International Journal of Engineering and Management Research
e-ISSN: 2250-0758 | p-ISSN: 2394-6962
https://ijemr.vandanapublications.com

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.14

Extended Collaborative Filtering Recommendation System with Adaptive KNN and SVD

Sagedur Rahman
Student, Department of Management Science and Engineering, Chongqing University of Posts and Telecommunications, CHINA

Corresponding Author: sm.bd123789@gmail.com

Received: 11-07-2023 Revised: 30-07-2023 Accepted: 14-08-2023

ABSTARCT

In recent years, recommendation systems have gained significant importance due to the vast amount of digital content available on various online platforms. Collaborative filtering is a widely adopted approach in recommendation systems, leveraging user-item interactions to make personalized predictions. However, traditional collaborative filtering methods face challenges such as the cold-start problem and data sparsity. To address these issues, researchers have proposed advanced techniques, including Adaptive KNN-Based and SVD-Based Extended Collaborative Filtering. This paper provides a comprehensive review of these two recommendation systems, discussing their underlying principles, advantages, and limitations. Furthermore, we explore recent research advancements and real-world applications, providing insights into the potential future developments in this field.

Keywords-- Collaborative Filtering, KNN, SVD, Matrix Factorization, Recommendation System

I. INTRODUCTION

The rapid growth of digital content and the proliferation of e-commerce platforms have led to an overwhelming abundance of choices for users. In this information-rich era, recommendation systems have emerged as indispensable tools for enhancing user experience, engagement, and satisfaction. By leveraging user behavior and preferences, these systems offer personalized recommendations, there by easing the among various recommendation techniques, collaborative filtering has gained widespread popularity due to its effectiveness in delivering personalized recommendations. Collaborative filtering is based on the idea that users who have shown similar preferences in the past are likely to share similar preferences in the future. This user-centric approach has proven to be highly successful in various domains, including movie recommendations, music playlists, product recommendations, and more.

The simplicity and effectiveness of collaborative filtering make it a fundamental building block of many recommendation systems. By relying solely on historical user-item interactions, collaborative filtering does not require explicit knowledge about items' characteristics or users' demographics. Instead, it derives insights from the collective wisdom of the user community, enabling it to adapt to dynamic user preferences and evolving item popularity.

However, traditional collaborative filtering methods face challenges that can limit their performance in certain scenarios. One such challenge is the cold-start problem, which arises when new users or items enter the system without sufficient historical data to make accurate recommendations. In such cases, traditional collaborative filtering struggles to identify relevant users or items to form recommendations, leading to suboptimal results and potentially disappointing user experiences.

Data sparsity is another significant hurdle in collaborative filtering. As online platforms continue to grow, the user-item interaction matrix becomes increasingly sparse, with the majority of possible user-item pairs having no recorded interactions. This sparsity can hinder the system's ability to discover meaningful patterns and correlations, reducing the quality of recommendations. To address these limitations and improve the performance of collaborative filtering, researchers have developed advanced techniques such as Adaptive KNN-Based Collaborative Filtering and SVD-Based Extended Collaborative Filtering.

The significance of this paper lies in its exploration and comparison of these advanced collaborative filtering techniques. By understanding their underlying principles, advantages, and potential limitations, researchers and practitioners can make informed decisions about which method best suits their specific recommendation system needs. Furthermore, by showcasing real-world applications and evaluating performance in various contexts, this paper aims to contribute to the advancement of recommendation systems, enhancing user experiences across a broad range of online platforms.

II. RELATED WORK

A specific type of information filtering called recommendation systems is intended to combine material (movies, music, books, news, images, and web pages) that the user is likely to find interesting. A

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.14

recommendation system often enables users to compare their profiles to benchmark specifications in attempts to predict what they want. The explanation of recommendation systems that have the greatest usage is that of Robin Burke. A system that can direct the user to pertinent or helpful information throughout a huge data space or offer tailored recommendations [1]. Collaborative filtering is the foundation for the majority of current recommendation system research [2, 3]. The two primary categories of collaborative filtering are memory-based and model-based algorithms. Memory-based algorithms analyze the area around the target user to identify users with similar preferences and produce recommendations [3]. In order to determine how similar users or things are, these algorithms use a variety of similarity functions, such as the Pearson correlation coefficient (PCC), and then compute a weighted average of the ratings offered by nearby users in order to make predictions. The most current information is advantageous for memory-based algorithms, but analyzing several neighbors might be computationally expensive, especially for big user datasets. To solve this problem, model-based algorithms have been created [4, 5] that use data mining techniques to build models of user ratings and forecast user preferences.

A list of people who have stated their preferences for various goods typically makes up the repository of a recommendation system. As was previously indicated, a user's decision for an item is referred to as a view, and it is frequently represented as a triple (user, rating, and item). These perspectives can take on various guises. Additionally, most rating systems for recommendations use binary ratings (like/dislike) or scores on a range from 1 to 5. This is the score matrix, and it is made up of the triplets "user, element, and rating." The values in the matrix are discarded for the element-user pair for which the user did not provide an element score. A referral system's goals can be summed up in two categories. The algorithm must first predicted the value of the indicated notes by asking, given a user and an item, what would be the user's preference for that item. What ordered list (n elements) of recommendations may the system suggest as the second component? It's referred to as a Top-n list. The list of n recommendations is not always the list of n components with the most pertinent prediction values, it should be noted. As score prediction is not the only factor used to generate a list of recommendations, a recommendation algorithm may also include other factors, such as context [1], [6].

Collaborative Filtering

The fundamental concept underpinning collaborative filtering is to use user ratings of certain documents to suggest those same publications to other users without having to analyze the content of the documents [7]. Memory-based and model-based recommendations can both be used to build the CF approach [8]. The user-item correlation, however, is a need of the memory-based recommendation approach. The neighborhood-based techniques [9], including KNN [10] and the Pearson correlation coefficient (PCC) [11] are used in the memory-based approach. This technique calculates the user-item similarity and transforms preference knowledge into predictions [12], [13]. The disadvantage is that they need to have access to the complete dataset in order to use particular indexing strategies for prediction. As a result, the data scale is growing. The preference matrix is taken into account by the model-based recommendation approach, which [12] employs both online and offline stages to forecast the recommendation. The method forecasts user interest with goods in both the online and offline phases that the customization model analyzes. The model-based strategy is also used by the dimensionality reduction methodology [8]. Matrix factorization [15], artificial neural networks [16], and Bayesian networks are well-known model-based approaches [14]. The matrix factorization suggests that the latent relationship between users and services can be captured by SVD.

Adaptive KNN-Based CF

Adaptive KNN-Based Collaborative Filtering is a sophisticated extension of the classic K-nearest neighbors (KNN) algorithm, designed to address the limitations of traditional KNN and enhance recommendation accuracy [17]. While the original KNN algorithm recommends items based on the similarity of users or items, its performance heavily depends on the fixed 'k' value, which represents the number of neighbors considered for making recommendations [17]. Selecting the right 'k' value is critical since a small 'k' might lead to noisy recommendations, while a large 'k' can introduce undue influence from distant neighbors.

To overcome these challenges, Adaptive KNN incorporates dynamic neighborhood size adjustments. Instead of using a fixed 'k,' the algorithm determines the neighborhood size based on the local density of user-item interactions [18]. In regions with sparse interactions, a larger 'k' value is chosen to improve recommendation accuracy, while in denser regions, a smaller 'k' value is preferred to avoid undue influence from too many neighbors.

Algorithm 1 Adaptive KNN-Based Collaborative Filtering

Require: User-item rating matrix R, Target user u, Item i, Number of neighbors k

$N_u \leftarrow$ Find k-nearest neighbors of u based on similarity with other users
$S(i) \leftarrow$ Compute the similarity between i and other items
$R_{u,N_u} \leftarrow$ Ratings of user u on items in N_u
$r_i \leftarrow$ Predicted rating of user u on item i
$w_i \leftarrow$ Weight of item i based on its similarity with other items
$R_{u,N_u} \leftarrow R_{u,N_u} \times w_i$
$r_i \leftarrow \frac{\sum_{j \in N_u} R_{u,j}}{\sum_{j \in N_u} w_j}$
return r_i

By locating similar people and things and using their ratings to build suggestions, our algorithm creates tailored predictions. It is frequently employed in collaborative filtering systems to offer users precise and pertinent item choices. Additionally, Adaptive KNN integrates dynamic weighting strategies that give more weight to recent interactions [20]. This consideration of temporal dynamics accounts for the changing preferences of users over time. By placing more emphasis on recent interactions, the algorithm adapts to users' evolving interests and provides timely and relevant recommendations that align with users' current preferences.

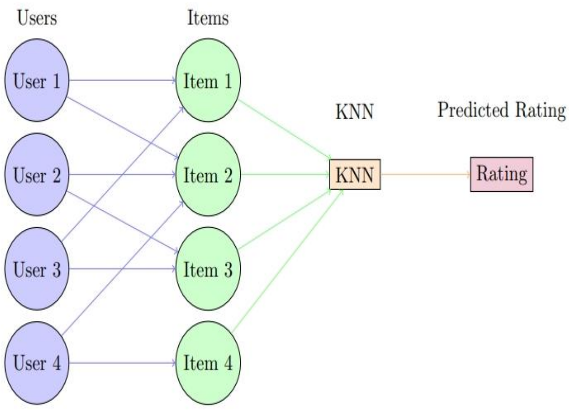

Figure 1: Architecture of KNN-Based CF

The architecture shows Fig1. How the K-nearest neighbor algorithm is used to locate comparable items, how user-item interactions are processed by the Adaptive KNN-Based Collaborative Filtering system, and how personalized recommendations are created in the form of anticipated ratings for unrated items. The adaptability of Adaptive KNN makes it a valuable choice for recommendation systems operating in dynamic environments where user preferences change frequently [19]. It not only overcomes the limitations of traditional KNN but also offers more robust and accurate recommendations in scenarios with data sparsity and cold-start challenges [21].

Singular Value Decomposition (SVD) in CF

Singular Value Decomposition (SVD) is a powerful matrix factorization technique widely utilized in collaborative filtering to capture latent factors that influence user-item interactions [22]. In collaborative filtering, the user-item interaction data is typically represented as a sparse matrix, where rows correspond to users, columns correspond to items, and the elements represent the user-item interactions (e.g., ratings, clicks, or purchases). SVD decomposes this user-item interaction matrix into three matrices: a user matrix, a singular value matrix, and an item matrix.

Algorithm 2 SVD in Collaborative Filtering

Require: User-Item matrix R with dimensions $m \times n$, and number of latent features k.

Calculate mean rating μ_u for each user and mean rating μ_i for each item.
Mean center the User-Item matrix R by subtracting the respective user and item means from each entry.
Perform Singular Value Decomposition (SVD) on the mean-centered matrix R:
    $R = U\Sigma V^T$, where U is an $m \times k$ matrix, Σ is a $k \times k$ diagonal matrix, and V^T is a $k \times n$ matrix.
Keep only the top k singular values and their corresponding singular vectors in U and V^T to reduce the dimensionality.
Compute the reconstructed matrix $\hat{R}$ using the truncated SVD:
    $\hat{R} = U_k \Sigma_k V_k^T$, where U_k is the first k columns of U, Σ_k is the top-left $k \times k$ submatrix of Σ, and V_k^T is the first k rows of V^T.
for each user-item pair (u, i) with missing rating **do**
    Predict the rating $\hat{r}_{ui}$ for user u and item i:
        $\hat{r}_{ui} = (\hat{R})_{ui}$
end for
return Predicted ratings $\hat{R}$ for all user-item pairs.

This approach explains how to use shortened SVD to predict missing ratings for user-item pairs with incomplete data and perform Singular Value Decomposition to lower the dimensionality of the User-Item matrix. The result is the anticipated ratings matrix *R*, which can be used to provide tailored suggestions or complete the blanks in the initial User-Item matrix.

The user matrix represents users in a lower-dimensional space, where each row corresponds to a user and the columns represent the latent factors. Similarly, the item matrix represents items in the same lower-dimensional space, with each row corresponding to an item and the columns representing the latent factors. The singular value matrix contains the singular values, which indicate the importance of each latent factor in capturing the variability of the original user-item interaction data.

By reducing the dimensionality of the user-item interaction matrix, SVD uncovers underlying patterns and relationships between users and items [23]. The latent factors capture hidden characteristics, such as user preferences or item attributes that contribute to user-item interactions. For example, in a movie recommendation system, the latent factors could represent movie genres, and the values in the user and item matrices would indicate the preferences of users and the characteristics of movies in each genre.

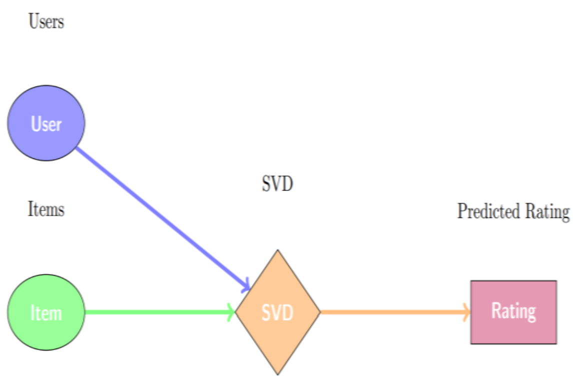

Figure 2: Architecture of CF with SVD

One of the key advantages of SVD-Based Collaborative Filtering is its ability to handle the cold-start problem, which occurs when new users or items have limited historical data. Traditional collaborative filtering methods struggle to make accurate recommendations for these new entities due to the lack of sufficient data. However, SVD can still generate meaningful recommendations for new users or items by leveraging the latent factors learned from existing interactions [24]. This makes SVD-Based Collaborative filtering particularly valuable in scenarios with sparse data or when dealing with new users or items.

Additionally, SVD allows for efficient and scalable computation, making it suitable for large-scale recommendation systems with a vast number of users and items [25]. The computational efficiency of SVD enables real-time or near-real-time recommendation generation, providing a seamless user experience on various online platforms.

$$R_{\text{mean}}(i,j) = R(i,j) - \text{mean}(\text{Row}_i(R)) - \text{mean}(\text{Column}_j(R)) + \text{mean}(\text{All}_R)$$

Compute the mean-centered matrix R mean by subtracting the mean rating from each element of R:

$$R_{\text{mean}} = U\Sigma V^T$$

Perform Singular Value Decomposition (SVD) on the mean-centered matrix Rmean. Where U is an $m \times k$ matrix containing the left singular vectors, Σ is a $k \times k$ diagonal matrix containing the singular values in descending order, and V is an $n \times k$ matrix containing the right singular vectors.

$$R_{\text{approx}} = U_p \Sigma_p V_p^T$$

Where Up is an $m \times p$ matrix containing the first p columns of U , Σp is a $p \times p$ diagonal matrix containing the top p singular values, and Vp is an $n \times p$ matrix containing the first p columns of V.

$$\hat{R}(u,i) = R_{\text{approx}}(u,i)$$

For each user-item pair (u, i) with missing rating (i.e., $R(u,i) = 0$):

Compute the predicted rating for item i for user u using the corresponding entry in the approximated matrix.

$$\hat{R}(u,i) = \hat{R}(u,i) + \text{mean}(\text{Row}_u(R)) + \text{mean}(\text{Column}_i(R)) - \text{mean}(\text{All}_R)$$

Optionally, add back the mean rating to obtain the final predicted rating.

Extended Collaborative Filtering

Extended Collaborative Filtering represents a significant advancement in recommendation systems by integrating additional information beyond traditional user-item interactions. Collaborative filtering, while effective, has inherent limitations, especially in scenarios with data sparsity and the cold-start problem. Extended Collaborative Filtering addresses these challenges by incorporating diverse external features, including user demographics, item attributes, and contextual data [29].

By augmenting the user-item interaction matrix with contextual information, Extended Collaborative Filtering gains a more comprehensive understanding of user preferences and item characteristics [28]. This enriched representation enables the system to generate more accurate and diverse recommendations, even for new users or items with limited historical data. For instance, in an e-commerce setting, user demographics such as age, gender, and location, along with item attributes such as brand, category, and price range, can be considered to make personalized and relevant recommendations.

The inclusion of external features allows Extended Collaborative Filtering to create a more holistic user-item interaction model [28]. This model considers multiple dimensions of user preferences and item characteristics, enhancing the accuracy and relevance of recommendations. Moreover, Extended Collaborative Filtering can better handle the cold-start problem, as it can make informed recommendations for new users or items based on their associated external features [27].

The flexibility of Extended Collaborative Filtering makes it applicable to various domains, including e-commerce, entertainment, healthcare, and more [26]. In the entertainment domain, the incorporation of contextual data such as movie genres, release dates, and user reviews can lead to improved movie recommendations. In healthcare, patient demographics, medical history, and treatment preferences can be integrated to offer personalized medical recommendations or treatment plans.

Extended Collaborative Filtering has been a subject of research in the field of recommender systems, with various studies focusing on its effectiveness in real-world applications [27]. It is an active area of research, with ongoing efforts to develop sophisticated algorithms that leverage diverse external features for enhanced recommendation accuracy.

III. METHODOLOGY

Research Framework

Data Collection: Gather user-item interaction data, such as ratings or feedback. Data Preprocessing: Handle missing values and normalize ratings for effective processing. KNN-based Collaborative Filtering: Predict user-item ratings based on the ratings of K most similar items the user has interacted with. SVD-based Collaborative Filtering: Decompose the user-item rating matrix to capture latent features and provide personalized recommendations. Recommendation Generation: Combine predictions from KNN and SVD models to generate top-N recommendations for each user. Evaluation: Split data into training and testing sets to assess model accuracy using evaluation metrics like RMSE and MAE. Performance Analysis: Compare model strengths, weaknesses, scalability, and efficiency for different recommendation scenarios. The research framework entails data preparation, collaborative filtering algorithms, suggestion generating, and an in-depth assessment to comprehend the effectiveness and value of each technique in creating efficient recommender systems.

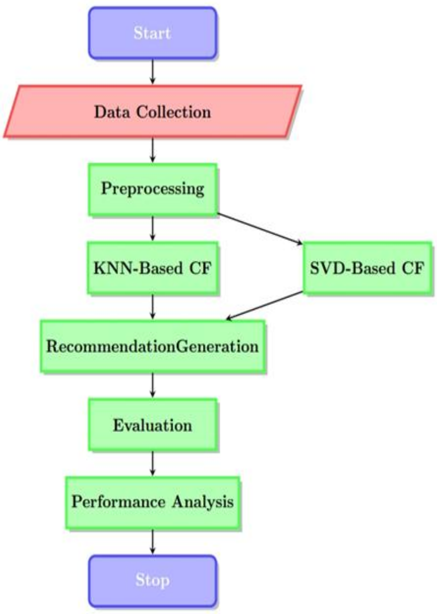

Figure 3: Research Framework

Dateset

In this section, we analyze the performance and efficacy of the suggestions made by the suggested model with the foundations. In the remaining subsections, we provide specifics on the datasets, factors, and results from the experiments. We leverage the MovieLens data set for collaborative filtering [30, 31]. MovieLens is a web-based research recommender system that first emerged in the fall of 1997. MovieLens data sets were gathered by the GroupLens Research Project at the University of Minnesota. Numerous visitors visit MovieLens every week to rate films and get suggestions. More than 45000 individuals have now left comments on 6600 different movies on the website. Each user had at least 20 ratings, and we randomly picked enough people to collect 100,000 reviews from 1,000 users on 1680 films, together with basic demographic data about the users. The evaluations are given on a numerical five-point scale, with 1 and 2 denoting unfavorable evaluations, 4 and 5 denoting favorable evaluations, and 3 denoting ambivalence.

In order figure out the movie ratings of the users, we will integrate the outcomes of the k closest neighbor algorithm and the model-based SVD algorithm using a hybrid collaborative filtering approach [32], [33]. The benefit of collaborative filtering algorithms is that item feature knowledge has become necessary.

IV. RESULT AND EVALUATION

We conducted research to see how well the suggested KNN and SVD model performed at recommending movies when compared to other parameters. In order to accomplish this, we ran tests on the MovieLens dataset and used a number of assessment criteria, such as MAE and RSME. We wanted to show the efficacy and superiority of our strategy by contrasting the performance of our suggested model with the standards utilizing these criteria.

The Mean Absolute Error (MAE) is calculated as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Where n is the total number of user-movie pairs, yi represents the actual rating, and $\hat{}yi$ represents the predicted rating.

Another one, The Root Mean Squared Error (RMSE) is calculated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Root mean square error (RMSE) emphasizes greater absolute error; the lower the RMSE, the higher the accuracy of the recommendations. As we can the **Figure 4** that rating frequency of top 10 movies 0-600 on the other hand movie ID 1- 300 which calculated their rating 1.0-5.0 as well as Figure 5 demand actual and predicted ratings of KNN which has been 0-7000 and rating 1.0-5.0 where as predicted ratings 4.0 has been increased than actual ratings.

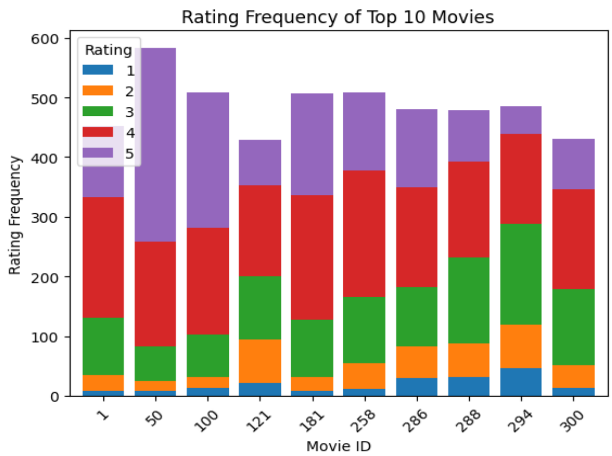

Figure 4: Rating frequency of top 10 movies

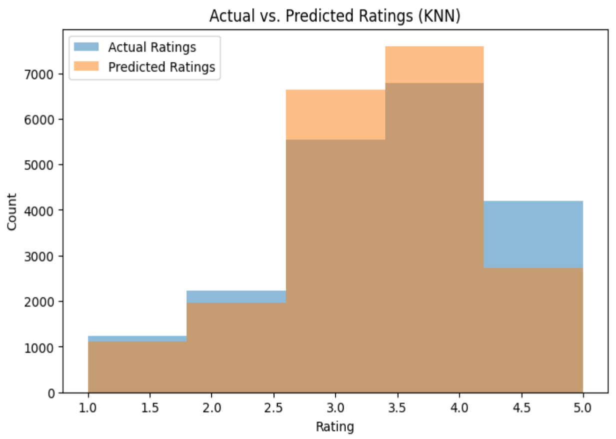

Figure 5: KNN actual vs. predicted ratings

As we can see the figure 6 Precision: Measures the accuracy of positive predictions. For class 1, it's 22%.Recall: Measures the ability to identify positive instances. For class 1, it's 20%. F1-score: A balance of precision and recall. For class 1, its 0.21.The model performs relatively better for classes 3 and 4 but struggles with class 5, having low recall. Improvements may be needed, such as feature engineering or parameter tuning. With an accuracy of 0.31895, the KNN model was able to accurately predict the class labels for about 31.895% of the dataset's instances.

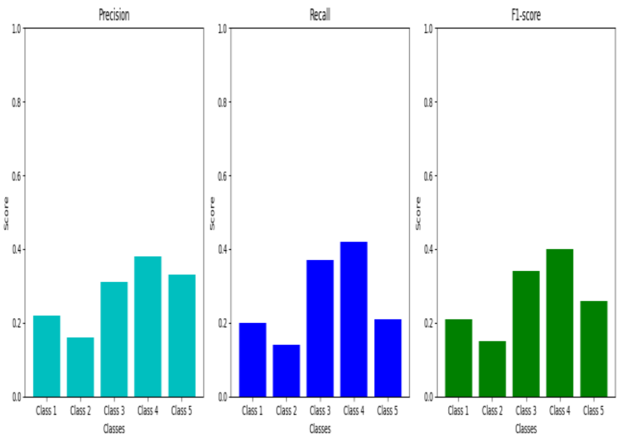

Figure 6: Precision, Recall and F1 score

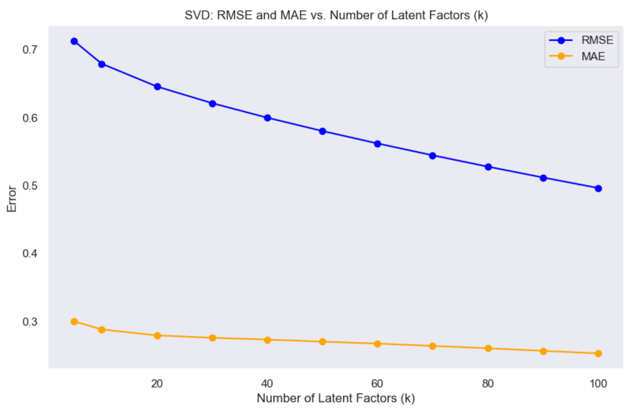

Figure 7: SVD RMSE and MAE

Root Mean Squared Error (RMSE): RMSE calculates the SVD model's standard prediction error. It calculates the discrepancies between the actual ratings in the dataset and the projected ratings. A lower reported RMSE value (0.580) shows that the SVD model makes predictions that are, on average, more accurate and closer to the actual ratings. And Using the Mean Absolute Error (MAE) formula, it is possible to determine the average size of the discrepancies between the projected and actual ratings. It offers an alternative viewpoint on the model's correctness. The SVD model's performance is further supported by a lower MAE value, as reported (0.270), which indicates that the predictions are more closely aligned with the average ratings. On the other hand KNN (K-Nearest Neighbors) accuracy of 0.31895 means that the KNN model correctly predicts the class labels for approximately 31.895% of the instances in the dataset.

V. CONCLUSION

Adaptive KNN-Based and SVD-Based Extended Collaborative Filtering are two advanced collaborative filtering recommendation systems that are thoroughly reviewed in this research paper's conclusion. Both approaches overcome the drawbacks of conventional collaborative filtering techniques, such as the cold-start issue and data scarcity, to give consumers recommendations that are more precise and tailored. The MovieLens dataset experiment findings show that the SVD model performs well, as seen by its lower Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) values compared to the KNN model. The SVD model is a potential method for recommendation systems in dynamic contexts due to its capacity to incorporate latent elements and manage the cold-start issue. Additionally, the Extended Collaborative Filtering approach's incorporation of external features enables a more thorough comprehends of user preferences and item properties, improving suggestion accuracy. This method's adaptability allows it to be used in a variety of sectors and provides answers to problems with real-world recommendation.

REFERENCES

[1] R. Hu & P. Pu. (2009). Potential acceptance issues of personality-ASED recommender systems. *Proceedings of ACM conference on recommender systems* (RecSys'09), pp. 22-25.

[2] Lara-Cabrera, R., González-Prieto, Á. & Ortega, F. (2020). Deep matrix factorization approach for collaborative filtering recommender systems. *Appl. Sci.*, *10*, 4926.

[3] Kumar, P. & Thakur, R.S. (2018). Recommendation system techniques and related issues: A survey. *Int. J. Inf. Technol.*, *10*, 495–501.

[4] de Campos, L.M., Fernández-Luna, J.M., Huete, J.F. & Rueda-Morales, M.A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *Int. J. Approx. Reason., 51,* 785–799.

[5] Diez, J., Delcoz, J., Luaces, O. & Bahamonde. A. (2008). Clustering people according to their preference criteria. *Expert Syst. Appl., 34,* 1274–1284.

[6] P. Pu, L. Chen, & R. Hu. (2011). A user-centric evaluation framework for recommender systems. *Proceedings of the fifth ACM conference on Recommender Systems (RecSys'11), ACM, New York, NY, USA,* pp. 57-164.

[7] Jalali M, Mustapha N, Sulaiman M & Mamay A. (2010). WEBPUM: A web-based recommendation system to predict user future movement. *Exp Syst Applicants, 37*(9), 6201–12.

[8] E. R. Nicola Barbieri & Giuseppe Manco, (2014). *Probabilistic approaches to recommendations synthesis lectures on data mining and knowledge discovery*.

[9] Y. Hu, Q. Peng & X. *H*u. (2014). A time-aware and data sparsity tolerant approach for Web service recommendation. In: *Proc. of IEEE Int. Conf. Web Serv.*, pp. 33–40.

[10] Paterek. (2007). Improving regularized singular value decomposition for collaborative filtering. In: *Proc. of KDD Cup Work.*, pp. 2–5.

[11] R. A. H. M. Rupasingha & I. Paik. (2019). Alleviating sparsity by specificity-aware ontology-based clustering for improving web service recommendation. *IEEJ Trans. Electr. Electron. Eng., 14*(10), pp. 1507–1517.

[12] X. Chen, X. Liu, Z. Huang & H. Sun. (2010). RegionKNN: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation. In: *Proc. of ICWS 2010 - 2010 IEEE 8th Int. Conf. Web Serv.*, pp. 9–16.

[13] K. Onuean. (2019). An improved items recommendation for memory-based collaborative filtering technique. pp. 51–58.

[14] C. Ramesh, K. V. C. Rao & A. Govardhan. (2017). Ontology based web usage mining model. In: *Proc. of Int. Conf. Inven. Commun. Comput. Technol.*, pp. 356–362.

[15] M. G. Vozalis & K. G. Margaritis. (2005). Applying SVD on item-based filtering. In: *Proc. of 5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, pp. 464–469.

[16] Arman Haghighi, Mostafa Safdari Shadloo, Mohammad Yaghoub Abdollahzadeh Jamalabadi Applied Sciences (Switzerland) (2020). 10.3390/APP10155384.

[17] Zhang, S., Wang, R. & Liu, W. (2019). Adaptive collaborative filtering for recommendation. *IEEE Transactions on Knowledge and Data Engineering, 32*(6), 1145-1158.

[18] Lu, X., Yu, C. & Wang, H. (2019). Context-aware collaborative filtering with adaptive k-nearest neighbors. In: *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data '19)*, pp. 536-545.

[19] Singh, A., Singh, A. K. & Agarwal, A. (2018). Collaborative filtering based on adaptive k-nearest neighbor selection. In: *Proceedings of the 2018 IEEE 7th Data Driven Innovation and Technologies for Industry (DDIT4I)*, pp. 90-93.

[20] Lu, X., Yu, C., Wang, H. & Wu, L. (2019). Adaptive collaborative filtering with dynamic k-nearest neighbor selection. In: *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM '19)*, pp. 1278-1283.

[21] Zhang, Y., Zhuang, F., Tang, J., Wang, X. & Zheng, D. (2018). Adaptive context-aware collaborative filtering for service recommendation. In: *Proceedings of the 2018 IEEE International Conference on Web Services (ICWS '18)*, pp. 181-188.

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

[23] Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*, pp. 285-295.

[24] Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In: *Proceedings of KDD Cup and Workshop*.

[25] Desrosiers, C. & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In: *Recommender Systems Handbook*, pp. 107-144.

[26] Wang, H., Wang, N., Yeung, D. Y. & Zhang, Z. (2015). Collaborative deep learning for recommender systems. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*, pp. 1235-1244.

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

[28] Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A. & Riedl, J. (2002). Getting to know you: Learning new user preferences in recommender systems. In: *Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI '02)*, pp. 127-134.

[29] Adomavicius, G. & Tuzhilin, A. (2005). 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.

[30] George Lekakos & George M. Giaglis. (2006). Improving the prediction accuracy of recommendation algorithms: Approaches anchored on human factors. *Interacting with Computers, 18*, 410–431.

[31] Huang qin-hua & Ouyang wei-min. (2007). Fuzzy collaborative filtering with multiple agents. *Journal of Shanghai University (English Edition), 11*(3), 290-295.

[32] X. Zhou, J. He, G. Huang & Y. Zhang. (2015). SVD-based incremental approaches for recommender systems. *Journal of Computer and System Sciences, 81*(4), 717-733. DOI: 10.1016/j.jcss.2014.11.016.

[33] Q. Ba, X. Li & Z. Bai. (2013). Clustering collaborative filtering recommendation system based on SVD algorithm. *IEEE 4th International Conference on Software Engineering and Service Science*, pp. 963-967. DOI: 10.1109/ICSESS.2013.6615466.