CSP 571 - PROJECT REPORT

Optimizing Movie Recommendation for Enhanced User Experience.

Project Team:

1. Ganya Janardhan (A20517083)
2. Jayanth Chidananda (A20517012)
3. Vismaya M (A20519405)

1. Abstract

In the era of digital entertainment, personalized movie recommendations play a pivotal role in enhancing the user experience on streaming platforms. In today's world, nearly everyone prefers to watch movies in the comfort of their own homes. This is exploited by streaming goliaths like Netflix and Amazon, whose OTT services have a sizable subscriber base. They maintain these customers by employing a recommendation engine that provides them with a list of films they would want to watch based on their prior streaming behavior; they even compare the reviews of other users who have viewed and rated a given film or television programme.

Ultimately, this project aims to create a movie recommendation system that not only recommends movies effectively but also caters to user preferences, fosters user engagement, and enhances the overall user experience. The findings and insights gained from this project contribute to the ongoing evolution of recommendation systems in the entertainment industry.

2. Overview

A recommendation engine offers consumers recommendations based on their browsing history, user-set preferences, and user-provided ratings. Every time someone chooses a TV show on Netflix using the "You May Also Like..." option or buys a product that Amazon recommends, powerful recommendation algorithms are deployed. Most platforms have a "Recommended for You" or "Suggestions" section where you can find personalized movie recommendations. These suggestions are based on your ratings and viewing history.

Some of the recommendation algorithms commonly used by Netflix, Amazon, etc.

1. Collaborative Filtering:
User-Based Collaborative Filtering: This approach identifies users who are similar to the target user and recommends items that those similar users have liked.
Item-Based Collaborative Filtering: It focuses on the similarity between items and recommends items that are similar to those the user has shown interest in.

2. Content-Based Filtering: This method recommends items based on their attributes and characteristics, such as movie genre, actors, director, or product features. It matches item attributes with user preferences.

3. Hybrid Recommender Systems: Amazon and Netflix often employ hybrid recommendation systems that combine collaborative filtering and content-based filtering to provide more accurate and diverse recommendations.

In this project we will use user-user based collaborative filtering technique (Matrix Factorization and Cosine similarity) and Item-item based collaborative filtering (KNN) to predict recommendations to users.

3 Dataset

3.1 Dataset used

The datasets describe ratings and free-text tagging activities from MovieLens, a movie recommendation service. It contains 20000263 ratings and 465564 tag applications across 27278 movies.

File	***Size (n×p)***	***Features***	***Description***
Ratings	20M*4	‘userId’, ‘movieId’, ‘Rating’, ‘timestamp’	Contains User IDs and ratings for movies. Ratings are provided on 5-star scale
Movie	27K*3	‘movieId’, ‘title’, ‘genres’	It has a genre information and also used as lookup to identify movies
Tags	465K*4	‘userId’, ‘movieId’,’tag’, ’timestamp’	These are user generated tags for movies

Genome Tags	1128*2	‘tagId’, ‘tag’;	Tga description are provided for TagIDs
Genome Scores	11M*3	‘movieId’,’tagId’, ‘relevance’,	Relevance score is provided for a tag with associated movie
Link		‘movieId’, imdbId’, ’tmdbid’	This dataset contains identifiers for linking to other sources

Due to restrictions in the local RAM, only 4M data points were considered from the rating dataset.

3.2 Data Analysis

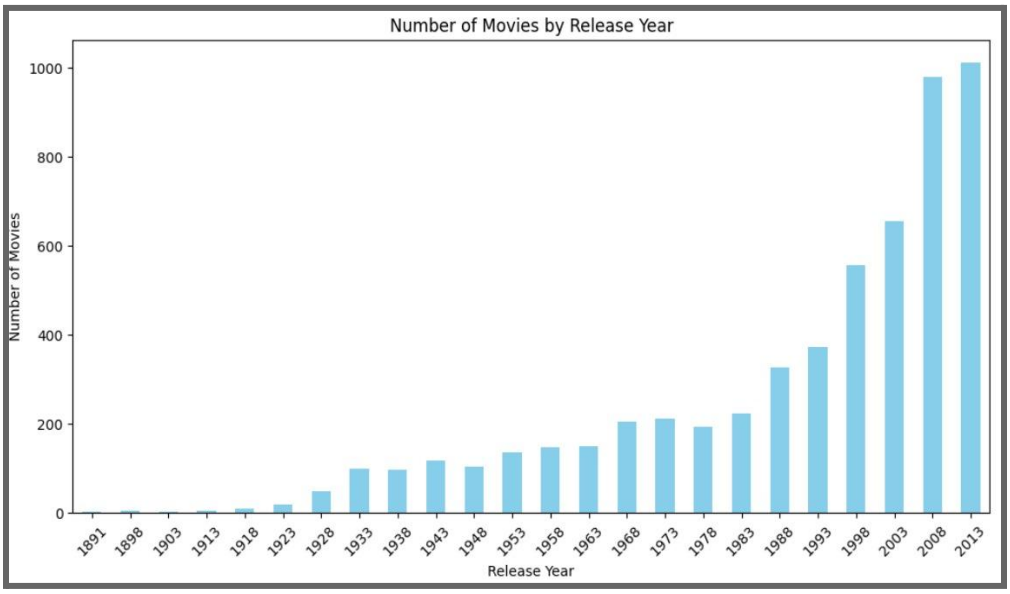

Figure 1: Movie count by year

The figure above depicts the number of movies in the dataset each year; as shown in the plot, the number of movies in the 2000s is substantially higher than in prior years.

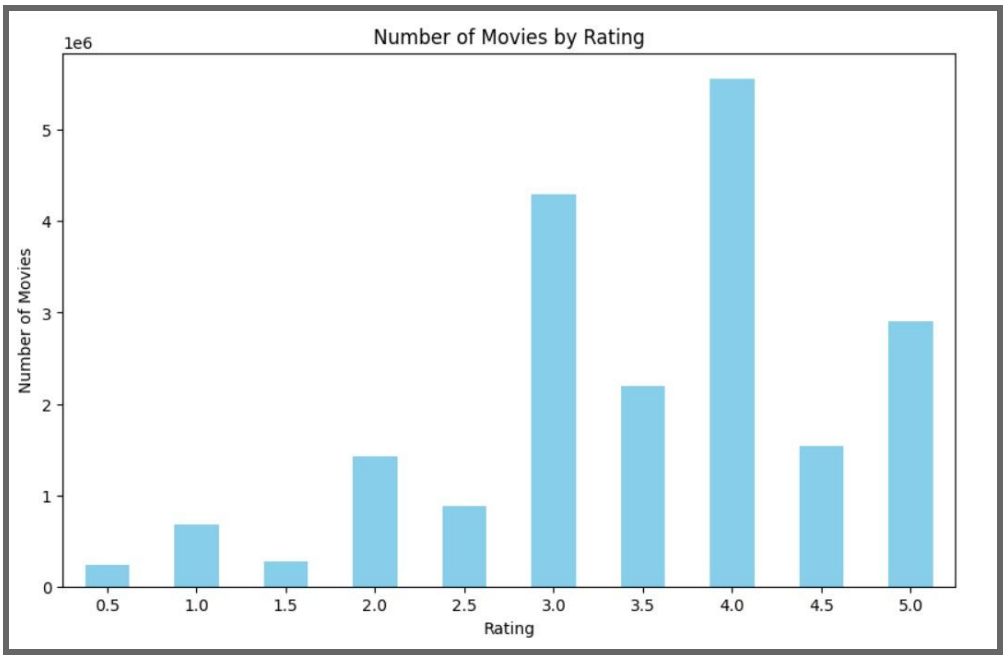

Figure 2: Movie count by Ratings

The plot above gives a visualization of the number of movies and their ratings. Maximum number of movies have a rating of 4, followed by 3

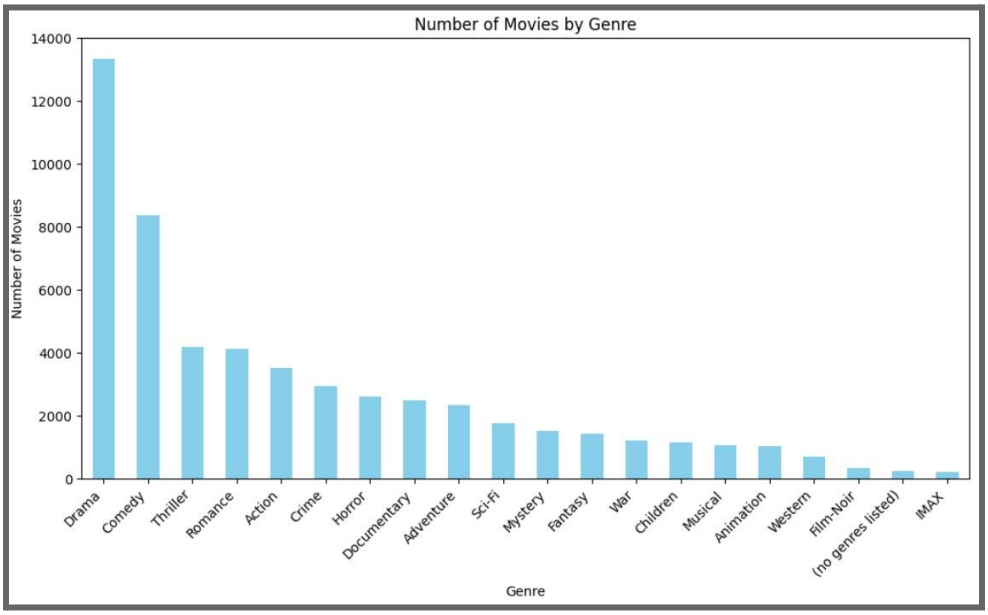

Figure 3: Movie count by Genre

The plot above visualizes the movies and their genres. It is noticeable that the maximum number of movies are classified as “Drama”.

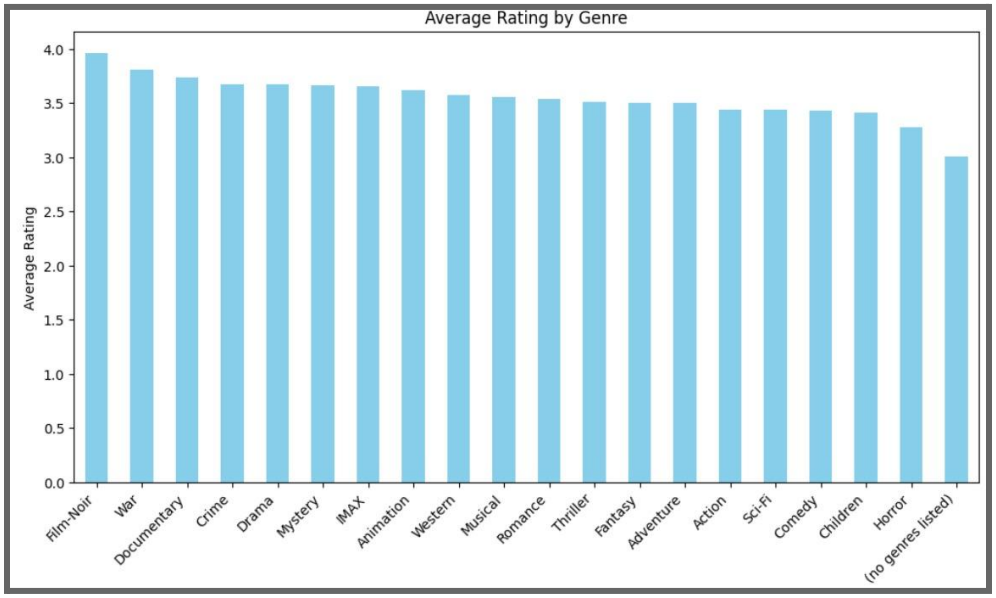

Figure 4:Average rating by genre

The plot above depicts the average rating of all movies by genre; it can be seen that the average rating in each genre ranges from 3 to 4.

3.3 Data Processing

There are 4 million user rating data points in the rating dataset. Every row shows a specific user's rating for a certain movieId. As a result, every row is distinct for every user and movieId. A user and rating matrix is created by pivoting the rating dataset. Because not every user has seen every movie, this matrix is sparse and will have a large number of null values. Movies not seen by at least 1% of all users were eliminated in an effort to decrease the sparsity of the matrix and increase the model's accuracy. This reduced the total number of movies from 19K to 3K. Then, all the null values were replaced with 0.

4. Methodology

For the recommendation system in this research, three collaborative filtering approaches will be used.

[1] The first one is a model-based approach that uses matrix factorization. In essence, it makes a matrix, with each row representing the user and the features representing the movie title. A straightforward illustration would be as follows: Person A has watched movies M1 and M2 and gives a high rating for both movies. Person B has watched movie M2 and gives a high rating for M2. Now, our model suggests movie M1 for Person B. In general, matrix factorization algorithms are more effective since they may uncover implicit connections between various users and films. Here is the mathematical formulation of the matrix factorization of R (user-movie matrix):

$$R \sim P.\ Q^T = \hat{R}$$

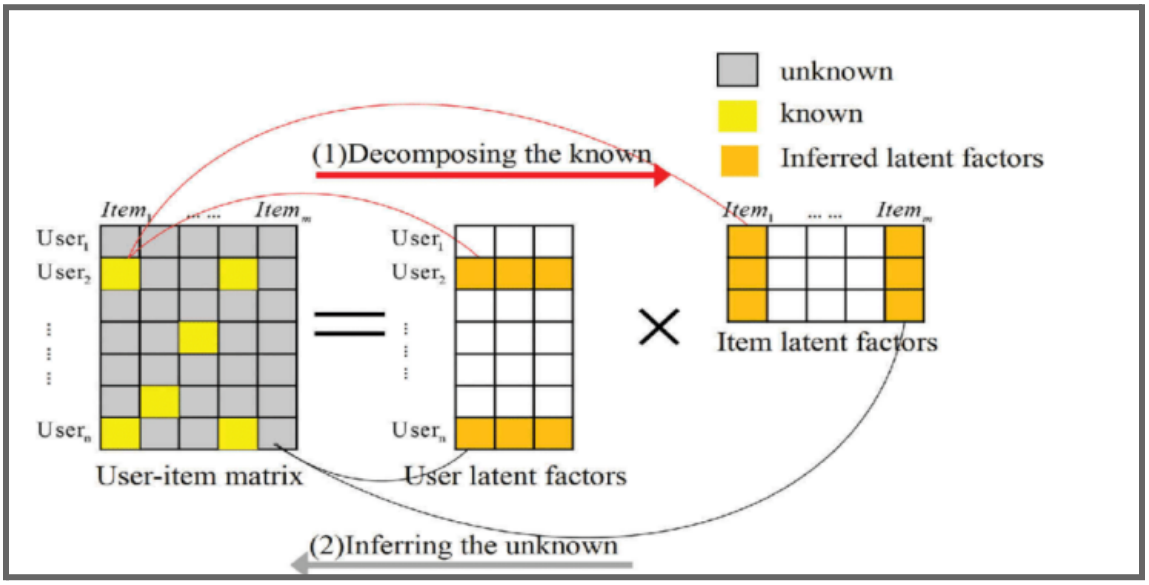

Figure 5: Matrix factorization illustration

The strength of linkages between users and features will be included in a P= $U \times k$ matrix if we assume that k hidden characteristics influence the relationship in the matrix with U users and M movies. A $M \times k$ matrix called Q will hold the strength of the links between features and movies. We utilise a gradient descent approach to iteratively adjust the values of a randomly initialised P & Q in order to estimate P & Q until $P.Q^T$ is sufficiently near to R. The error (E) that needs to be minimized so that we achieve this can be written as:

$$P, Q = argmin_{P,Q}(R - P.Q^T) + \frac{\beta}{2}\left(||P||^2 + ||Q||^2\right)$$

The second term is a penalty term that is used to avoid overfitting similar to the ridge regression equation.

[2] The second approach used is Cosine Similarity which compares the similarity between two non-zero vectors (irrespective of their size). In simple words, this is the cosine of angle between the two vectors.
Mathematically,

$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \, \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

Cosine Similarity Formula

The result obtained is a value that ranges between 0 and 1. As the angle gets smaller, the value of cosine similarity gets closer to 1, this implies that the two vectors are highly similar.

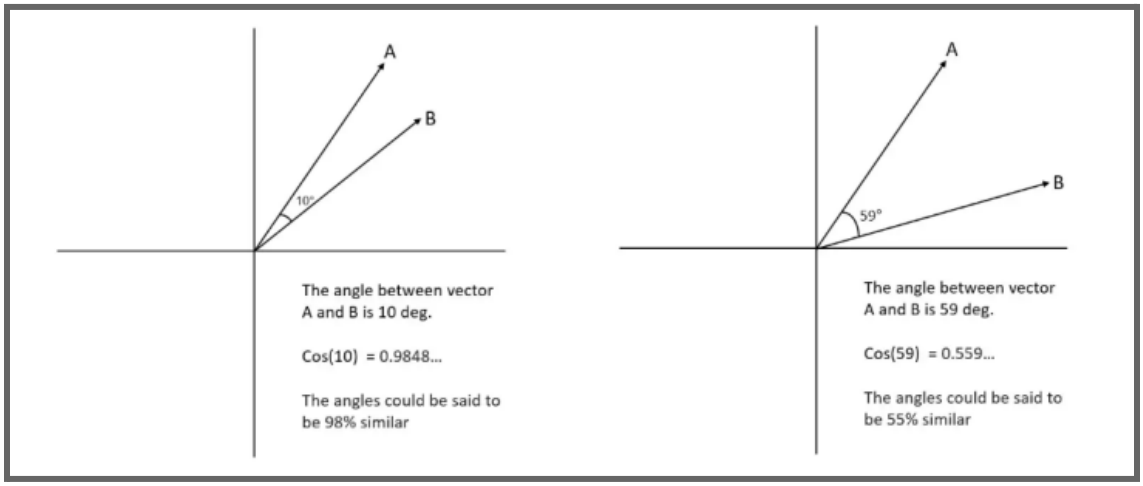

Figure 6: Left: Two vectors with 98% similarity based on cosine angle between the vectors. Right: Two vectors with 55% similarity based on cosine angle between the vectors

The use of cosine similarity has two main benefits. First of all, two vectors may still have a lower angle between them even when there is a very huge Euclidian distance between them; the more the similarity, the smaller the angle. Furthermore, in a multidimensional space, the cosine similarity captures the angle between them and not the magnitude.

[3] The third approach is K-Nearest Neighbors (KNN) which helps identify movies or users with similar profiles, facilitating personalized recommendations based on user history and preferences.

Initially, the system finds the similarity between every pair of elements in order to carry out a model-building step. There are other ways to represent this similarity function, including the cosine of those rating vectors or the correlation between ratings. Similarity functions can employ normalised ratings, much like in user-user systems (adjusting, for example, for each user's average rating). Second, the system executes a recommendation stage. It uses the most similar items to a user's already-rated items to generate a list of recommendations. Usually this calculation is a weighted sum or linear regression.

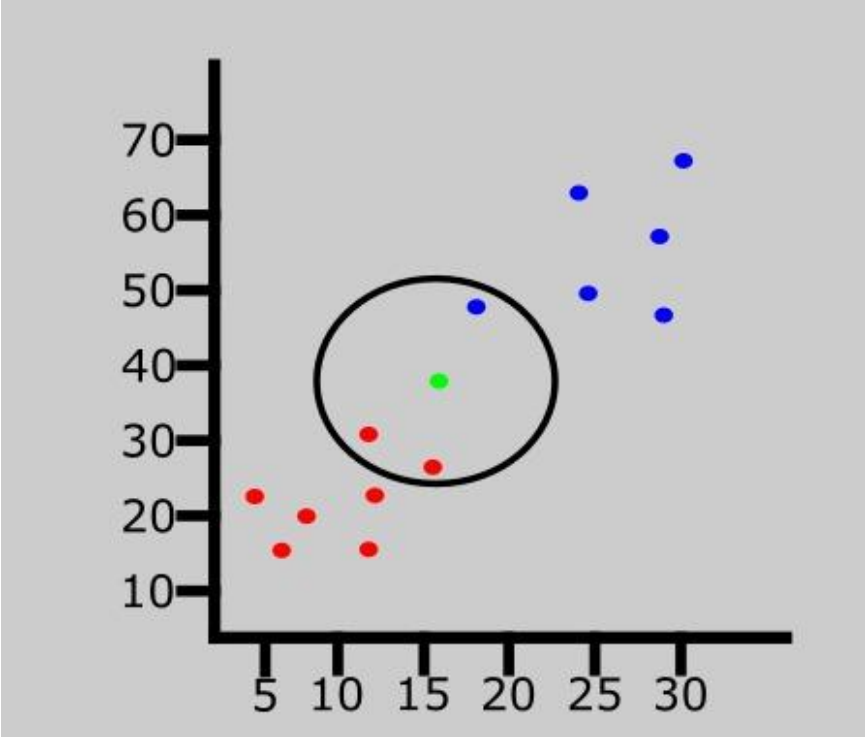

Figure 7: KNN

4.1 Metrics for evaluation

Matrix factorization :

For matrix factorization, Mean Absolute Error (MAE) and Mean Cubic Error (MCE) were used to compare the train and test sets. Mean Absolute Error would give an idea of the absolute error on our predictions and Mean Cubic Error would help us penalize predictions that are far off from the true value. Comparing both metrics would give an idea on the performance of the model.

5 Model Training and Validation

5.1 Matrix Factorization

The dataset was split into train and test matrices. This presents a problem for the train and test division because, in contrast to the traditional division method, we must make sure that every row of the train and test comprises every user. Initially, a copy of the user movie matrix was allocated to the train matrix. The user-movie matrix's size is used to initialize the test matrix to 0s. The user-movie matrix's size is used to initialize the test matrix to 0s. The relevant rating was then assigned in the test matrix for each user, and 80% of the ratings in the train matrix were given a value of 0. The train matrix is used as the input by the model to generate the predicted ratings, and the model is assessed for each test rating and observation. Every value in the anticipated matrix that has a value of more than 5 or less than 0 is capped at that number.

Multiple hyper-parameters, including features (k), learning rate (α), regularization penalty (β), and epochs, were used to train the model. The loss graphs were visualized to find the optimal stopping point after the model had been trained for 50 epochs.

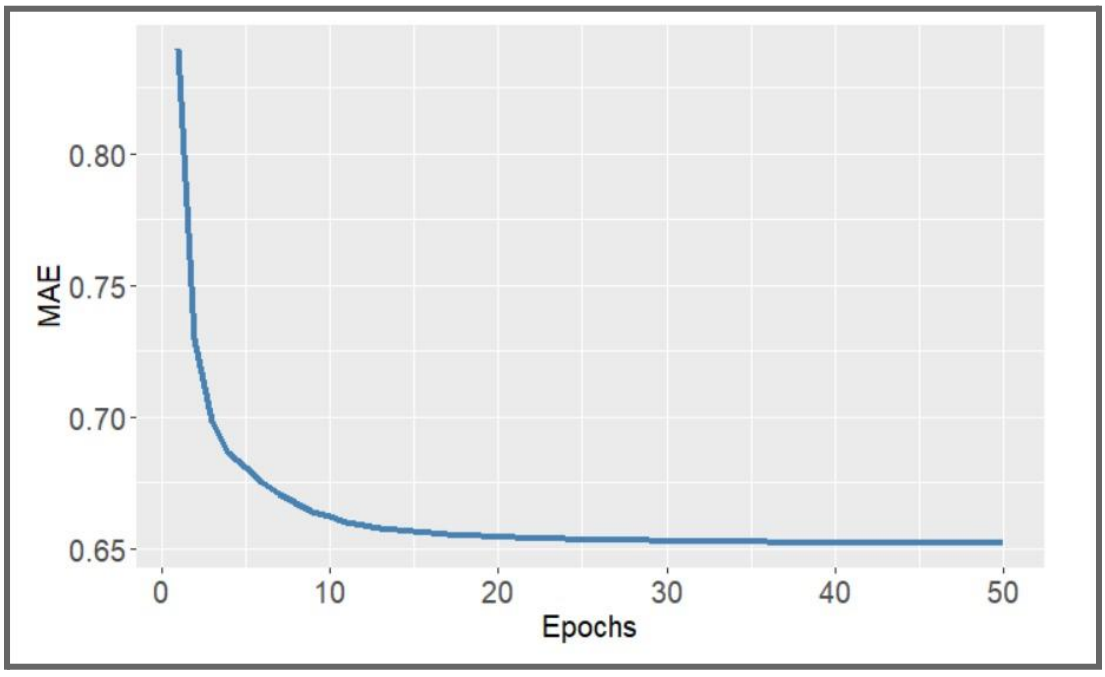

Figure 8: Mean Absolute Error vs Epochs

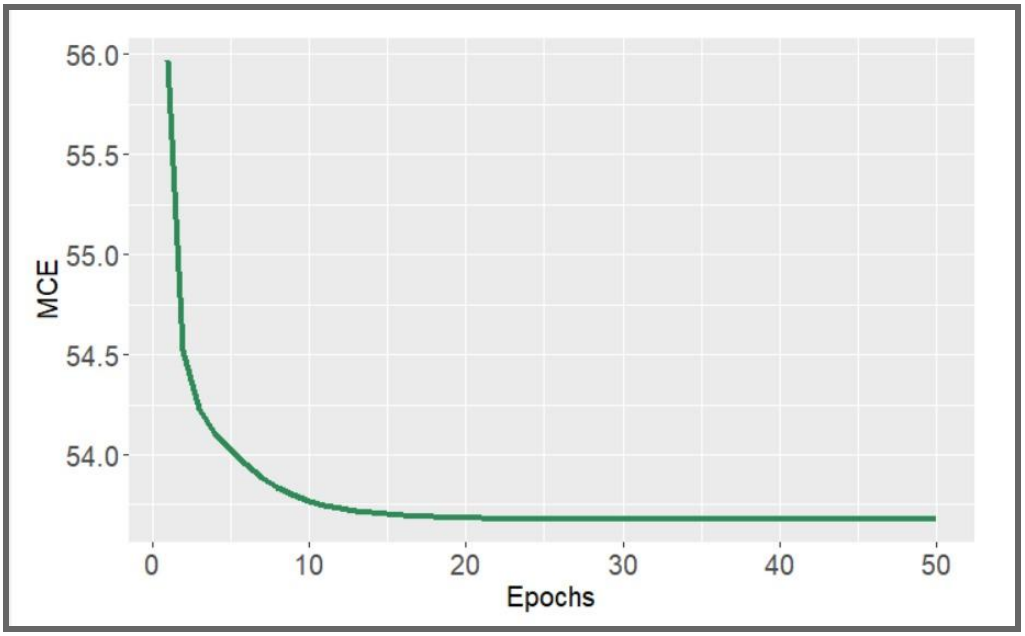

Figure 9: Mean Cubic Error vs Epochs

The ideal number of epochs that the model needed to be trained is 15 as after that the MAE and MCE values start to plateau and there is not much improvement in the model post that.

5.2 Results

The results provided by collaborative filtering (method 1), cosine similarity (method 2) and KNN (method 3) are similar. Let us consider an example to clearly explain this. Here, the user has watched the following movies, and based on this, we make the following observations about both methods.

Sample 1

User ID= 123 has rated the following movies:

Table 1: Movies watched by User ID 123

movieId <chr>	rating <dbl>	title <chr>	genres <chr>
778	5	Trainspotting (1996)	Comedy|Crime|Drama
10	3	GoldenEye (1995)	Action|Adventure|Thriller
104	3	Happy Gilmore (1996)	Comedy
1103	3	Rebel Without a Cause (1955)	Drama
21	3	Get Shorty (1995)	Comedy|Crime|Thriller
762	3	Striptease (1996)	Comedy|Crime
135	1	Down Periscope (1996)	Comedy
736	1	Twister (1996)	Action|Adventure|Romance|Thriller
849	1	Escape from L.A. (1996)	Action|Adventure|Sci-Fi|Thriller

Using method 1, the model recommends the following top movies:

Table 2: Movies recommended by Matrix Factorization

movieId <chr>	rating <dbl>	title <chr>
1089	3.90	Reservoir Dogs (1992)
1094	3.94	Crying Game, The (1992)
1136	3.91	Monty Python and the Holy Grail (1975)
1206	4.03	Clockwork Orange, A (1971)
1223	3.85	Grand Day Out with Wallace and Gromit, A (1989)
1272	3.96	Patton (1970)
1298	4.13	Pink Floyd: The Wall (1982)
1729	3.87	Jackie Brown (1997)
194	3.87	Smoke (1995)
2076	3.93	Blue Velvet (1986)

1-10 of 30 rows | 1-3 of 4 columns

Using method 2, the model recommends the following top movies:

Table 3: Movies recommended by Cosine Similarity

movieId <chr>	rating <dbl>	title <chr>	genres <chr>
1080	4	Monty Python's Life of Brian (1979)	Comedy
1185	5	My Left Foot (1989)	Drama
1193	5	One Flew Over the Cuckoo's Nest (1975)	Drama
1208	5	Apocalypse Now (1979)	Action|Drama|War
1212	5	Third Man, The (1949)	Film-Noir|Mystery|Thriller
1220	5	Blues Brothers, The (1980)	Action|Comedy|Musical
1231	5	Right Stuff, The (1983)	Drama
1249	4	Femme Nikita, La (Nikita) (1990)	Action|Crime|Romance|Thriller
1261	5	Evil Dead II (Dead by Dawn) (1987)	Action|Comedy|Fantasy|Horror
1262	4	Great Escape, The (1963)	Action|Adventure|Drama|War

Using method 3, the model recommends the following top movies:

Table 4: Movies recommended by KNN

```
First $20 Million Is Always the Hardest, The
Bottle Rocket
Ghost and the Darkness, The
Lt. Robin Crusoe, U.S.N.
Browning Version, The
Faat Kiné
Die Hard 2
Women Without Men (Zanan-e bedun-e mardan)
Why Worry?
```

Both the models Matrix Factorization and Cosine Similarity recommended a movie from the series Monty Python.
Meanwhile KNN suggested different movies from the Comedy genre.

Sample 2

Consider another instance. Here, the user has watched the following movies, and based on this we make the following observations about both methods.

User ID = 243 has rated the following movies:

Table 5: Movies watched by User ID 243

	movieId <chr>	rating <dbl>	title <chr>	genres <chr>
7	253	5	Interview with the Vampire: The Vampire Chronicles (1994)	Drama|Horror
9	296	5	Pulp Fiction (1994)	Comedy|Crime|Drama|Thriller
10	318	5	Shawshank Redemption, The (1994)	Crime|Drama
13	356	5	Forrest Gump (1994)	Comedy|Drama|Romance|War
2	150	4	Apollo 13 (1995)	Adventure|Drama|IMAX
3	153	4	Batman Forever (1995)	Action|Adventure|Comedy|Crime
4	165	4	Die Hard: With a Vengeance (1995)	Action|Crime|Thriller
8	288	4	Natural Born Killers (1994)	Action|Crime|Thriller
12	349	4	Clear and Present Danger (1994)	Action|Crime|Drama|Thriller
16	457	4	Fugitive, The (1993)	Thriller

Using matrix factorization, the model recommends the following top movies:

Table 6: Movies recommended by Matrix Factorization

movieId <chr>	rating <dbl>	title <chr>	genres <chr>
1077	4.70	Sleeper (1973)	Comedy|Sci-Fi
1136	4.77	Monty Python and the Holy Grail (1975)	Adventure|Comedy|Fantasy
1196	4.63	Star Wars: Episode V - The Empire Strikes Back (1980)	Action|Adventure|Sci-Fi
1212	4.57	Third Man, The (1949)	Film-Noir|Mystery|Thriller
1214	4.78	Alien (1979)	Horror|Sci-Fi
1234	4.59	Sting, The (1973)	Comedy|Crime
1249	4.58	Femme Nikita, La (Nikita) (1990)	Action|Crime|Romance|Thriller
1285	4.61	Heathers (1989)	Comedy
1292	4.62	Being There (1979)	Comedy|Drama
1960	4.68	Last Emperor, The (1987)	Drama

Using cosine similarity, the model recommends the following top 5 movies:

Table 7: Movies recommended by Cosine Similarity

movieId <chr>	rating <dbl>	title <chr>	genres <chr>
10	4	GoldenEye (1995)	Action|Adventure|Thriller
161	3	Crimson Tide (1995)	Drama|Thriller|War
185	3	Net, The (1995)	Action|Crime|Thriller
329	3	Star Trek: Generations (1994)	Adventure|Drama|Sci-Fi
339	3	While You Were Sleeping (1995)	Comedy|Romance
357	4	Four Weddings and a Funeral (1994)	Comedy|Romance
434	3	Cliffhanger (1993)	Action|Adventure|Thriller
593	5	Silence of the Lambs, The (1991)	Crime|Horror|Thriller

Using matrix factorization, the model recommends the following top 5 movies:

Table 8: Movies recommended by KNN

```
Open Your Eyes (Abre los ojos)
Piranha
I Am Cuba (Soy Cuba/Ya Kuba)
Hollywood Knights, The
Blood Done Sign My Name
Tetsuo II: Body Hammer
On the Silver Globe (Na srebrnym globie)
Dirty Dozen, The
Hunger, The
Azur & Asmar (Azur et Asmar)
Awful Truth, The
I Don't Want to Talk About It (De eso no se habla)
Replacements, The
Polish Wedding
From Hell
Film with Me in It, A
Evelyn
Frozen Assets
Gung Ho
Misery
```

All the three methods recommended movie from the genre Crime/Thriller
Method 1: Femme Nikita, La (Nikita) (1990)
Method 2 : Silence of the lambs, The (1991)
Method 3: from Hell

6. Conclusion

Three recommender systems are used in this research to suggest movies. The user's suggestions from three models appear to be comparable; however, there are some compromises. Matrix factorization requires extra RAM and processing time. Optimisation of the hyperparameters is also necessary. Nonetheless, inference proceeds quickly when the expected matrix is created. More control is provided by defining the quantity of characteristics that would be pertinent to the issue. Using the user as input, the Cosine similarity approach produces recommendations. Although it doesn't use much memory, depending on the approach taken, it performs more slowly when it must be done for a big number of people.KNN is computationally intensive, especially as the size of dataset grows and it Is also sensitive to noise and irrelevant features which impact the accuracy and reliability of KNN.

With an MAE of 0.6, the Matrix factorization model indicates that the projected ratings are not too far off. We can make sure that the projected values are not outliers by comparing the MCE graphs of several models that have had their hyper-parameters adjusted. The model outputs offer recommendations that are based on common sense, and the outputs may be adjusted to offer a variety of suggestions.

7. Future Work

There are many improvements that can be done to improve our user-based recommendations and Item based recommendations. In future work, an enhanced movie recommendation system could be developed by combining the strengths of KNN, matrix factorization, and cosine similarity. Integration of these methods could involve leveraging KNN for its simplicity and interpretability, matrix factorization for handling sparse data and scalability, and cosine similarity for robustness to varying scales. Fine-tuning the hybrid model and exploring dynamic adaptation to evolving user preferences would be essential for building a comprehensive and effective movie recommendation system.

It's possible that some people haven't seen any of the movies that are offered by OTT platforms. Therefore, suggestions for new users may take the shape of the most popular films or a genre/actor filter based on information from a survey completed during registration.

Other methods can also be utilized, such as content-based filtering, to benchmark, compare, and provide more suggestions. More accurate suggestions have been demonstrated by models like hybrid recommenders, which assist in combining collaborative and content-based filtering. They can assist with the cold start problem or sparse matrices.

8. Repository

https://github.com/Jayanthchidananda/CSP_571_Project

9. References

[1] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872

[2] Vilakone, P., Park, DS., Xinchang, K. et al. An Efficient movie recommendation algorithm based on improved k-clique. Hum. Cent. Comput. Inf. Sci. 8, 38 (2018).

[3] (2021, August 10). How to Calculate Cosine Similarity in R. R-bloggers. https://www.rbloggers.com/2021/08/how-to-calculate-cosine-similarity-in-r/

[4] Prabhakaran, S. (2022, April 20). Cosine Similarity – Understanding the math and how it works (with python codes). Machine Learning Plus.
https://www.machinelearningplus.com/nlp/cosine-similarity/

[5] Javed, M. (2021, December 16). Using Cosine Similarity to Build a Movie Recommendation System.Medium.
https://towardsdatascience.com/using-cosine-similarity-tobuild-a-movie-recommendation-system-ae7f20842599