
5. Scream 3 (2000)
6. Kicking & Screaming (2005)
7. Scream 4 (2011)

You selected: Scream (1996)

Recommended Movies KNN:
1. Scream 2 (1997) (Similarity: 0.57)
2. Blair Witch Project, The (1999) (Similarity: 0.53)
3. Jaws (1975) (Similarity: 0.52)
4. Face/Off (1997) (Similarity: 0.51)
5. Misery (1990) (Similarity: 0.49)

Recommended Movies SVD:
1. Blair Witch Project, The (1999) (Similarity: 0.81)
2. Wizard of Oz, The (1939) (Similarity: 0.75)
3. Scream 2 (1997) (Similarity: 0.75)
4. Face/Off (1997) (Similarity: 0.74)
5. Austin Powers: International Man of Mystery (1997) (Similarity: 0.73)
```

V. Discussion

This paper found that the implementation of KNN and SVD algorithms for user based filtering in a movie recommendation algorithm is very effective as the similarity scores between films is high and the recommendations are generally related to the query in the themes and genres from the movies.

Implemeting a cuttoff requiring more than 50 reviews always improved the KNN alogrithm similarity scores while slightly improving or decreasing the SVD similarity scores. Both algorithms tend to recommend the same movies and in some cases the KNN tends to recommend more similar movies in the sense of themes and genres while SVD maximizes the similairty score that is based on the distance between the algorithm latent features.

A. Strengths and Weaknesses of Similarity Based Comparison

A strength of user-based recommendation system is that if two movies have similar user rating patterns, they are likely to be recommended together. However, this can lead to problems such as in “Scream (1996)”, a horror slasher where the SVD recommends “The Wizard of Oz (1939)” a family classic movie. Therefore, some content based integration should be done to prevent this great thematical outliers.

These methods work extremely well with small or sparse datasets which can help cases where data is a problem. Due to this usability while lacking data, they are great candidates for “Cold Starts” (where there is a lack of user data and preference) as they can work with item-item recommendations.

A problem with this system is that it does not produce explicit predictions and that complicates the use of objective measures such as accuracy and RMSE. Scalability also becomes an issue as datasets grow the pairwise similarities also increase drastically.

Another issue is the lack of personalization. As there are no user profiles, it relies on existing data to do any recommendations. This is also affects KNN by the lack of global relationships which are only present through matrix factoriztion techniques like SVD. Therefore it makes it harder to give more tailored recommendations to each user.

B. Future Work

This project focused on making a basis for a recommendation algorithm and evaluating the usage of KNN and SVD algorithms with cosine similarity with no user data and sparse existing data. This system can be improved by making a hybrid approach by combining this two algrorithms to leverage both local relationships and global patterns. An example would be using KNN cosine similarity to recommend similar movies, and validate recommendations with predicted ratings with SVD. This would also allow the usage of quantifiable metrics such as RSME or MAE.

A further analysis of the data could also prove useful in evaluating user profiles ratings and establish user-user similarity scores. This could help generate user profiles that match closely with the request user profile, improving quality recommendations.

VI. Conclusion

In this paper, we developed a movie recommendation system using two collaborative filtering techniques: K-Nearest Neighbors (KNN) with cosine similarity and Singular Value Decomposition (SVD) with cosine similarity. The system was implemented and tested on the MovieLens dataset, which provided user ratings and movie metadata.

Although the current implementation focuses solely on user rating patterns, incorporating additional features such as movie genres or user demographics could further improve the system's relevance and diversity. Moreover, combining collaborative filtering with content-

based techniques into a hybrid model may provide a more comprehensive recommendation framework.

Future work will explore integrating explicit rating prediction to evaluate recommendation accuracy using metrics such as RMSE and extending the system to include ranking-based metrics.

This study demonstrates the effectiveness of collaborative filtering techniques in building personalized movie recommendation systems. By comparing KNN and SVD with cosine similarity, we highlight the trade-offs between simplicity, scalability, and accuracy, providing a foundation for future advancements in recommendation algorithms.

VII. References

[1] J. Han, Data Mining: Concepts and Techniques, Amsterdam: Elsevier/Morgan Kaufmann, 2011.

[2] M. Alghobari, "Location-Aware Personalized Traveler Recommender System (LAPTA) Using Collaborative Filtering KNN.," *Computers, Materials & Continua,* no. 69, 2021.

[3] P. Husb, "On the Use of the Singular Value Decomposition for Text Retrieval," *U.S. Department of Energy Office of Scientific and Technical Information,* 2001.

[4] M. Harper and J. Konstan, "The MovieLens Datasets: History and Context," *ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19.*, vol. 5, no. 4, pp. 1-19, 2015.

[5] GroupLens, "MovieLens," [Online]. Available: https://grouplens.org/datasets/movielens/. [Accessed 24 November 2024].