Journal of Informatics Education and Research
ISSN: 1526-4726
Vol 4 Issue 3 (2024)

A Novel Approach to Movie Recommendation Using Weighted Collaborative Filtering with Activity and Rating Variability Analysis

[1,2] J Somasekar, [3] Bhuvan Unhelkar, [4] Siva Shankar S, [5] Prasun Chakrabarti

[1]Post Doctoral Researcher, Information Systems and Decision Sciences, University of South Florida (USF), 8350 N. Tamiami Trail Sarasota, Florida, USA.

[2]Department of CSE, JAIN (Deemed-to-be-University), Bengaluru, Karnataka, 562 112, India.

[3]Professor, Information Systems and Decision Sciences, University of South Florida, 8350 N. Tamiami Trail Sarasota, Florida, USA.

[4]Department of CSE, KG Reddy College of Engineering and Technology, Moinabad,Telangana, India – 501 504.

[5]Department of CSE, Sir Padampat Singhania University, Udaipur, Rajasthan, India – 313 601.

[1] jsomasekar@gmail.com , [3]bunhelkar@usf.edu , [4]drsivashankars@gmail.com , [5]drprasun.cse@gmail.com

Abstract— This paper introduces a novel approach to movie recommendation systems through a weighted collaborative filtering technique that integrates user activity and rating variability analysis. Traditional methods, such as Pearson correlation, fail to account for variations in user activity and rating behaviours. To address this gap, we propose a weighted Pearson correlation that adjusts similarity scores based on both the number of ratings and the variability in those ratings between users. This adjustment improves the precision and robustness of the recommendations. The similarity calculation is further refined by incorporating weights that reflect user activity and rating consistency, which are essential in mitigating biases introduced by users with differing rating patterns. Furthermore, the methodology incorporates a weighted adjustment formula to improve the prediction of user ratings for unrated items. Experimental results show that the proposed algorithm surpasses traditional methods, delivering better prediction accuracy and higher recommendation quality. The findings emphasize the effectiveness of incorporating activity and rating variability into collaborative filtering, resulting in a more reliable and robust recommendation system for real-world applications.

Keywords—Recommendation Systems, Collaborative Filtering, Pearson Correlation, Similarity Scores, Utility matrix, Cosine similarity.

I. INTRODUCTION

Recommendation systems have become an integral part of numerous applications, especially in domains like e-commerce, streaming services, and online content platforms. Netflix reports that 80% of its viewed content comes from recommendations, saving $1 billion annually through customer retention. YouTube drives 70% of its watch time via personalized suggestions, adapting dynamically based on user behavior. Amazon attributes 35% of its sales to its item-based collaborative filtering, delivering real-time suggestions. Social media platforms like TikTok excel with engagement-driven feeds, while the Play Store uses machine learning to suggest apps based on reviews and history. Traditional collaborative filtering techniques, such as those leveraging Pearson correlation (PC), focus on identifying similarities between users based on their ratings. However, these methods often overlook key factors such as user activity levels and rating variability, which can significantly affect the accuracy and reliability of recommendations [1]. Among the various recommendation techniques, CF has emerged as one of the most popular methods, primarily because of its simplicity and effectiveness. However, traditional collaborative filtering algorithms, such as Pearson correlation, often face challenges when accounting for variability in user activity and rating behaviours. These challenges can lead to inaccurate or biased recommendations, especially when dealing with users who exhibit diverse rating patterns or engage with the system at different levels of activity Addressing these limitations is crucial for creating robust systems capable of delivering high-quality, personalized recommendations.

This study introduces a novel method for movie recommendation that enhances collaborative filtering through a weighted Pearson correlation approach. By incorporating user activity and rating variability, the proposed methodology adjusts similarity scores to account for differences in user behavior. These adjustments mitigate biases arising from inconsistent rating patterns and varying levels of user engagement, addressing critical gaps in traditional techniques [2].

The proposed approach integrates preprocessing steps, such as handling missing data and normalizing ratings, to improve data consistency and reduce noise. These steps are essential in ensuring that the similarity calculations and predictions remain accurate and meaningful. The weighted Pearson correlation is further refined by introducing activity-based and variability-based weights, which dynamically adjust the influence of users on similarity measures[3]. The prediction model employs a weighted adjustment formula that incorporates these similarity scores to refine predictions, ensuring that user-specific factors are accurately captured.

Experimental results demonstrate the superiority of the proposed algorithm compared to traditional methods, such as KNN, SVD, and basic CF [4]. With improved prediction accuracy and enhanced recommendation quality, the methodology underscores the importance of integrating activity and rating variability into collaborative filtering (CF) [5]. This innovation provides a robust framework for handling real-world challenges, such as sparse datasets and diverse user behaviors, thereby setting a new benchmark for personalized movie recommendation systems [6]. Inthe following sections, we will cover related works, proposed methodology, evaluation measures, experimental results withcomparisons, and conclusion.

II. RELATED WORKS

Recommendation systems have revolutionized the way users discover content across various domains, particularly in entertainment. At the heart of many such systems is Collaborative Filtering (CF), a method that leverages shared preferences and interactions among users to predict and recommend items. Over time, CF has evolved with enhancements like weighting schemes, hybrid models, and latent factor techniques, making it adaptable to address challenges like data sparsity, cold starts, and dynamic user preferences. Below, we explore its key advancements and applications.

Collaborative filtering (CF) is a key technique in recommendation systems. First proposed by Resnick et al. (1994), CF focuses on user-item interactions, serving as the foundation for user-based and item-based recommendation methods [7]. The approach assumes that users with comparable preferences in the past are likely to share similar tastes in the future, enabling more accurate predictions of movie ratings. Weighted collaborative filtering builds on basic CF by adjusting similarity measures. Sarwar et al. (2001) demonstrated that incorporating weights based on similarity improves prediction accuracy by prioritizing more similar users or items [8]. This method helps address issues like sparsity, improving the reliability of recommendations in sparse datasets. Activity-based filtering incorporates user engagement data such as clicks and watch time, which provides a richer context than traditional ratings alone. Oard & Kim (2000) explored how combining activity data with CF enhances recommendation accuracy, particularly when explicit ratings are sparse [9]. This model captures more dynamic user preferences based on real-time interactions. Rating variability, or the spread in user ratings for the same item, influences recommendation accuracy. Konstan et al. (1997) highlighted the importance of understanding rating variability to improve CF models, suggesting that variation in user opinions should be considered in prediction algorithms [10].

Latent factor models, particularly matrix factorization, have significantly advanced CF. Koren et al. introduced matrix factorization techniques, which uncover hidden features that explain user preferences [11]. These models improve predictions by capturing the latent structure in sparse user-item matrices. Hybrid models combine multiple recommendation strategies to enhance performance. Burke (2002) reviewed various hybridization techniques, noting how combining CF with content-based filtering and knowledge-based methods helps overcome challenges like data sparsity and cold starts [12]. This approach leads to more accurate and personalized recommendations.

Pazzani and Billsus (2007) highlighted the effectiveness of content-based filtering, which generates recommendations by analyzing item attributes such as genre or director. By integrating content-based methods with CF, the cold-start problem is mitigated, as new items or users can be recommended based on their features [13]. Incorporating temporal aspects in CF can improve recommendations by accounting for the evolving nature of user preferences. Zhang et al. (2007) discussed how user preferences change over time, and how temporal dynamics can be incorporated into CF models to enhance prediction accuracy [14]. Adomavicius & Tuzhilin (2011) explored how these systems improve recommendation relevance by considering contextual factors in addition to user ratings and activity [15]. Deshpande & Karypis (2004) focused on scalability in CF systems, highlighting methods to handle large datasets while maintaining high accuracy [16]. Their work emphasizes the need for efficient algorithms to scale with growing user bases and item catalogs. The following describes the proposed methodology followed by experimental results.

III. PROPOSED SYSTEM

The user-item matrix, also known as the utility matrix, is a key concept in recommendation systems, particularly for movie recommendations. It organizes user preferences for various items (such as movies) in a structured way. This two-dimensional matrix has rows representing users and columns representing items, with each cell containing a rating or preference score assigned by a user to an item [17][18]. If a user hasn't rated an item, the corresponding cell is left empty. The matrix serves two main purposes: predicting missing ratings by estimating how users might rate unrated items and generating recommendations by analyzing user preferences and identifying similar users or items. Ratings are typically on a scale from 1 to 5. Consider a set of **n**-users, denoted as $\{\mathbf{U_1}, \mathbf{U_2}, \mathbf{U_3}, \ldots, \mathbf{U_n}\}$, and a set of **m**-movies, represented by $\{\mathbf{M_1}, \mathbf{M_2}, \mathbf{M_3}, \ldots, \mathbf{M_m}\}$. Let $\mathbf{r_{ij}}$ represent the rating assigned by user $\mathbf{U_i}$ to movie $\mathbf{M_j}$, where **i** corresponds to a specific user and **j** to a particular movie. These ratings form the entries of a utility matrix, which encapsulates the relationship between users and movies in a matrix of size $\mathbf{m} \times \mathbf{n}$. The utility matrix as shown in Table-1.

Table-1: Utility matrix

	M_1	M_2	M_3	…	M_m
U_1	r_{11}	r_{12}	r_{13}	…	r_{1m}
U_2	r_{21}	r_{22}	r_{23}	…	r_{2m}
U_3	r_{31}	r_{32}	r_{33}	…	r_{3m}
……..					
U_n	r_{n1}	r_{n2}	r_{n3}	…	r_{nm}

Creating a novel user-based collaborative filtering approach involves innovating on existing methods or integrating new concepts. We proposed a method that introduces a weighted similarity calculation incorporating both user activity level and rating variability [19][20]. This method aims to improve prediction accuracy and robustness. Traditional Pearson correlation does not account for the differences in user activity levels and rating variability. We propose a weighted Pearson correlation that incorporates these factors.

Data Preprocessing is a crucial step in the movie recommendation system to ensure the quality and consistency of the data before applying the weighted collaborative filtering algorithm [21]. The preprocessing process involves handling missing data, normalizing ratings, and ensuring the integrity of the user-item matrix. Handling Missing Data in movie dataset is very significant to improve the accuracy. In real-world datasets, many entries in the user-item matrix will be missing, as not every user rate every movie. To deal with this, mean imputation can be used, where missing ratings for a user are replaced with the average rating of that user. Alternatively, more sophisticated techniques like k-nearest neighbors (KNN) imputation or matrix factorization methods can be applied for a more accurate estimate of missing values. These

Journal of Informatics Education and Research
ISSN: 1526-4726
Vol 4 Issue 3 (2024)

methods help ensure that the matrix remains usable for calculating similarities and making predictions. In moralizing ratings, Users may have different rating behaviors, with some users rating movies more generously or strictly than others. To address this, it is important to normalize ratings. This can be done by subtracting a user's average rating from each of their ratings, resulting in a normalized score centered around zero. This normalization ensures that differences in individual rating scales do not bias the similarity calculation. After normalization, similarities between users are more accurately based on their relative preferences rather than their absolute rating values. Together, these preprocessing steps enhance the performance of the recommendation algorithm by ensuring a cleaner and more standardized dataset. For two users u and v, the Pearson correlation coefficient is given by:

$$r_{uv} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \overline{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \overline{r}_v)^2}} \quad (1)$$

where I_{uv} is the set of items rated by both users u and v, r_{ui} is the rating given by user u to item i, and $\overline{r}_u$ is the average rating of user u. The covariance between the ratings of users users u and v, weighted by w_{uv} is defined by,

$$C_{uv} = \sum_{i \in I_{uv}} w_{uv} (r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v) \quad (2)$$

In Weighted Pearson Correlation, we introduced the weights based on factors like the reliability of ratings (e.g., numberof ratings provided by the user) or the importance of specific ratings. The weighted formula could look like this:

$$sim_{uw}(u, v) = \frac{C_{uv}}{\sqrt{\sum_{i \in I_{uv}} w_{uv} (r_{ui} - \overline{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} w_{uv} (r_{vi} - \overline{r}_v)^2}} \quad (3)$$

Weighted Pearson Correlation helps to account for biases and discrepancies in ratings, leading to potentially more accurate and personalized recommendations [22]. Weighted Pearson Correlation improves recommendation accuracy by adjusting for biases in user ratings, such as the number of ratings or variability. Unlike standard Pearson Correlation, it accounts for differences in rating behaviour and item popularity, resulting in more reliable similarity scores and personalized recommendations. w_{uv} is the weight associated with the user similarity, computed as,

$$\mathrm{w_{uv}} = 1/[1 + \mathrm{AD(u, v)} + \mathrm{RVD(u, v)}] \quad (4)$$

Where, Activity Difference between users u and v ie., AD(u,v) , which represents the difference in the number of ratings given by the two users u and v. The Rating Variability Difference between users u and v i.e., RVD(u,v), which represents the difference in the variance of ratings given by the two users. The AD and RVD are used to adjust the similarity score between two users based on their activity level and the variability in their ratings. These adjustments aim to account for the fact that users with different levels of activity or rating variability may not be directly comparable. The denominator of w_{uv} ensures that higher differnces lead to lower similarity weights. The following mathematical equations are used to calculate $AD(u, v)$ and $RVD(u, v)$, respectively.

$$AD(u, v) = \left| \frac{N_u - N_v}{N_u + N_v} \right| \quad (5)$$

Where N_u and N_v are the number of ratings provided by users u and v.

$$RVD(u, v) = \left| \frac{\sigma_u^2 - \sigma_v^2}{\sigma_u^2 + \sigma_{v\,u}^2} \right| \quad (6)$$

Where σ_u^2 and σ_v^2 are the variances of ratings provided by users u and v.

In our proposed movie recommendation methodology, user activity and rating variability play a crucial role in enhancing the accuracy of the similarity calculation. User activity refers to the number of ratings a user provides. It is measured by counting the total number of movies rated by each user. A higher number of ratings typically indicates a more reliable profile, while users with fewer ratings may provide less stable or consistent preferences. To account for this, the similarity calculation is adjusted by introducing a weight based on the activity level. Users with a greater number of ratings receive higher weights in the similarity calculation, as their preferences are considered more representative of their actual tastes. Specifically, the weight is inversely proportional to the difference in the number of ratings provided by two users, ensuring that highly active users contribute more significantly to the similarity measure. Rating variability captures how much a user's ratings fluctuate across different movies. It is measured by the variance in the user's ratings. A user with high variability may rate some movies very highly while rating others very low, whereas a user with low variability tendsto give more consistent ratings. To incorporate this, the similarity score between two users is adjusted by their rating variances. Users with lower variability are weighted more heavily, as their preferences are more stable, leading to more reliable predictions. Integrating these two factors in to the weighted Pearson correlation is to adjust the similarity score between users. These factors are combined into a weighted Pearson correlation to improve the precision of movie recommendations.

The Weighted Adjustment is used in collaborative filtering to refine the prediction of a user's rating by incorporating the influence of other users who have rated the same item. By adjusting for the difference between a similar user's rating and their average rating $(r_{vi} - \bar{r}_v)$, the formula captures how much a similar user's rating deviates from their baseline behaviour. This makes the adjustment by ensuring that users' biases do not distort the prediction.

Table-2: **The Proposed method Pseudocode**

Step 1: (Preprocessing)

```
user_item_matrix = impute_missing_data(user_item_matrix)
normalized_matrix = normalize_ratings(user_item_matrix)
```

$$normalized\_rating_{ui} = \frac{user\_item\_matrix_{ui} - \mu_u}{\sigma_u}$$

Step 2: (Calculate Similarities)

```
similarity_matrix = {}
for each pair of users (u, v):
activity_diff = calculate_activity_difference(u, v)
rating_variability_diff = calculate_rating_variability_difference(u, v)
weight_uv = 1 / (1 + activity_diff + rating_variability_diff)
similarity_matrix[(u, v)] = calculate_weighted_pearson(u, v, weight_uv)
```

Step 3: (Predict Ratings)

```
predicted_ratings = {}
for each user u and movie i:
    predicted_rating = predict_rating(u, i, similarity_matrix)
    predicted_ratings[(u, i)] = predicted_rating
```

Step 4: (Generate and Output Recommendation)

```
recommendations = {}
    for each user u:
        top_k_movies = recommend_top_k_movies(u, predicted_ratings)
          recommendations[u] = top_k_movies
              return recommendations.
```

The Weighted Adjustment (A_{ui}) Formula is as shown in Eqn. (7) in which the numerator computes the weighted difference between the actual rating) given by a similar user for the item and their average rating. This approach ensures that the adjustment made to the prediction is proportional to the typical rating behavior observed for each of the similar users being considered. The denominator in the formula plays a crucial role in normalizing this adjustment by taking into account the total sum of similarity values associated with the neighboring users. This normalization step is essential for preventing any single user's contribution from becoming disproportionately large and dominating the overall prediction.

Furthermore, the design of this formula guarantees that the influence exerted by each similar user on the final prediction remains directly proportional to the degree of similarity they share with the target user, denoted as user u. By doing so, it maintains a balanced and fair incorporation of information from the most relevant users. This ensures the adjustment is relative to the typical rating behavior of each similar user and the denominator normalizes the adjustment bythe total similarity values of the neighbors, preventing the contribution of any individual user from dominating the prediction and this formula ensures that the influence of similar users is proportional to their similarity to the target user u.

$$A_{ui} = \frac{\sum_{v \in N_u} sim_{uw}(u,v).(r_{vi} - \bar{r}_v)}{\sum_{v \in N_u} |sim_{uw}(u,v)|} \tag{7}$$

The Personalized Rating Prediction as shown in Equation (7) is to predict a user's rating $\hat{r}_{ui}$ that the target user u would give to the item i. It adds the target user's average rating $\bar{r}$ to the weighted adjustment (A_{ui}) calculated in the Eqn. (7).

$$\hat{r}_{ui} = \bar{r}_u + A_{ui} \tag{8}$$

Where, N_u is the set of users similar to u based on the weighted similarity and $\bar{r}_u$ is the average rating of user u. $\hat{r}_{ui}$ is the predicted rating for user u on item i, $sim_{uw}(u, v)$ is the similarity weight between user u. and user v, r_{vi} is the rating of item i by user v and $|sim_{uw}(u, v)|$ is the absolute value of the similarity weight between user u and user v. To Generate Recommendations, for each user u, rank the movies based on the predicted ratings $\hat{r}_{ui}$ and recommend the top 'k' movies with the highest predicted ratings. Hence, Recommended movies for each user based on weighted collaborative filtering using activity and rating variability. The proposed novel Algorithm Pseudocode is as shown in Table-2.

IV. RESULTS AND DISCUSSION

The effectiveness of recommendation systems is evaluated through a thorough experimental validation process that assesses the performance of various collaborative filtering techniques and their enhancements. These enhancements include methods like weighted filtering, activity-based models, and hybrid approaches that aim to improve the accuracy and relevance of recommendations. The evaluation process is crucial to determine how well these algorithms perform in predicting user preferences and providing personalized recommendations.

Key metrics used in this evaluation include precision, recall, and RMSE (Root Mean Squared Error), which are essential in quantifying the prediction accuracy and user satisfaction. Precision measures the proportion of relevant recommendations among all recommended items, while recall assesses the ability of the system to identify all relevant items for a user. The F1-score, which balances precision and recall, is often used as a combined metric to gauge the overall recommendation quality. RMSE evaluates the difference between predicted ratings and actual user ratings, providing insights into the system's prediction accuracy. These metrics help to objectively compare different recommendation methods, identify their strengths and weaknesses, and determine which approach best enhances the recommendation system's performance, ultimately contributing to higher user satisfaction and better user experience [23] [24].

A. Accuracy Measures:

A.1 Prediction Accuracy Measures

These metrics assess the accuracy of predicted ratings by comparing them to the actual ratings provided by users. The MAE indicates the overall prediction accuracy. A lower MAE reflects better performance.

$$MAE = \frac{1}{|T|} \sum \left| r_{ui} - \bar{r}_{ui} \right| \tag{9}$$

The Root Mean Squared Error (RMSE) Penalizes larger errors more heavily than MAE and useful when large deviations are more critical to the application.

$$RMSE = \sqrt{\frac{1}{|T|} \sum \left(r_{ui} - \bar{r}_{ui} \right)^2} \tag{10}$$

A.2 Classification Accuracy Measures

Precision, recall, and F1 score are metrics used to evaluate the performance of classification models, particularly in cases of imbalanced data. **Precision** measures the accuracy of positive predictions, indicating the proportion of correctly identified positive instances out of all instances predicted as positive. **Recall** (or sensitivity) measures the ability of the model to identify all actual positive instances, representing the proportion of true positives out of the total actual positives. The **F1 score** is the harmonic mean of precision and recall, providing a single metric that balances the trade-off between them. It is especially useful when the cost of false positives and false negatives is comparable, offering a holistic view of the model's effectiveness in handling both correctly identified and missed cases. These metrics measure how well the system suggests items that users end up rating highly. These measures Evaluate the ability to recommend relevant movies (high ratings) versus irrelevant ones. The Precision reflects the relevance of recommendations, while recall measures coverage [25]. The F1 score balances both. These measures are calculated by using the following mathematical equations.

$$Precision = \frac{TP}{TP+FP} \tag{11}$$

$$Recall = \frac{TP}{TP+FN} \tag{12}$$

$$F1 = \frac{2\,(Precision(u) \times Recall(u))}{Precision(u)+Recall(u)} \tag{13}$$

B. Results and Comparison

In this section, we provide a detailed analysis of the evaluation metrics used to assess the performance of the proposed recommendation algorithm and compare it with several traditional methods. The dataset employed for this evaluation was sourced from reputable public repositories, including the UCI ML Repository and GitHub. These datasets were selected due to their widely acknowledged use in the research community, ensuring that the results are relevant and reproducible.

The proposed algorithm demonstrates clear advantages over traditional recommendation techniques, such as KNN (K-Nearest Neighbors) and basic collaborative filtering, in terms of both prediction accuracy and recommendation quality. These improvements are primarily due to the integration of two crucial factors—user activity and rating variability—into the similarity calculation. Traditional methods often rely on simplistic similarity measures (e.g., Pearson correlation) that do not account for variations in user behavior, particularly in terms of how active users are in providing ratings and the diversity in their rating patterns. By addressing these aspects, the proposed method is able to offer more robust and reliable predictions, especially when dealing with users who have diverse rating behaviors and varying activity levels.

One of the significant innovations of the proposed method is the introduction of a weighted similarity measure that incorporates these two factors. The weighting scheme allows the algorithm to prioritize more reliable ratings based on users' activity levels and the consistency of their ratings, leading to improved prediction accuracy. This makes the model more effective in personalizing recommendations, as it can adjust its recommendations based on the user's interaction history and behavior. By considering both the frequency of a user's ratings and the variability in their preferences, the method is better suited to handling diverse user profiles and providing more accurate predictions, resulting in higher user satisfaction.

To assess the effectiveness of the proposed method, a comparative evaluation was conducted with five other recommendation algorithms: Matrix Factorization, KNN, Bayes Classifier, Cosine Similarity, and SVD (Singular Value Decomposition). These algorithms represent a broad spectrum of approaches commonly used in recommendation systems, ranging from neighborhood-based methods (KNN, Cosine Similarity) to matrix factorization techniques (SVD, Matrix Factorization), and classification-based methods (Bayes Classifier). The proposed algorithm outperformed all of these methods across various performance metrics, including prediction accuracy and recommendation relevance.

In terms of prediction accuracy, the proposed algorithm achieved the lowest MAE (Mean Absolute Error) of 0.78 and RMSE (Root Mean Squared Error) of 1.04. These values indicate that the proposed method produces predictions that are more accurate than the other algorithms. The MAE and RMSE metrics assess how closely the predicted ratings match the actual user ratings, with lower values reflecting higher accuracy. The superior performance of the proposed algorithm in these metrics suggests that it provides more reliable and precise predictions.

Method	MAE	RMSE	Precision	Recall	F1
Proposed	0.78	1.04	0.85	0.82	0.83
Matrix Factorization	0.81	1.09	0.81	0.79	0.80
KNN	0.92	1.24	0.76	0.72	0.74
Bayes Classifier	0.94	1.30	0.72	0.68	0.70
Cosine Similarity	0.90	1.15	0.74	0.70	0.72
SVD	0.85	1.16	0.82	0.79	0.80

Table-3: Comparison of the proposed method with existing approaches

Additionally, the proposed algorithm excelled in classification accuracy measures, such as Precision, Recall, and F1-Score. The Precision value of 0.85 indicates that the algorithm is highly effective at recommending items that users are likely to rate highly, while the Recall of 0.82 shows that the algorithm is capable of covering a broad range of relevant recommendations. The F1-Score, which balances Precision and Recall, was 0.83, further emphasizing the algorithm's ability to strike a good balance between relevance and completeness in its recommendations.

In comparison, Matrix Factorization and SVD produced competitive results, achieving F1-Scores of 0.80 each. However, these methods had slightly higher MAE and RMSE values, suggesting that while they performed well in some areas, they were not as precise in their predictions as the proposed method. KNN and Cosine Similarity, which are based on simpler neighborhood techniques, performed relatively poorly, with F1-Scores of 0.74 and 0.72, respectively. These lower scores reflect their limitations in handling complex user preferences and rating behavior, particularly when users have diverse or inconsistent rating patterns.

Furthermore, the Bayes Classifier ranked the lowest, with the highest MAE (0.94) and RMSE (1.30), indicating that it struggled to capture the intricate relationships within sparse datasets, a common issue for classification-based methods. While traditional algorithms like Matrix Factorization and SVD remain reliable, the proposed algorithm offers significant improvements in prediction accuracy and recommendation quality. By incorporating user activity and rating variability into the similarity measure, it better handles complex user behaviors, making it ideal for personalized movie recommendations. This highlights the algorithm's potential to enhance recommendation systems in real-world applications, providing a more tailored user experience and boosting user satisfaction.

V. CONCLUSION

In this work, we introduced a novel user-based collaborative filtering approach for movie recommendation systems, which integrates weighted similarity calculations to enhance prediction accuracy. By considering both user activity levels and rating variability, our proposed method improves upon traditional Pearson correlation-based models, which often fail to account for differences in user behavior. The weighted Pearson correlation, alongside a preprocessing phase to handle missing data and normalize ratings, allows for more accurate and personalized recommendations. Through rigorous evaluation using key metrics like MAE, RMSE, precision, recall, and F1-score, our method demonstrated superior performance compared to existing recommendation algorithms, such as KNN, Matrix Factorization, and SVD. The results showed a significant improvement in prediction accuracy and recommendation quality, validating the effectiveness of incorporating user activity and rating variability into the similarity calculation. The proposed algorithm outperformed others, particularly in handling diverse user behaviors and ensuring more robust, personalized suggestions. These findings highlight the potential of weighted collaborative filtering in addressing common challenges in recommendation systems, offering a more tailored user experience with higher accuracy and relevance in movie recommendations.

REFERENCES

[1] Breese, J. S., Heckerman, D., & Kadie, C.**,** Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence **,** 1998.

[2] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. , Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World Wide Web,2001.

[3] Adomavicius, G., & Tuzhilin, A. , Toward the next generation of recommender systems: A survey of the state-of-the-

art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749, 2005.

[4] Koren, Y. , Factorization meets the neighborhood: A multifaceted collaborative filtering model. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008.

[5] Lee, J., & Lee, K. (2014). Collaborative filtering with rating variability and user activity. Journal of Computer Science and Technology, 29(2), 369-384.

[6] Zhang, S., Yao, L., & Li, C. , Personalized recommendation with matrix factorization techniques. Proceedings of the 23rd International Conference on World Wide Web, 2014

[7] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW), 175-186.

[8] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World Wide Web (WWW), 285-295.

[9] Oard, D. W., & Kim, J. (2000). Implicit feedback for recommender systems. Proceedings of the ACM Workshop on Recommender Systems, 81-85.

[10] Konstan, J. A., Riedl, J., & Miller, B. (1997). Collaborative filtering: A powerful tool for building recommendations. IEEE Computer, 30(7), 88-90.

[11] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer Science and Engineering, 42(8), 30-37.

[12] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370.

[13] Pazzani, M., & Billsus, D. (2007). Content-based recommendation systems. In The Adaptive Web: Methods and Strategies of Web Personalization (pp. 325-341). Springer.

[14] Zhang, Y., Karypis, G., & Niu, F. (2007). A survey of collaborative filtering techniques. Proceedings of the International Conference on Data Mining, 297-301.

[15] Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems. AI Magazine, 32(3), 67-80.

[16] Deshpande, M., & Karypis, G. , Item-based top-N recommendation algorithms. ACM Transactions on Information Systems, 22(1), 143-177, 2004.

[17] Raza, S., Rahman, M., Kamawal, S., Toroghi, A., Raval, A., Navah, F., & Kazemeini, A. , A Comprehensive Review of Recommender Systems: Transitioning from Theory to Practice, 2024.

[18] Francesco Ricci, Lior Rokach, and Bracha Shapira.Recommender Systems: Techniques, Applications, and Challenges.In Recommender Systems Handbook: Third Edition, chapter Chapter 1, pages 1–35. Springer, New York, NY, 2022.

[19] Maryam Etemadi, Sepideh Bazzaz Abkenar, Ahmad Ahmadzadeh, Mostafa Haghi Kashani, Parvaneh Asghari, Mohammad Akbari, and Ebrahim Mahdipour.A systematic review of healthcare recommender systems: Open issues, challenges, and techniques, 2023.

[20] ACM Recommender Systems Conference.ACM RecSys Conferences, 2024.

[21] Mansoor Iqbal.Netflix Revenue and Usage Statistics (2024) - Business of Apps, 2024.

[22] Precision Reports.Worldwide Market Research Report, and Industry Analysis, 2024.

[23] Alvise De Biasio, Nicolò Navarin, and Dietmar Jannach.Economic recommender systems – a systematic review.Electronic Commerce Research and Applications, 63:101352, 1 2024.

[24] Marwa Sharaf, Ezz El Din Hemdan, Ayman El-Sayed, and Nirmeen A. El-Bahnasawy.A survey on recommendation systems for financial services.Multimedia Tools and Applications, 81(12), 2022.

[25] Robin De Croon, Leen Van Houdt, Nyi Nyi Htun, Gregor Štiglic, Vero Vanden Abeele, and Katrien Verbert.Health recommender systems: Systematic review, 2021.

[26] Yue Sun, Jia Zhou, Mengmeng Ji, Lusi Pei, and Zhiwen Wang.Development and Evaluation of Health Recommender Systems: Systematic Scoping Review and Evidence Mapping, 2023.

[27] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian.A Survey on Session-based Recommender Systems.ACM Computing Surveys, 54(7), 2022.