Filmview: A Review Paper on Movie Recommendation Systems

PRIYANSHU MODI[1], ATUL KUMAR[2], BHASKAR KAPOOR[3]
[1, 2, 3] *Maharaja Agrasen Institute of Technology*

Abstract- The proliferation of streaming platforms has led to a vast array of movie options, making it increasingly difficult for users to discover relevant Content. To address this challenge, recommendation systems have emerged as valuable tools for suggesting movies based on user preferences. We discuss the impact of temporal dynamics and social influence in improving recommendation accuracy and effectiveness. Moreover, we emphasize the importance of incorporating explanations to enhance user understanding and satisfaction. Through an examination of evaluation metrics, we assess the performance of these systems. Overall, this review contributes to the knowledge base, providing insights into the strengths, limitations, and future directions of Movie Recommendation Systems.

I. INTRODUCTION

With the increasing amount of Content available on streaming platforms, it has become more challenging for users to find what they want to watch. To solve this problem, Movie Recommendation Systems are a type of personalized Recommendation System that uses a combination of machine learning algorithms and user data to provide movie suggestions.

In this review paper, we will explore the different types of Movie Recommendation Systems, their strengths and weaknesses, and the factors that affect their performance. We will also examine the various evaluation metrics used to measure the efficacy of these systems.

II. LITERATURE REVIEW

The study by Mahesh et al. [1] presents a review of the various techniques used in Movie Rec. Systems. It explores Collaborative filtering, Content-based filtering, Hybrid approaches, and deep learning methods. Various similarity measures are examined. The widespread use of recommendation systems by companies like Facebook, LinkedIn, Pandora, Netflix, and Amazon is highlighted. The review provides a brief overview of different techniques and methods, offering valuable insights for further research in recommendation systems. Jiang et al. [2]'s study addresses scalability and practical usage feedback in Movie Recommendation Systems. It proposes a high-efficient recommendation algorithm based on user clustering. The method achieves comparable performance to traditional CF schemes with reduced time complexity. A Movie Recommendation System, is constructed for evaluation.

The creation of a Movie Recommender System employing tools like K Means Clustering and K Nearest Neighbour Algorithms is the subject of research by Rishabh et al. [3]. MovieLens is the dataset utilised, and Python is utilised to run the system. We offer a number of machine learning ideas, tools, and methods, such as KNN, K-Means Clustering, Collaborative Filtering, and Content-Based Filtering. The suggested system's implementation details, including its architecture, process flow, and pseudocode, are given. With a best RMSE value of 1.081648, the results demonstrate that the suggested system performs better than the state-of-the-art methods. Choudhury et al. [4]'s Recommender systems (RS) address the information overload problem, particularly in movie recommendations. Comparisons are made between the four recommendation models BPNN, SVD, DNN, and DNN with Trust. The DNN with trust model demonstrates the highest accuracy of 83% and a low MSE value of 0.74, making it an effective choice for movie recommendations.

A content-based movie recommendation system using a variety of factors is suggested by Sahu et al [5]. Based on RS results, film ratings, and voting data, a CNN deep learning model forecasts movie popularity.

The study surpasses benchmark models in accuracy, achieving 96.8%. Behera et al.'s paper [6] offers a collaborative filtering method for movie recommendations that takes temporal effects into account. With improvements of 1.35% and 1.28% on the ML-100K and 1M datasets, respectively, analysis of the Movielens dataset reveals that it outperforms leading models. Partitional Weighted co-clustering for movie recommendation is the main topic of Airen et al.'s [7] research. The objective is to optimize user and movie neighborhoods by adjusting row and column cluster parameters. Experimental results on a movie database demonstrate the proposed method's ability to provide more accurate personalized recommendations, outperforming existing methods by 7.91%.

Gupta et al. [8]'s study incorporates K-NN algorithms and Collaborative filtering, utilizing cosine similarity. The approach effectively combines the strengths of both methods, mitigating the limitations of Content-based filtering. The use of cosine similarity provides accuracy which is comparable to Euclidean Distance. Tahmasebi et al. [9]'s study proposes a deep autoencoder network-based hybrid social recommender system with collaborative filtering, content-based filtering, and user social impact from Twitter. The findings of the evaluation show that it is more accurate and efficient than cutting-edge techniques. Pecune et al. [10]'s study focuses on the importance of providing explanations in Movie Recommendation Systems. A conversational agent was created using a human-centered design methodology and it explains its suggestions in a manner similar to how a human would. The agent's architecture included the computational model of explanations, which was then tested in a user experiment. The findings show that, regardless of the quality of the recommendations, social explanations improve how well the system and interactions are received.

A brand-new graph-based model that takes into account user similarities, demographic data, and location is proposed by Darban et al. [11]. The use of Autoencoder feature extraction improves the reliability of recommendations and solves the cold-start issue. The efficiency of the suggested approach is demonstrated by experimental findings on the dataset. Kalyan Kumar et al. [12]'s work aims to develop a neural network model for accurate recommendations using the MovieLens dataset. The data manipulation process and model implementation are carried out in Python. The evaluation metric used is the Hit-Ratio, with the model achieving an 87% Hit-Ratio. Fiagbe [13]'s study develops a model using the MovieLens rating dataset, employing matrix factorization to predict movie ratings for unwatched films. The model then recommends movies with the highest predicted ratings to users. The evaluation showcases the system's accuracy, with low values of root mean squared error (RMSE) obtained.

III. METHODOLOGY

Machine learning algorithms are often the foundation of movie recommendation systems, which analyse user behaviour and data to produce tailored recommendations. Collaborative filtering and content-based filtering are the two methods utilised for movie recommendation systems most frequently.

- Collaborative Filtering:

Collaborative filtering is a technique that uses the similarities and differences in user behavior to make recommendations. It can be done through two ways:

a. User-Based Collaborative Filtering: This method pairs people with similar likes and suggests products they've enjoyed. A distance metric like Pearson Correlation or Cosine Similarity is used to determine how similar two users are to one another. The Pearson Correlation formula is as follows:

$$r = \frac{\Sigma(x_i - \underline{x})(y_i - \underline{y})}{\sqrt{\Sigma(x_i - \underline{x})^2 \Sigma(y_i - \underline{y})^2}}$$

correlation coefficient, or r.
xi = values of a sample's x-variable
x = the average of the x-variable's values
yi = values of a sample's y-variable
y = the average of the y-variable's values

b. Item-Based Collaborative Filtering: This method suggests products that are comparable to ones that a consumer has previously enjoyed. A distance metric like Pearson Correlation or Cosine Similarity is used to determine how similar two objects are. The formula for Cosine Similarity is:

$$cos(\Theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Where A_i and B_i are the i^{th} components of the vectors A and B respectively

Content-Based Filtering:

Using this method, objects are recommended based on their characteristics. It is used to make recommendations for products that are comparable to past favourites of a user. A distance metric, such as the Euclidean distance or the cosine similarity, is used to determine how similar two items are. The formula for Euclidean Distance is:

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Euclidean n-space's p and q are two points.
Starting from the space's origin (the starting point), qi and pi are Euclidean vectors.

Hybrid Filtering:

Hybrid filtering is a technique that combines both Collaborative and Content-based filtering to make recommendations. It is used to overcome the limitations of each technique and provide more accurate recommendations. Using a weighted average of their recommendations, collaborative and content-based filtering are integrated in this method. The weighted average formula is as follows:

$$W = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i}$$

Weigted Average is W
There are n terms to be averaged
The weights added to the x values are represented by w_i
The data values to be averaged are represeted by X_i

Matrix Factorization:

Matrix factorization is a technique used to reduce the dimensionality of large data sets. It is used to generate recommendations based on the similarities and differences in user behavior. In this approach, the user-item matrix is decomposed into two or more matrices using singular value decomposition (SVD) or other matrix factorization techniques. The formula for SVD is:

$$M = U\Sigma N$$

M stands for the initial matrix that needs to be broken down. The eigenvectors of the matrix MMt are contained in the left singular matrix U. The matrix has singular values and is diagonal. The eigenvectors of the matrix MtM make up V, the right singular matrix. For applications like movie recommendation systems, this breakdown offers insights on the data's structure.

Evaluation and Comparison of Recommendation Models:

To evaluate and compare various recommendation models, several metrics and techniques are used, including accuracy metrics, ranking metrics, diversity metrics, and coverage metrics.

Accuracy Metrics:

Accuracy metrics are used to measure the degree to which a recommendation model accurately predicts user preferences. The most commonly used accuracy metrics are:

a. Mean Absolute Error (MAE): This calculates the typical discrepancy between an item's actual rating and anticipated rating for a certain user.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

b. Root Mean Square Error (RMSE): The difference between the actual rating and the projected rating is measured by the standard deviation. The formula for RMSD is:

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}{N}}$$

Root-mean-square deviation (RMSD)
I = the variable i N = number of data points that are not missing
real observations (x_i) time sequence
predicted time series = $\hat{x}_i$

c. Precision: This measures the proportion of relevant items recommended to the user.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

d. Recall: This measures the proportion of relevant items recommended out of the total relevant items.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

IV. MODEL ARCHITECTURE

The steps below are often taken into consideration while creating a movie recommendation system:

1. Data Collection:

The initial stage is to gather information on user preferences and movie choices. This information can be found on a variety of websites, including those that offer movie reviews, social media sites, and streaming services with user reviews.

2. Data Preprocessing:

The data, after collection, should be pre-processed to remove any duplicates, missing values, or irrelevant information. The data is also transformed into a suitable format for the recommendation model.

3. Feature Extraction:

The next step is to extract features from the preprocessed data that can be used to make recommendations. These features may include movie genres, actors, directors, ratings, and user preferences.

4. Model Selection:

There are several alternative recommendation methods that can be used, including matrix factorization, collaborative filtering, content-based filtering, and hybrid filtering. The kind of data that is available and the particular requirements of the application determine the best model to use.

5. Model Training:

The selected model should be trained on the preprocessed data to learn the underlying patterns and relationships between movies and user preferences.

6. Model Evaluation:

To assess the trained model's suitability for producing recommendations, a variety of measures are used, including accuracy, ranking, diversity, coverage, and novelty.

7. Model Deployment:

Deploying the trained model in a production environment is the final step, where it can make recommendations to users based on their preferences.

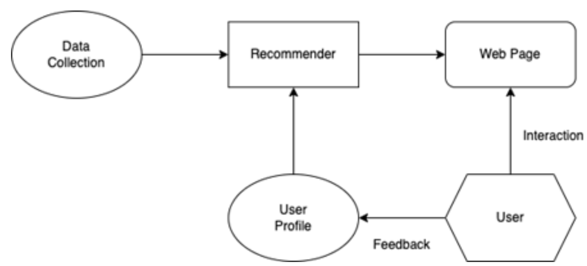

Fig. The recommendation system basic architecture

V. DISCUSSION

The research presented in this review paper emphasises the crucial part that recommendation systems play in the film business. Many methods have been investigated and contrasted, including collaborative filtering, content-based filtering, and hybrid approaches. It is clear that combining different strategies, such as applying deep learning algorithms, can improve the efficacy and functionality of movie recommendation systems.

The review also emphasizes the importance of considering dynamic factors in recommendation systems. Traditional methods often fail to capture the temporal and evolving nature of user-item interactions. However, incorporating temporal effects in Collaborative filtering models has shown promising results in improving recommendation accuracy.

Furthermore, the integration of social influence and explanations in recommendation systems has demonstrated positive outcomes. The use of social characteristics and behaviors can provide valuable insights into users' preferences and enhance the quality of recommendations. Additionally, incorporating explanations in the recommendation process improves users' perception and understanding of the system's decisions.

CONCLUSION

In conclusion, Movie Recommendation Systems have become indispensable tools for users to navigate the vast amount of available Content. This review paper has provided an overview of various techniques and approaches employed in Movie Recommendation Systems. It is clear that combining various techniques can produce recommendations that are more precise and tailored, such as collaborative filtering, content-based filtering, and deep learning algorithms.

The study also highlights the importance of addressing dynamic factors, such as temporal effects and evolving user preferences, in recommendation systems. By incorporating these factors, the performance and relevance of recommendations can be significantly improved.

Moreover, the inclusion of social influence and explanations in recommendation systems has shown promising results. The use of social characteristics and behaviors, along with providing explanations for recommendations, enhances user satisfaction and perception of the system.

Overall, this review paper contributes to the existing knowledge by presenting an in-depth analysis of different techniques and approaches used in Movie Recommendation Systems. It provides insights into the challenges faced by these systems and suggests future directions for research. Incorporating contextual data and user comments, for example, could improve movie recommendation systems' effectiveness even further.

REFERENCES

[1] Goyani, Mahesh; Chaurasiya, Neha. "A Review of Movie Recommendation System: Limitations, Survey and Challenges". ELCVIA : Electronic Letters on Computer Vision and Image Analysis, Vol. 19 No. 3 (2020), p. 18-37 DOI 10.5565/rev/elcvia.1232 https://ddd.uab.cat/record/232276

[2] J. Zhang, Y. Wang, Z. Yuan and Q. Jin, "Personalized real-time Movie Recommendation System: Practical prototype and evaluation," in Tsinghua Science and Technology, vol. 25, no. 2, pp. 180-191, April 2020, doi: 10.26599/TST.2018.9010118.

[3] R. Ahuja, A. Solanki and A. Nayyar, "Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 263-268, doi: 10.1109/CONFLUENCE.2019.8776969.

[4] Choudhury, S.S., Mohanty, S.N. & Jagadev, A.K. Multimodal trust based recommender system with machine learning approaches for movie recommendation. Int. j. inf. tecnol. 13, 475–482 (2021). https://doi.org/10.1007/s41870-020-00553-2

[5] S. Sahu, R. Kumar, M. S. Pathan, J. Shafi, Y. Kumar and M. F. Ijaz, "Movie Popularity and Target Audience Prediction Using the Content-Based Recommender System," in IEEE Access, vol. 10, pp. 42044-42060, 2022, doi: 10.1109/ACCESS.2022.3168161.

[6] Gopal Behera, Neeta Nain, Collaborative Filtering with Temporal Features for Movie Recommendation System, Procedia Computer Science, Volume 218, 2023, Pages 1366-1373, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2023.01.115.

[7] Sonu Airen, Jitendra Agrawal, Movie Recommender System Using Parameter Tuning of User and Movie Neighbourhood via Co-Clustering, Procedia Computer Science, Volume 218, 2023, Pages 1176-1183, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2023.01.096.

[8] M. Gupta, A. Thakkar, Aashish, V. Gupta and D. P. S. Rathore, "Movie Recommender System Using Collaborative Filtering," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 415-420, doi: 10.1109/ICESC48915.2020.9155879.

[9] Tahmasebi, H., Ravanmehr, R. & Mohamadrezaei, R. Social movie recommender

system based on deep autoencoder network using Twitter data. Neural Comput & Applic 33, 1607–1623 (2021). https://doi.org/10.1007/s00521-020-05085-1

[10] Florian Pecune, Shruti Murali, Vivian Tsai, Yoichi Matsuyama, and Justine Cassell. 2019. A Model of Social Explanations for a Conversational Movie Recommendation System. In Proceedings of the 7th International Conference on Human-Agent Interaction (HAI '19). Association for Computing Machinery, New York, NY, USA, 135–143. https://doi.org/10.1145/3349537.3351899

[11] Zahra Zamanzadeh Darban, Mohammad Hadi Valipour, GHRS: Graph-based Hybrid recommendation system with application to movie recommendation, Expert Systems with Applications, Volume 200, 2022, 116850, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2022.116850.

[12] Jena, K.K., Bhoi, S.K., Mallick, C. et al. Neural model based Collaborative filtering for Movie Recommendation System. Int. j. inf. tecnol. 14, 2067–2077 (2022). https://doi.org/10.1007/s41870-022-00858-4

[13] Fiagbe, Roland, "Movie Recommender System Using Matrix Factorization" (2023). Data Science and Data Mining. 4. https://stars.library.ucf.edu/data-science-mining/4