Hybrid Movie Recommendation System with Temporal Contextual Filtering

Karthik Ravi Kumar
1225910467
ASU
Tempe, Arizona
kraviku8@asu.edu

Naveena Raipole
1228052438
ASU
Tempe, Arizona
nraipole@asu.edu

A. *Abstract*

This research introduces a robust hybrid movie recommender system that combines diverse recommendation methodologies and datasets to enhance movie suggestions. The system implements content-based recommendation through K-Nearest Neighbors (KNN), user-user collaborative filtering using KNN, employs Singular Value Decomposition (SVD) for item-item collaborative filtering, and presents a hybrid approach combining KNN content-based and collaborative filtering techniques. Additionally, the system employs temporal contextual filtering to adapt recommendations based on recent user preferences, ensuring relevance over time. Leveraging a consolidated dataset from MovieLens and TMDb Movies, the content-based KNN utilizes movie attributes, while collaborative filtering methods exploit user-item interactions and item-item similarity. Evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) demonstrate the efficacy of the models in providing accurate and diverse movie recommendations, highlighting its potential real-world applicability for delivering personalized user experiences.

I. Introduction

A. *Background*

Recommender systems are pivotal in the digital landscape, using algorithms to predict user preferences for items based on historical data. These systems streamline user experiences by offering tailored suggestions across diverse online platforms, addressing the challenge of information overload.

B. *Problem Statement*

Users face challenges in finding personalized items amidst vast choices, leading to information overload and dissatisfaction. Generic recommendations often fail to cater to individual preferences, necessitating personalized suggestions.

Recommender systems revolutionize user experiences by predicting personalized content in an age of information overload. Leveraging machine learning and data mining techniques, these systems offer tailored suggestions, spanning from e-commerce to streaming platforms. They address the challenge of content discovery by analyzing user behavior and item attributes, driven by a wealth of research in information retrieval and AI. As data volumes surge, the role of recommender systems becomes pivotal in navigating vast datasets, employing various components to refine accuracy. This report presents a hybrid recommender system combining diverse techniques, dataset merging, and rigorous evaluation, showcasing its impact on movie recommendations through experimental insights.

C. *Existing Literature*

In 2017, B. R. Cami, H. Hassanpour and H. Mashayekhi [1] developed a content-based movie recommender system integrating temporal user preferences within a Dirichlet Process Mixture Model. By incorporating content attributes of rated movies, this approach constructs a user-centric framework for inferring preferences and delivering tailored movie recommendations. Experimental evaluations utilizing the MovieLens dataset showcased improved performance compared to existing movie recommender systems.

Jinbo Zhang et al.[2] introduced a model that initially tackled various similarity metrics, including Cosine-based Similarity, Adjusted Cosine Similarity, and Correlation-based Similarity. Following this, they devised an algorithm designed to enhance the accuracy of identifying similar items or neighbors. This optimized algorithm was developed to overcome the inherent limitations observed in conventional item-based collaborative filtering approaches.

Mas Brillianesa Faydhurrahman et al.[3] introduced a web-based movie recommendation system, emphasizing the challenges in enhancing recommended systems' effectiveness while balancing accuracy and computational efficiency. The system employs content-based filtering techniques adaptable to various movie characteristics, with a focus on genre preferences and utilizing cosine similarity and the KNN algorithm to achieve over 87

D. System Overview

The dataset includes TMdb movies dataset and Movielens dataset for comprehensive movie information and user ratings, respectively. After preprocessing this data into a more workable format, it is then used to train and test multiple movie recommender models which include content based and collaborative filtering recommender systems. Each of these models are then used to predict movies which are to be recommended to the user. After evaluating the performances of each model and predictions provided, a hybrid recommendation system combining content-based and collaborative filtering techniques was implemented. This hybrid model gives movie recommendations based on both the user historical data and movie attributes. The effectiveness of different similarity measures (cosine similarity, Pearson correlation) and the system's performance is then evaluated using metrics such as MAE and RMSE.

E. Data collection

The project utilized the TMDB Movie dataset, a comprehensive collection encompassing various attributes of movies. This dataset was chosen due to its rich content, providing essential features such as genre, director, cast, and plot keywords, pivotal for content-based filtering techniques. In conjunction with the TMDB dataset, the project also integrated data sourced from Movielens. This dataset primarily contributed user ratings, which served as a fundamental component for user-user and item-item collaborative filtering methods.

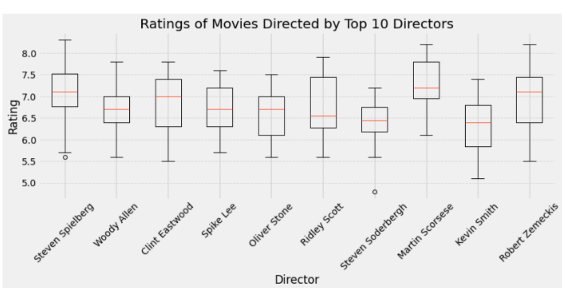

Fig. 1. Spread of ratings of each director's movie in MovieLens dataset

The datasets were merged to create a unified and enriched dataset. This fusion facilitated a holistic view of movie attributes, incorporating content-related information from TMDB Movies with user preferences obtained from Movielens. By combining these datasets, the system aimed to leverage the strengths of both, allowing for a more comprehensive analysis and recommendation mechanism.

From the TMDB dataset the Genre, Director, Cast, Keywords were utilized, these features were crucial for content-based filtering, enabling the system to recommend movies based on similarities in these attributes.The user ratings feature from Movielens data formed the basis for collaborative filtering techniques, facilitating recommendations by identifying similarities between users or items based on these ratings.

The rationale behind employing a merged dataset was to exploit the strengths of both content-based and collaborative filtering approaches. This strategy aimed to overcome limitations inherent in singular approaches by combining them, thereby enhancing the accuracy and diversity of movie recommendations provided by the system.The integration of these datasets allowed for a more robust and comprehensive recommendation system, catering to diverse user preferences while leveraging both movie attributes and user behavior.

F. Components of the ML system

The components of the movie recommender system include Data Collection and Integration, where data is acquired from two distinct sources, namely the TMDB movies dataset and MovieLens dataset, and the datasets are merged to create a comprehensive source of movie-related information. The Preprocessing step involves handling missing values, cleaning the dataset, merging relevant information, and potentially encoding categorical variables [4]. Feature Engineering focuses on extracting or selecting relevant features from the merged dataset for use in recommendation models, including attributes such as movie genres, cast, directors, actors, plot keywords, and ratings. Model Selection involves choosing specific algorithms for different purposes, such as K-Nearest Neighbors (KNN) for content-based recommendation, Singular Value Decomposition (SVD), and KNN for collaborative filtering. The Model Training step entails training the selected models on the prepared dataset, allowing them to learn patterns and relationships within the data. Model Evaluation employs metrics like Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to assess the predictive performance of the models in accurately predicting user preferences. Finally, Model Combination integrates predictions from content-based and collaborative filtering models, creating a hybrid recommendation approach to mitigate limitations inherent in each method.

II. Important Definitions

A. Prediction target

Movie Ratings/Predicted Ratings: Predicting the rating a user might give to a movie they haven't rated yet.

This involves estimating the likelihood of a user liking or disliking a specific movie.
Recommendations: Providing a list of the top N movies that a user is most likely to enjoy or engage with based on their historical preferences or behavior.
User ID: Identifiers for individual users in the system.
User Behavior: This refers to the historical movie ratings of the user.
User Profiles: Aggregated features summarizing a user's preferences or behavior.
Movie ID: Unique identifiers for each movie in the database.
Movie Attributes: The feature of a movie which are used to train models to predict similiar movies such as Genres, release year, director, cast, plot keywords.
Movie Ratings: Ratings given by users who have watched the movie.
Root Mean Square Error (RMSE): It is a commonly used metric to assess the accuracy of a predictive model by measuring the difference between predicted and observed values. It quantifies the average magnitude of errors or residuals between predicted and actual values.
Mean Absolute Error (MAE): Mean Absolute Error (MAE) is a metric used to determine the average magnitude of errors between predicted and actual values in a dataset. It measures the average absolute differences between predicted and observed outcomes.
RMSE and MAE are given by the Formulae:

$$\text{RMSE} = \sqrt{\frac{\sum (y_i - y_p)^2}{n}}$$

$$\text{MAE} = \frac{|(y_i - y_p)|}{n}$$

y_i = actual value
y_p = predicted value
n = number of observations/rows

Fig. 2. RMSE and MAE Formula

Cosine Similarity: It is a measure used to determine the similarity between two non-zero vectors in a given space. It calculates the cosine of the angle between these vectors, providing a measure of their orientation or similarity, disregarding their magnitudes.

It is given by:

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

Where:

- $A \cdot B$ denotes the dot product of vectors A and B.
- $\|A\|$ and $\|B\|$ represent the magnitudes (or norms) of vectors A and B, respectively.

Fig. 3. Cosine Similiarity

Pearson Correlation Coefficient: It is a statistical measure that evaluates the linear relationship between two continuous variables. It assesses how much one variable changes concerning another variable.

It is given by:

$$r = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2 \sum (Y - \bar{Y})^2}}$$

Where:

- $\bar{X}$ and $\bar{Y}$ denote the means of variables X and Y, respectively.
- The numerator represents the covariance between X and Y.
- The denominator represents the product of the standard deviations of X and Y.

Fig. 4. Pearson Correlation Coefficient

B. *Problem Statement*

Given the extensive range of available movies and the diverse nature of user preferences, the challenge at hand is to craft an effective movie recommender system capable of providing accurate and personalized movie suggestions that evolve with users' changing tastes over time. The main aim objective is to create a robust hybrid recommender system that combines various recommendation methodologies, thereby enhancing the precision and diversity of movie suggestions. To tailor recommendations according to user preferences, a combination of content-based filtering and collaborative filtering approach is employed [5][6]. Additionally, temporal contextual filtering is implemented to ensure the system adapts its recommendations based on recent user preferences, ensuring continued relevance and timeliness. The evaluation of the system's performance and effectiveness is conducted using metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). One of the challenges to overcome is the Cold Start Problem, where the focus is on delivering precise recommendations for new users who haven't given enough ratings or provided adequate data. This involves developing strategies to cater to users with limited history in the system. Another constraint is the Sparsity of Data, which involves handling situations where there is not enough interaction data between users and items, especially when there are limited ratings for movies. This sparsity poses a hurdle to the accuracy of recommendations, and addressing this issue is essential for improving the system's effectiveness.

III. Overview of proposed approach and system

In **Content-based filtering** , the movie recommendation system adeptly combines data from TMDb and MovieLens datasets, utilizing diverse movie metadata such as genre, director, cast, and plot keywords sourced from TMDb. This information enables the system to identify movies that closely align with similar thematic and cinematic attributes based on these features. Through this process, the system can identify movies that share analogous genres, directors, cast members, and plot keywords, ensuring recommendations that reflect the content's inherent features and characteristics. This analysis enhances the precision of recommended movies by emphasizing content with highly comparable attributes.

The **Collaborative Filtering** component of the recommendation system uses user ratings to personalize recommendations. It employs a KNN-based approach to find users with similar movie tastes, organizing them based on shared movie preferences. Using metrics like cosine similarity and Pearson correlation coefficients from these shared ratings, it gauges user likeness and computes weighted average scores for movie recommendations. Subsequently, the system suggests the top 10 unrated movies to the input user, assessing accuracy through metrics like RMSE and MAE. Comparative analysis between KNN methods shows that Pearson correlation-based KNN outperforms in offering more precise movie recommendations based on user preferences. Another Collaborative Filtering method explored here is Matrix factorization using Singular Value Decomposition of a matrix in which the users are the rows and movies are the columns. The numbers in the matrix are the ratings users give to each movies. We are using SVD to predict similiar movies using the latent interaction factors between movies and users.

The **Hybrid Recommendation System** addresses the limitations found in content-based and collaborative filtering by combining their strengths. Content-based filtering excels in suggesting items similar to user preferences but sometimes lacks in offering diverse recommendations. Conversely, collaborative filtering predicts preferences based on user behavior but encounters difficulties in handling new user or item introductions. In this hybrid approach, both techniques are integrated to improve recommendation accuracy by simultaneously considering user preferences and item attributes. This fusion aims to enrich recommendations by employing content-based filtering to resolve the cold-start problem and leveraging collaborative filtering for increased diversity. The fusion of these methodologies results in tailored and all-encompassing recommendations that bridge the gaps inherent in individual techniques.

Additionally, the content-based system implemented within the hybrid recommender incorporates temporal context. It operates by utilizing a subset of movies highly rated by the input user in the most recent two years. Leveraging the KNN content-based filtering methodology, this system identifies movies that most closely resemble the attributes of this subset. This approach ensures that the recommendations are aligned with the user's recent preferences, thereby enhancing the system's ability to provide more relevant and up-to-date suggestions.

IV. Technical details of proposed approaches/system

A. Feature extraction

Feature extraction for Content-Based Filtering using K-Nearest Neighbors relied on the extensive attributes obtained from the TMDB Movie dataset, encompassing genre, director, cast, and plot keywords essential for content-based filtering. Integrating data from Movielens supplemented this with crucial user ratings pivotal for collaborative filtering methods. The selected features are One-Hot Encoded, converting categorical attributes into binary vectors to represent movie characteristics effectively.

This transformation is crucial for categorical features,

```
0    [1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
1    [1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
2    [1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
3    [1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
4    [1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
Name: genres_bin, dtype: object
```

Fig. 5. One-Hot Encoding

as storing them in a list format is suboptimal for the recommendation system's functionality. The utilization of One-Hot Encoding allocated distinct columns for each genre, director, cast, and plot keywords, assigning values of 1 or 0 to signify a movie's presence within a specific feature category. Subsequently, leveraging these one-hot encoded vectors, the project computed the cosine distance between movies by summing the cosine distances of each feature, using this metric to gauge the similarity between movies based on their respective feature representations.

Collaborative Filtering with KNN utilized Ratings data from the Movielens dataset. Collaborative Filtering with SVD relied on user-item interaction matrix.

B. Predictive Modeling

Content-Based Filtering using K-Nearest Neighbors:

The system provides personalized movie suggestions using content-based filtering. Utilizing movie features like genre, director, cast, and plot keywords from TMDb, it employs one-hot encoding on these categorical attributes. Subsequently, cosine distances are calculated between each vector, followed by the summation of these distances to determine overall cosine distances between movies. Consequently, the system recommends the top 10 movies exhibiting the least cosine distance from the subset of highly rated movies by the input user. This methodology allows for the identification of movies closely resembling the thematic and cinematic preferences of the user's recent viewing patterns.

Collaborative Filtering using K-Nearest Neighbors: The system leverages user ratings on movies to offer tailored recommendations. Initially the system identifies users most similar to the input user, by finding a subset of users who share common rated movies with the input user and arranges them in the decreasing order of the number of movies watched in common. Subsequently, it computes both the cosine similarity and Pearson correlation coefficient between the input user and this subset, utilizing ratings from the movies they share. These similarity measures serve as indices to gauge the likeness between the input user and others, acting as weights to compute the weighted average recommendation scores for all movies. The system then suggests the top 10 movies that the input user has yet to watch. Moreover, it calculates the root mean square error (RMSE) and mean absolute error (MAE) of predicted ratings against the actual ratings of movies rated by the input user to assess recommendation accuracy. Additionally, a comparative analysis between cosine similarity-based k-nearest neighbors (KNN) and Pearson correlation-based KNN highlights the superior performance of the latter in generating more precise movie recommendations based on user preferences.

Collaborative Filtering using Singular Value Decomposition:
Matrix Factorization is a mathematical operation applied to matrices, particularly effective in collaborative filtering. It enables the identification of latent features underlying user-movie interactions. We are taking movies as rows and users as columns. As we know that not all users watch all the movies, resulting in numerous missing values. We are filling those missing observations with 0s as we are going to perform linear algebra operations A utility matrix is constructed from the dataframe and normalized across movies (items) with which we will find similarity. The final step involves transforming the dataframe values into a scipy sparse matrix for optimal computational efficiency.

SVD reduces dimensionality by capturing essential latent features that can approximate all values within the utility matrix. These latent factors correspond to item characteristics, such as music, genre etc. Through the reduction of the utility matrix A's dimensionality, SVD extracts these latent factors, mapping each user and item to an r-dimensional latent space. This mapping provides a representation of the relationships between users and items or movies.

We must also consider the fact that Matrix factorization works on the simple inner product of the item and user feature vectors and may fall short in capturing

```
array([[-0.02574607, -0.04109006, -0.01891121, ...,  0.        ,
         0.        ,  0.        ],
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       ...,
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.        ,  0.        , -0.01891121, ...,  0.        ,
         0.        ,  0.        ]])
```

Fig. 6. Utility Matrix

and representing the intricate relations within users and items.

The implemented **Hybrid Recommender System** combines the outcomes of content-based and collaborative filtering methods—specifically, the Content-based KNN and Pearson Correlation-based Collaborative KNN. This fusion results in a curated list of the top 10 personalized recommendations for users. By combining these diverse approaches, the system significantly enhances its capability to propose movies that closely align with individual preferences, delivering more pertinent and accurate suggestions to users.

The Hybrid Recommender System incorporates temporal context within the content-based recommendation mechanism. It optimizes personalized movie suggestions by filtering a subset of highly rated movies from the user's activity within the recent two years. Leveraging this subset, the system utilizes KNN content-based filtering to identify the most similar movies to this subset. By integrating temporal context, the system tailors its recommendations to align closely with the user's recent viewing patterns. This emphasis significantly boosts the precision and relevance of the movie suggestions, ensuring that the recommendations better reflect the user's present preferences and interests.

Furthermore, this hybridization strategy represents a promising approach to improve recommendation systems by addressing the limitations inherent in each method. It provides users with a more polished and comprehensive selection of movie suggestions tailored to their unique preferences.

V. Experiments

A. Data description

The recommendation system integrates two main datasets: TMDB Movies dataset and MovieLens dataset. TMDB provides extensive movie metadata, including genres, directors, cast, and plot keywords, crucial for content-based filtering. MovieLens, on the other hand, contributes user ratings essential for collaborative filtering techniques.

The TMDB dataset comprises comprehensive details on movies, encompassing genre labels, director names, cast information, and plot keywords. This dataset, containing over [Specify Number] movies, serves as the primary source for content-based recommendations.

In contrast, the MovieLens dataset captures user-movie interactions through user-provided ratings. This dataset records user ratings over time, offering valuable insights into user preferences and aiding collaborative filtering methods.

Data preprocessing steps involved ensuring data integrity by handling missing values, maintaining feature consistency, and removing duplicate entries. Relevant features like genre, director, cast, and plot keywords were extracted from the TMDB dataset, while normalization and encoding techniques were applied to facilitate effective model training.

The Input to the system is a user ID and output is the list of top 10 recommended movies
Example:

Fig. 7. Recommender System Input

```
Movies recommended by hybrid recommendation system:
Sense and Sensibility (1995)
Into the Wild (2007)
Nixon (1995)
Topsy-Turvy (1999)
Casino (1995)
Tarnation (2003)
Four Rooms (1995)
Safe (1995)
Get Shorty (1995)
There Will Be Blood (2007)
```

Fig. 8. Recommender System Output

B. Evaluation Metrics

The evaluation process of the recommendation system involved the utilization of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as primary evaluation metrics. These metrics were employed to assess the performance of various algorithms including KNN (content-based) and KNN user-user collaborative (Pearson correlation and cosine similarity-based) and Singular Value Decomposition (SVD).

By utilizing RMSE and MAE across different algorithms, the evaluation aimed to discern the effectiveness of each algorithm in accurately predicting user ratings for movies. Lower RMSE and MAE values indicated higher accuracy in predictions, aiding in the selection and optimization of algorithms for delivering precise and personalized movie recommendations to users.

C. Comparision

We are using Root Mean Square Error (RMSE) and Mean Square Error (MAE) to compare the predictions accuracy of different models.

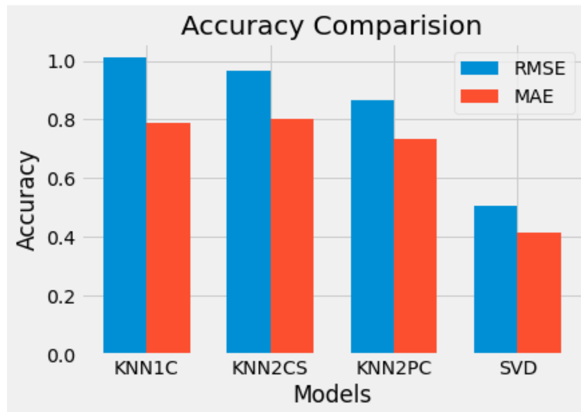

Fig. 9. RMSE and MAE Comparision

Here is the comparision, The evaluation metrics, RMSE and MAE, indicated that collaborative filtering models outperformed the KNN content-based filtering approach. Among the collaborative filtering methods, Singular Value Decomposition (SVD) demonstrated superior performance compared to KNN-based collaborative filtering. SVD's strength lies in its ability to extract latent features and reduce the dimensionality of the utility matrix effectively, enabling it to capture intricate user-item interactions and generate more accurate predictions.

Moreover, within the KNN collaborative filtering methods, the model utilizing Pearson correlation-based similarity outperformed the one based on cosine similarity. This outcome was in line with expectations due to Pearson correlation's ability to account for mean-centered ratings and handle varying rating scales more effectively than cosine similarity. Pearson correlation considers the rating patterns of users more comprehensively, leading to more accurate and refined recommendations.

The observed performances indicate that each model has its strengths and weaknesses. SVD, a collaborative filtering method, excels in capturing latent features and providing accurate predictions but may face challenges in handling sparse data or cold-start problems for new users or items. Conversely, KNN-based methods are more straightforward and intuitive, yet they might struggle with scalability and the curse of dimensionality.

While collaborative filtering models, particularly SVD, excel in capturing latent features and enhancing recommendation accuracy, the KNN content-based approach offers distinct advantages. KNN content-based filtering demonstrates superiority in scenarios where user-item interactions are sparse or when the system lacks sufficient user ratings. Unlike collaborative filtering, which heavily relies on historical user behavior, KNN content-based methods can recommend items solely based on their inherent features, making them more resilient to the cold-start problem for new users or less-rated items. This flexibility allows the system to provide relevant suggestions even when user interaction data is limited.

D. Case Studies

We have chosen the KNN content filtering model and KNN collaborative model as the baseline for comparision. The hybrid recommendation model developed in this project presents significant improvements over the KNN content-based and collaborative recommender systems. The content-based filtering model relies solely on movie attributes such as genre, director, cast and plot keywords, limiting its ability to offer diverse recommendations and accurate recommendations. In contrast, the hybrid model intelligently combines the strengths of both content-based and collaborative filtering techniques to provide more refined and accurate movie suggestions.

Collaborative filtering may suffer from the ”cold start” problem, where it's challenging to make accurate recommendations for new users or items. Hybrid systems can mitigate this by incorporating other information. One of the notable strengths of the hybrid model presented here lies in its incorporation of temporal context. Unlike the content-based filtering model that overlooks the timing of user interactions, the hybrid system places emphasis on recent user behavior within the most recent two years. This consideration of temporal context significantly enhances the system's recommendation accuracy, aligning the suggestions closely with the user's current viewing patterns. By prioritizing recent user preferences, the hybrid model ensures a more tailored and relevant selection of movies, thereby enhancing the overall user experience.

In conclusion, the hybrid recommendation system outperforms the both the individual content-based model and collaborative filtering techniques by combining them and integrating temporal context. This integration not only enhances the accuracy of movie recommendations but also ensures a more tailored and comprehensive movie selection, ultimately delivering a superior user experience.

VI. Conclusion

A combination of context-based filtering and collaborative filtering is employed in a hybrid approach to implement this scheme. This approach enhances the overall efficiency of the system by addressing the limitations of each method alone. Techniques such as similarity analysis, and classification, temporal context are employed to enhance recommendations, thereby reducing Mean Absolute error (MAE)and Root Mean Squared error(RMSE) also boosting precision and accuracy, offering the user personalized movie recommendations.

VII. References

B. R. Cami, H. Hassanpour and H. Mashayekhi, ”A content-based movie recommender system based on temporal user preferences,” 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS), Shahrood, Iran, 2017, pp. 121-125, doi: 10.1109/ICSPIS.2017.8311601.

Jinbo Zhang, Zhiqing Lin, Bo Xiao and Chuang Zhang, ”An optimized item-based collaborative filtering recommendation algorithm”, 2009 IEEE International Conference on Network Infrastructure and Digital Content, 2009.

M. K. Delimayanti et al., ”Web-Based Movie Recommendation System using Content-Based Filtering and KNN Algorithm,” 2022 9th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, Indonesia, 2022, pp. 314-318, doi: 10.1109/ICITACEE55701.2022.9923974.

S. Ghosh, “A comprehensive guide to data preprocessing,” neptune.ai, 22-Aug-2023. [Online]. Available: https://neptune.ai/blog/data-preprocessing-guide.

S. Jayalakshmi, N. Ganesh, R. Čep, and J. Senthil Murugan, “Movie Recommender Systems: Concepts, methods, challenges, and future directions,” Sensors (Basel, Switzerland), 29-Jun-2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9269752/.

P. Karthikeyan and C. Tejasvini, ”Review of Movie Recommendation System,” 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 1538-1543, doi: 10.1109/ICACCS54159.2022.9785014.

VIII. related Work

- M. Saadati, S. Shihab, and M. S. Rahman, Movie Recommender Systems: Implementation and Performance Evaluation, Sep. 2019.

- Thakker, U., Patel, R. & Shah, M. A comprehensive analysis on movie recommendation system employing collaborative filtering. Multimed Tools Appl 80, 28647–28672 ,April. 2021.

- J. F. Mohammad and S. Urolagin, ”Movie Recommender System Using Content-based and Collaborative Filtering,” 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET), Arad, Romania, 2022,
- C. S. M. Wu, D. Garg and U. Bhandary, ”Movie Recommendation System Using Collaborative Filtering,” 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2018,

IX. Code Link

https://github.com/karthikravikumar3/CSE572-HybridMovieRecommender