International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 10, October 2022)

A Hybrid Collaborative Filtering-Based Music Recommendation System

Karishma Kushwaha[1], Saurabh Sharma[2]

Department of Computer Science & Engineering Global Nature Care Sangathan's Group of Institutions, India

***Abstract*--Even though individuals now listen to music of many genres, algorithms are still having difficulties in many areas. How does the system determine that listeners enjoy a fresh song or artist with less historical figures? How do you decide which music to recommend to brand- new users? According to this viewpoint, the proposed research activity tries to determine the probability that a user will return to the music after their initial apparent listening experience ends throughout the time period. If the person experiences repeated auditory events within a month of the first apparent listening event, its aim is identified as 1, otherwise as 0. Additionally, techniques like factorization machines (FM), singular value decomposition (SVD), and collaborative filtering are applied. Finally, by combining SVD and FM, the proposed system is hybridised.**

***Keywords*-- Collaborative Filtering; Content Based Filtering; Singular Value Decomposition; Factorization Machine; Hybridization.**

I. Introduction

Recommendation Systems (RS) have been one of the most successful and wide spread applications emerging from the fields of Data Mining and Machine Learning. RS are responsible for filtering and selecting the most relevant items personalised to each in- dividual user‟s interests, needs and desires. RS have practically become omnipresent and indispensable since contemporary individuals, knowingly or unknowingly, have numer- ous daily encounters with and have come to rely on them for myriad tasks. The content presented in our Social Media feeds (Facebook), the ads and products in e-Commerce websites (Amazon), the „similar‟ articles in online news sites (CNN), movies and TV series suggested in on-demand streaming platforms (Netflix), available job positions (LinkedIn),vacation rentals (Airbnb) and matching dates (Tinder), among numerous others, are the result of recommendation algorithms.

Historically, people have relied on friends, family, colleagues, trusted sources and reviewers to discover products, books, movies, music, job offerings, vacation rentals etc that might be of personal interest, and most still do.

When it came to the domain of music, the audience would mostly rely on their local album store, music magazines, radio stations and online blogs for receiving relevant recommendations. Although all these sources are still available, the advent of information technology, the internet, social media and on-demand streaming services - and by extension the immense magnitudes of produced information - have necessitated the development of advanced filtering al- gorithms, able to select - among thousands or millions of items - the most appropriate and relevant items for each user. Otherwise, navigating through these amounts of data without any filtering mechanism would be practically impossible, labour intensive and extremely time-consuming leading to Information Overload and Choice Paralysis [1]. In this context, specific to the domain of music, multiple on-demand platforms have been developed, including Spotify, Pandora, Deezer, Youtube Music and Apple Music competing for the retention of user attention and engagement of millions of music lovers.

Each industry may have domain-specific requirements but the logic and architecture behind their recommendation algorithms can be very similar and are generally classified into three broad categories. Collaborative, Content-Based and Hybrid Filtering. Col- laborative Filtering (CF) models are based on the assumption that users with similar past preferences will also prefer similar items in the future. One important advantage of CF models is their architectural simplicity due to solely relying on past User - Item interactions - either explicit ratings or implicit information - without the need for further user-side or item-side meta data. Nonetheless, they frequently face difficulty recommend- ing items to new users or finding appropriate users for new items, known as the Cold Start problem. Additionally, they tend to disproportionately recommend already popu- lar items and by extension neglecting less popular ones, leading to the problem known as Popularity Bias [2]. On the other hand, Content-Based Filtering (CBF) rely on ex- tensive item meta-data in order to identify items with similar characteristics with items that a user has shown interest in the past. CBF models do not suffer from cold start and popularity bias but have a tendency for over specialisation and a lack in novelty and diversity [3].

Finally, in order to overcome the challenges that CF and CBF models tend to face, Hybrid Filtering models are being used that combine different approaches, for example aspects of both CF and CBF, leading to increased performance but usually with the trade-off of increased computational complexity [4].

In recent years, RS have been continuously advancing and becoming increasingly more complicated due to the growing magnitudes of produced information and generated content, the expanding demands of users and the ever increasing competition among different platforms to attract new users and keep them engaged. Researchers in the field of RS, apart from experimenting with models of increased architectural complexity, such as hybrid models and neural networks, are also utilizing advanced data integration and feature engineering techniques in order to increase the performance of their systems and better understand the interests and desires of their users [4]. Established techniques include Demographic Filtering, Context-Aware, Knowledge-based and Cross- Domain systems [5]. The first two techniques utilize user-based information, such as their demographics (age, gender, location etc) or the contextual environment (e.g time of day, weather etc) while the last two, employ knowledge and rules specific to the target domain or importing them from a separate but relevant domain (e.g transferring knowl- edge from book-related into movie- related recommenders). All of the above methods, require the collection of additional information that may be difficult to obtain but have proven worth-while endeavors in certain contexts. Indicatively, YouTube will suggest the most popular and trending items based on a new user‟s location and specific contextual information and demographics - mitigating the problem of user cold start (Demographic Filtering) [6], while Spotify will recommend different playlists for different time windows of the day, depending on each user‟s listening habits (Context-Aware) [7].

Along the same lines, more recently, with the advancement of psychometric tools, Personality Detection and Emotion Analysis techniques, researchers have been experimenting with extracting the personality traits or the emotional states of users and integrating them in the pipelines of recommendations algorithms. The central idea is that different personality traits may be correlated with different item characteristics (e.g acoustic features or genres in music) and that users under different emotional states and moods will prefer different types of items. Consequently, having access to users‟ psychological background could lead to increased predictive accuracy and personalisation.

In this vein, a recent survey on music recommendation systems (MRS) [5] concluded that three of the most promising visions for further improving MRS are:

- Psychologically inspired music recommendation (Personality and Emotions)
- Situation-aware music recommendation
- Culture-aware music recommendation

So far, in the domain of music, Personality-Based recommendation systems have shown limited performance and mixed results, probably due to small participant sizes and difficulty in accurately assessing personality traits [8]. On the other hand, Emotion-Aware Recommendation Systems (EA-RS) have had very promising results, showing that recommendation algorithms utilizing the emotional states and responses of users can enhance their predictive accuracy and refine personalisation [9] [10] [11]. In this direction, numerous research works have been testing different possible sources of extracting user emotions, including written comments or reviews [12], social media posts [9], face emotion recognition [13] and wearable devices [14].

II. Tasks Of A Recommender System

- Herlocker lists the following eleven typical tasks that a recommender system can assist in implementing: [3]
- Find some beneficial items: the user's search parameters displayed a featured list of the top-ranked products.
- Compile a list of every beneficial item from the item database that satisfies all of the user-specified requirements.
- Text annotation: a list of products recommended depending on the user's present environment and long-term preferences. The long-term watching habits of the viewer can be used to suggest a certain TV programme on a specific channel.
- Provide a list of objects that are similar to the one being searched for yet may be of interest to the user despite not being relevant to the search criteria.
- Offer a package: a number of connected items that work together to better meet the demands of the user. In general, when you buy a camera, you might also consider getting a memory card, a pouch, and other accessories to finish the purchase.
- Leisure browsing: The recommender system's task is to assist users who are just surfing for fun in finding items within the scope that they will find interesting during that specific browsing session.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 10, October 2022)

- Find a trustworthy recommender: Some people are sceptical of the system's suggestions. It is then the recommender system's responsibility to give the user a chance to assess the system's effectiveness.
- To improve the profile: The system can be given access to the user's explicit preferences as well as their general likes and dislikes.
- Be expressive: Even though some consumers don't give much thought to suggestions, it's important for them to be able to express their opinions about a given product. Such suggestions can be made in the comment section and submitted to the system; the satisfaction it produces can then be utilised to persuade consumers to purchase the linked item.
- Help others: Some customers may even be more likely to give a full review or rating of the product if they feel that doing so will benefit the community. And it could act as a potent inducement for other potential buyers to decide.
- Influencing others: Some users may have an outsized degree of influence, trying to convince other users to purchase or refrain from purchasing the goods. There are even malicious users in this group.

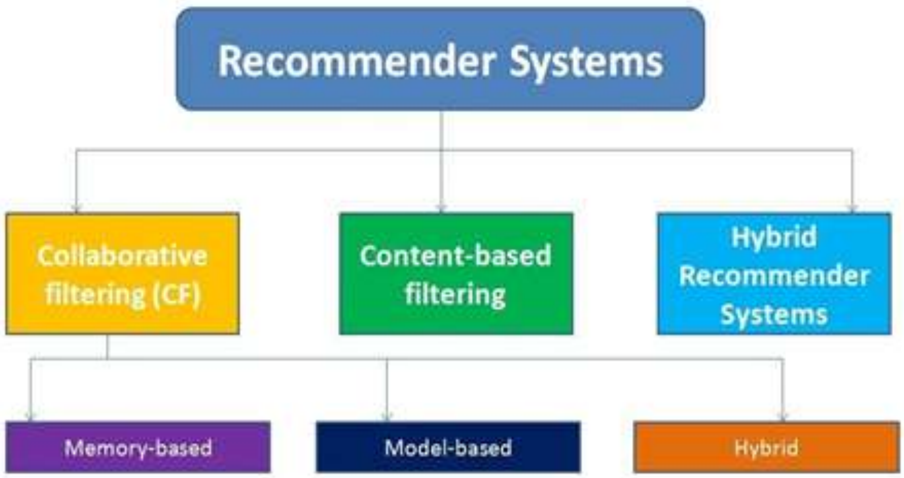

Fig. 1. Different types of Recommender systems

III. Result

The outcomes of our research on will be displayed in this section.

Utilizing movie-lens data sets and last.Fm million song data sets, recommendation systems' accuracy can be increased by merging implicit and explicit data [13,14].

With a modification in the data sparsity level, we will compare our two new algorithms to conventional collaborative filtering methods. In order to use the million song datasets [13] and movie lens datasets [15] that are the subject of this paper's discussion on music and movie recommendations, respectively.

Large and open music databases are available on Last.Fm. Each user plays a modest selection of songs, and it is made up of around a million songs and users. It had implicit input on user preferences, item spare matrices, and Last.Fm datasets for song tagging activities. Additionally, we are evaluating the effectiveness of our algorithms using 1M movie-lens data sets rated from 1 to 5. Six thousand users have rated around 4,000 internet movies, totaling one million ratings. Evaluation metrics and experimental design We compare these data sets between our novel algorithms and fundamental collaborative filtering using the precision and recall equation [16]. The table 2 description of the accuracy metric.

TABLE 2
RECOMMENDATION ACCURACY METRIC

Predicted items/actual	Relevant	Irrelevant
Recommended	True Positive (TP)	**False Positive (FP)**
Not recommended	**False Negative(FN)**	**True Negative (TN)**

The precision measure [16] the ability of the system to return relevant items among a set of irrelevant and relevant items and it's calculated by the equation (3)

Precision=TP/(TP+ FP) (3)

The Recall measure [16] the ability of the system to return the relevant items only and it's calculated by the equation (4).

Recall=TP/(TP+ FN) (4)

Another evaluation metric is an F-measure [16] used to find the difference between precision and recall function and to an equal weight of each of them. The metric equation is (equation 5). The higher result means higher accuracy of recommendation.

F-measure =(2*Precision*Recall) / (Precision + Recall) (5).

We concentrate on the amount of songs that people have explicitly evaluated on their user profiles from 1 to 5, as this will show which users are most inclined to give a song a 5 or a 2- 1 rating based on personal preference. Additionally, we concentrate on implicit user profile preference data. Depending on how many songs are played frequently or how we list songs such that they are played frequently, we divide the data into groups from 1 to 5 Songs between 20% and 80% are rated 5, and those between 20% and 1% are rated 1.

After that, we create a user-item rating matrix that we will contrast with fundamental collaborative filtering. The first preprocessing step is that. To identify patterns and develop models based on user profile preferences such as (user-song-play count), 80% of the data will be used for training and 20% for testing.

Dataset.

These actions were used last. movie-lens datasets and the FM million song datasets. We divide the data sets into three clusters to identify the song levels (level 0: all songs, level 1: song tags (pop, rock, jazz, etc.), and level 3: song duration (very short: less than one minute, short: between one minute and three, medium: between three minutes and five, long: between five and eight, very long: longer than eight minutes). this was utilised to aid association rule mining by specifying the ideal cluster sizes and using the song durations to construct it. To get the maximum performance out of an association rule, we also need to limit the quantity of the data sent to it. Following this initial preprocessing stage, we create datasets with a spare level based on song play counts.

From listing records, we divide the data into ten groups with varying levels of sparsity; the last group, with levels between (0.2 and 0.4), (0.4 -0.6), (0.6-0.8), and (0.8-1.0), has the highest level of sparsity. The sparsity level is determined by Equation 6 [13].

Sparsity measure = 1- (nR/ nUsers _ nItems) (6)

A number of songs or items on the user-item matrix are represented by the symbols nR, which stands for total number of play counts, nUsers, and nItems. The effectiveness of our algorithm will be evaluated using the predictively accurate statistic known as RMSE (root mean square error). A lot of recommendation systems use it. A lower RMSE value indicates higher performance.

Equation 7 gives the following definition of RMSE:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}(p_{u,i} - r_{u,i})^2}$$

Where the total number of ratings on the items is N, the projected rating for user u on an item is pu,i, and the actual rating is ru,i. An environment for testing on a foundation computer with 16 GB of RAM, an Intel Core I7 CPU, and Windows 7, we conduct our trials. To operate our recommender systems, we wrote Java programme code using IntelliJIDEA software. Additionally, we applied association rule and clustering-based approaches like K-means in our novel algorithms using the WEKA environment.

Experimentation Outcomes

In this section, we execute our four experiments in accordance with the level of sparsity specified in the previous section, and we compare the accuracy results of our two new proposed algorithms to those of the fundamental collaborative filtering methods. The accuracy utilising precision, recall, and F-measure metrics is displayed in Table 3. The merge datasets between the implicit and explicit datasets were used last, it is also important to note. Fm data sets [13] to discover a high level of accuracy regarding suggested products or tunes to users. The accuracy with basic CF decreases as the sparsity level increases, according to the values in table 3 and the findings given in figures 4,5,6,7. However, our two new algorithms, which combine implicit and explicit data and manage the accuracy through the sparsity level versus basic CF, increased by 22% as a result of their capacity to discover neighbours, apply association rules to offer products to users, and find neighbours.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 10, October 2022)

Ideals in Music

Because we only suggest a small number of songs to the user that don't match, utilise SVD to reduce the dimensionality of the data, and apply association rules to uncover hidden relationships, our suggested algorithms have good precision values.

We now have a greater precision value, up 37%. Our system has increased the recall based on songs that are not user-recommended by 10%. The F measure has increased by 17%, making our suggested algorithm the best in suggesting songs to users.

TABLE 3
EXPERIMENTAL RESULTS

Techniques	*Sparsity from (0.2-0.4)*		
	Precision	*Recall*	*F-measure*
Basic CF	*0.54*	*0.71*	*0.61*
SVD and clusteringalgorithm	*0.70*	*0.66*	*0.67*
Association rule and clustering algorithm	*0.96*	*0.64*	*0.76*
Techniques	*Sparsity from (0.4-0.6)*		
	Precision	*Recall*	*F-measure*
Basic CF	*0. 63*	*0.7*	*0.66*
SVD and clusteringalgorithm	*0.75*	*0.67*	*0.7*
Association rule and clustering algorithm	*0.93*	*0.64*	*0.75*
Techniques	*Sparsity from (0.6-0.8)*		
	Precision	*Recall*	*F-measure*
Basic CF	*0.57*	*0.6*	*0.58*
SVD and clustering algorithm	*0.85*	*0.61*	*0.71*
Association rule and clustering algorithm	*0.95*	*0.62*	*0.75*
Techniques	*Sparsity from (0.8-1.0)*		
	Precision	*Recall*	*F-measure*
Basic CF	*0.52*	*0.53*	*0.52*
SVD and clustering algorithm	*0.88*	*0.58*	*0.69*
Association rule and clustering algorithm	*0.89*	*0.6*	*0.71*

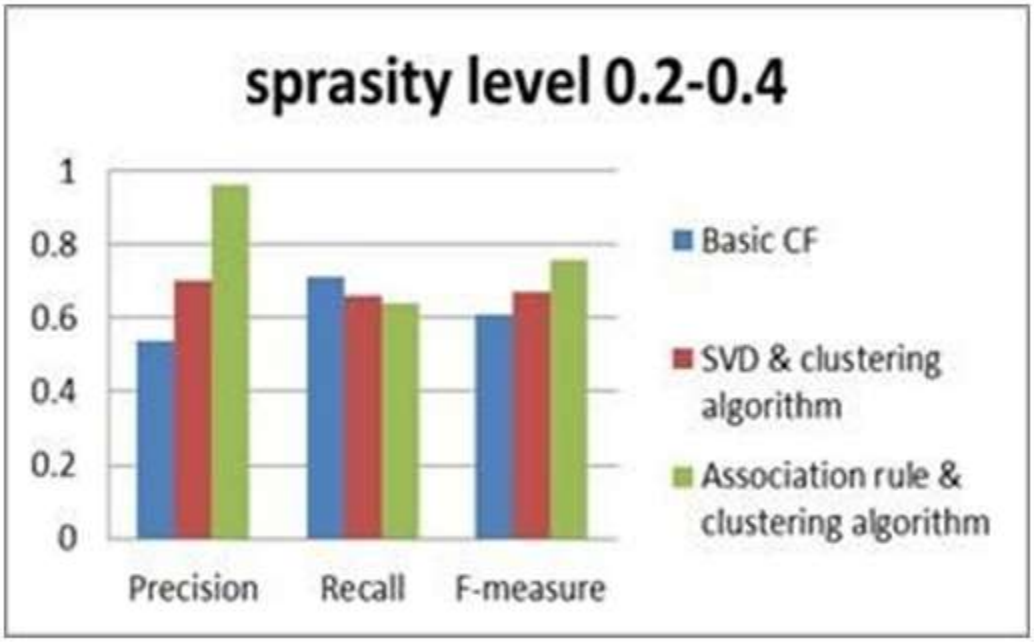

Fig. 4 experimental result while sparsity levelfrom 0.2-0.4.

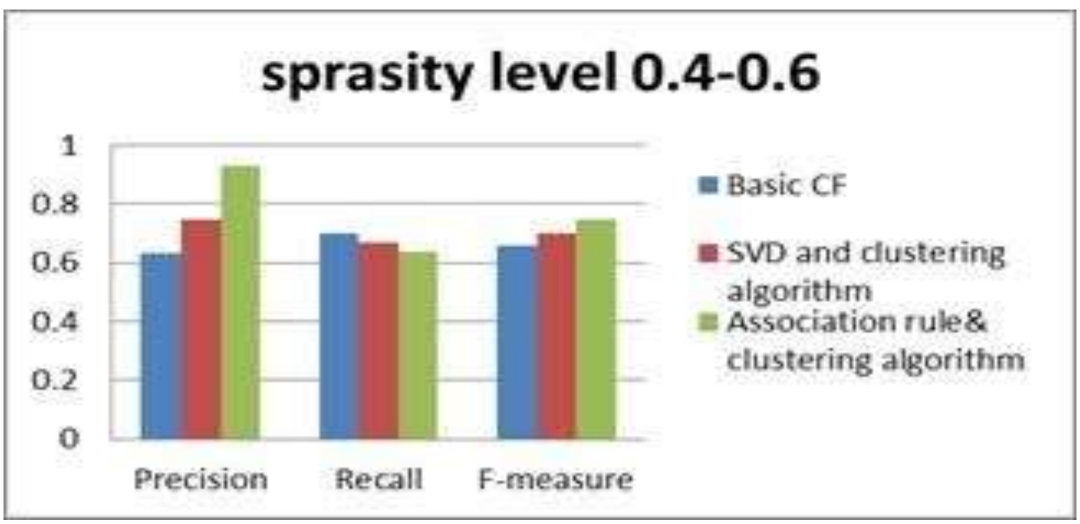

Fig. 5 experimental result while sparsity levelfrom 0.4-0.6.

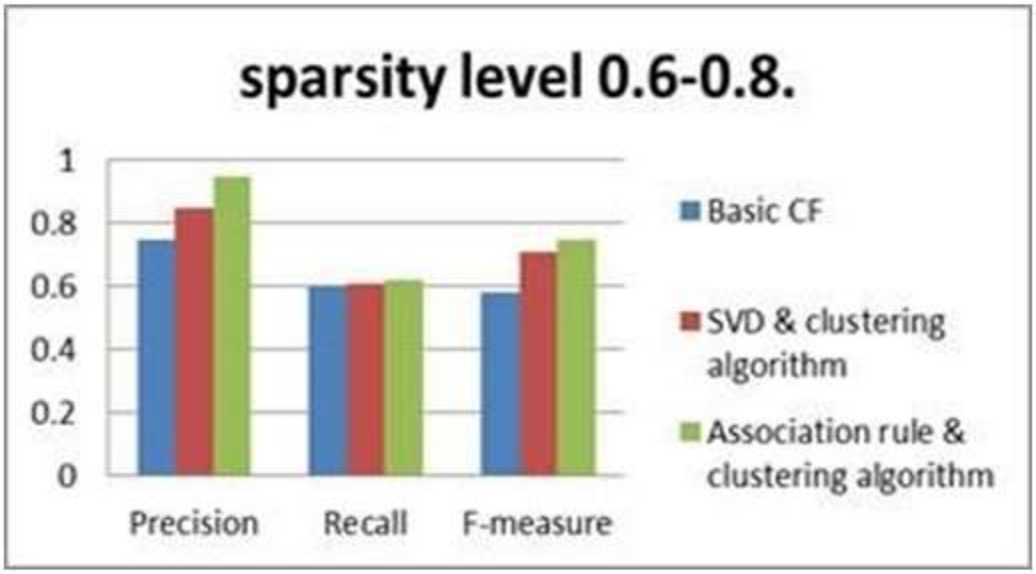

Fig. 6 experimental result while sparsity levelfrom 0.6-0.8.

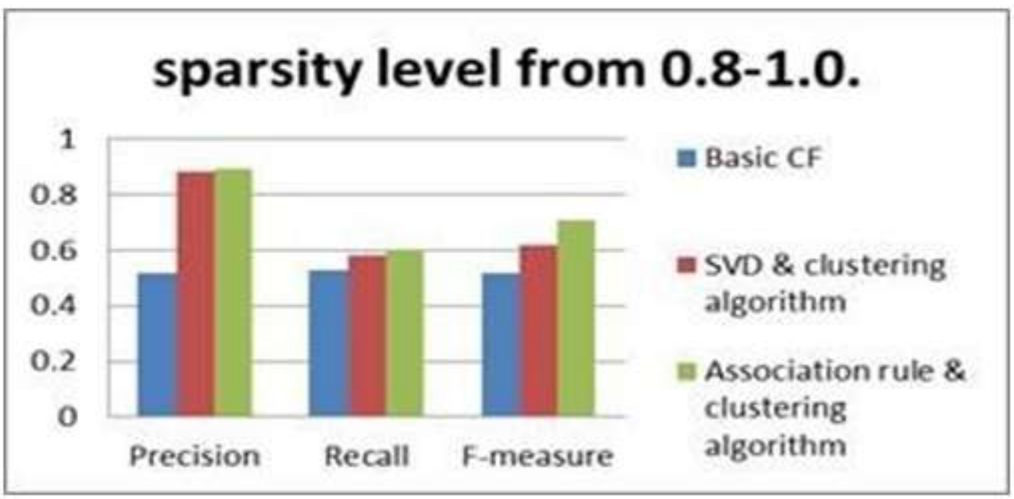

Fig. 7 experimental result while sparsity levelfrom 0.8-1.0.

In order to evaluate the performance of our suggested algorithms and make comparisons with K-means collaborative filtering methods, we use additional datasets, including the 1M movie-lens data sets. RMSE is used to evaluate performance and accuracy. When using more k-nearest neighbor-based recommendations, Figure 8 compares the RMSE value for k-means collaborative filtering to that of k-means SVD new techniques. Figure 8 demonstrates that ournovel approach outperforms k-means collaborative filtering strategies in terms of accuracy, and that it achieves the greatest RMSE results when clustering neighbours from 10 to 100.

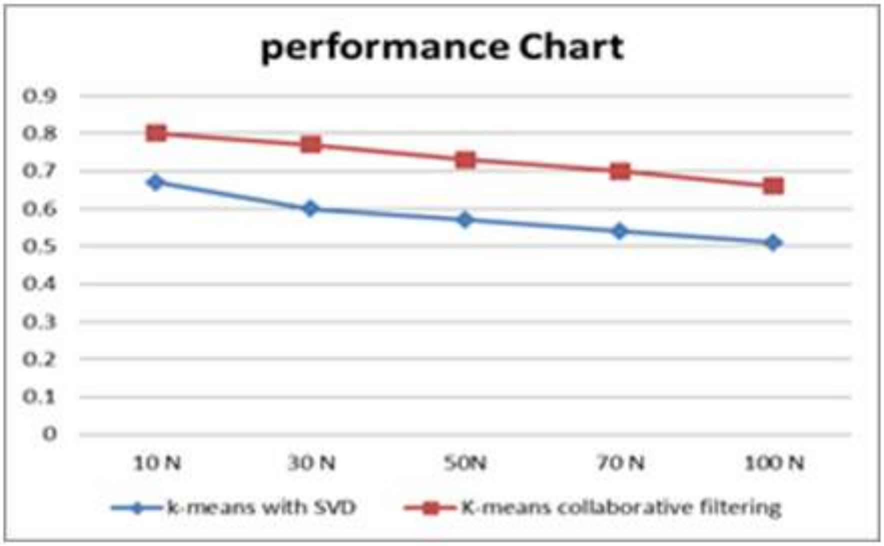

Fig. 8 Experimental results for performancemeasure.

The following are some benefits of our suggested algorithms:

- They can suggest products based on user preferences rather than just rating them based on item tags and the number of songs and movies that users have played.
- To increase accuracy, we are combining explicit and implicit data.
- The controllability of precision as sparsity grew.
- To uncover hidden connections between users, we combine the association rule with number of played counts.
- We can suggest songs to users based on their tastes and our capacity to suggest songs that are novel or diverse, which is one of the challenges with recommending products.
- Compared to standard k-means with collaborative filtering, utilising K- means with SVD increases performance when choosing the cluster centroid and using SVD to lessen the dimensionality effect.
- When employed with various data sets, our two new algorithms outperformcollaborative filtering methods.
- In this study, we evaluate accuracy and performance and address issues with accuracy and sparsity in recommender systems.

IV. CONCLUSION

We have discussed about the item based collaborative filtering method for music recommendation system with matrix factorization technique-SVD.

This system is taking the user interest into consideration without taking the user feedback explicitly as user logs areone of the implicit feedback. We addressed the problem of Sparsity by using SVD which is a dimensionality reduction technique. We also evaluated our system on benchmark dataset.

V. FUTURE WORK

This work can be extended for recommendations by taking the sessions into consideration.

REFERENCES

[1] G. Adomavicius, and Alexander Tuzhilin, ―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, Vol. 17, No. 6, June 2005.

[2] M.Kavitha Devi, P.Venkatesh, ―An Improved Collaborative Recommender System‖,2009 First International Conference on Networks & Communications- © 2009 IEEE DOI 10.1109/NetCoM.2009.69.

[3] G. Linden, B. Smith, and J. York, ―Amazon.com recommendations:item-to-item collaborative filtering,‖ IEEE Internet Computing, 2003, vol. 7, no. 1, pp. 76–80.

[4] Mojdeh Talabeigi et.al, ―A Hybrid Web Recommender System Based on Cellular Learning Automata‖, International Conference on Granular Computing, IEEE, 2011.

[5] Vivek Arvind. B Swaminathan. J Viswanathan. K. R. , ―An Improvised Filtering Based Intelligent Recommendation Technique For Web Personalization‖ – India conference (INDICON) 2012, Annual IEEE.

[6] Kangning Wei, Jinghua Huang, Shaohong Fu, ―A Survey of E-Commerce Recommender Systems‖, 1-4244-0885-7/07/$20.00 ©2007 IEEE.

[7] Jian-Guo Liu, Michael Z. Q. Chen, Jianchi Chen, Fei Deng, ―Recent Advances In Personal Recommender Systems‖ , International Journal Of C 2009 Institute For Scientific Information And Systems Sciences Computing And Information Volume 5, Number 2, Pages 230–247.

[8] Hu Jimning,―Application and Research of Collaborative Filtering in E-commerce Recommendation System‖, 978-1-4244-5540-9/10/$26.00 ©2010 IEEE.

[9] Joeran Beel , Bela Gipp, Stefan Langer, ―Corinna Breitinger4, ―Research-paper recommender systems: a literature survey‖, Received: 24 February 2014 / Revised: 10 June 2015 / Accepted: 22 June 2015 © Springer-Verlag Berlin Heidelberg 2015.

[10] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, Zuoyin Tang, ―Collaborative Filtering and Deep Learning Based Recommendation System For Cold Start Items‖, Expert Systems With Applications10.1016/j.eswa.2016.09.040, 28 September 2016

[11] Mehdi Elahi , Francesco Ricci , Neil Rubens, ―A survey of active learning in collaborative filtering recommender systems‖, http://dx.doi.org/10.1016/j.cosrev.2016.05.002 1574-0137/© 2016 Elsevier Inc. All rights reserved.

[12] Shreya Agrawal, Pooja Jain, ―An Improved Approach for Movie Recommendation System‖, International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) ©2017 IEEE.

[13] Leidy Esperanza Molina Fernández, ―Recommendation System for Netflix‖, Faculty of Science Business Analytics, January 2018.

[14] Steven Postmus,‖ Recommender system techniques applied to Netflix movie data‖, Vrije Universiteit Amsterdam, February 2018.

[15] Hamid Jazayeriy , Saghi Mohammadi and Shahaboddin Shamshirband , ―A Fast Recommender System for Cold User Using Categorized Items‖, Math. Comput. Appl. 2018, 23, 1; doi:10.3390/mca23010001.

[16] Surabhi Dwivedi , Dr Kumari Roshni, ―Recommender System for Big Data in Education‖ , 978-1-5386-1922-3/17/$31.00 © 2017 IEEE.-7/07/$20.00 ©2007 IEEE.

[17] Hossein Tahmasebi, Reza Ravanmehr, Rezvan Mohamadrezaei, ―Social movie recommender system based on deep autoencoder network using Twitter data‖, Springer- Verlag London Ltd., part of Springer Nature 2020.

[18] Soma Bandyopadhyay1 · S. S. Thakur1 · J. K. Mandal, ―Product recommendation for e‑ commerce business by applying principal component analysis (PCA) and K‑ means clustering: benefit for the society‖, Springer-Verlag London Ltd., part of Springer Nature 2020.

[19] Hamidreza Koohi, Kourosh Kiani, ―Two new collaborative filtering approaches to solve the sparsity problem‖, Springer Science+Business Media, LLC, part of Springer Nature 2020.

[20] Jesu´ s Bobadilla, Francisco Serradilla, and Jesus Bernal. A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems, 23(6):520–528, 2010.

[21] Yehuda Koren, Robert Bell, and Chris Volinsky. Ma- trix factorization techniques for recommender systems. Computer, 42(8), 2009.

[22] Antonio Hernando, Jesu´ s Bobadilla, and Fernando Or- tega. A non negative matrix factorization for collabora- tive filtering recommender systems based on a bayesian probabilistic model. Knowledge-Based Systems, 97:188–202, 2016.

[23] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. An efficient non- negative matrix-factorization- based approach to collaborative filtering for recom- mender systems. IEEE Transactions on Industrial In- formatics, 10(2):1273–1284, 2014.

[24] Mei Lu, Li Zhang, Xiang-Jun Zhao, and Fan-Zhang Li. Constrained neighborhood preserving concept factoriza- tion for data representation. Knowledge-Based Systems, 102:127–139, 2016.

[25] Meng-Hui Chen, Chin-Hung Teng, and Pei-Chann Chang. Applying artificial immune systems to collab- orative filtering for movie recommendation. Advanced Engineering Informatics, 29(4):830–839, 2015.

[26] Sang-Min Choi, Sang-Ki Ko, and Yo-Sub Han. A movie recommendation algorithm based on genre cor- relations. Expert Systems with Applications, 39(9):8079– 8085, 2012.

[27] David M Blei and Peter I Frazier.Distance dependent chinese restaurant processes. Journal of Machine Learn- ing Research, 12(Aug):2461–2488, 2011.

[28] Steven Bird. Nltk: the natural language toolkit. In Pro- ceedings of the COLING/ACL on Interactive presentation sessions, pages 69–72. Association for Computational Linguistics, 2006.

[29] Iman Avazpour, Teerat Pitakrat, Lars Grunske, and John Grundy. Dimensions and metrics for evaluating rec- ommendation systems. In Recommendation systems in software engineering, pages 245–273. Springer, 2014.