ISSN 2278-3091

Volume 9, No.5, September - October 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse249952020.pdf

https://doi.org/10.30534/ijatcse/2020/249952020

Enhancing User-based Collaborative Filtering for Music Recommendation System by combining user correlations

M.Sunitha[1], Dr.T.Adilakshmi[2], Mir Zahed Ali[3]

[1]Vasavi College of Engineering, India, m.sunithareddy@staff.vce.ac.in

[2] Vasavi College of Engineering, India, hodcse@staff.vce.ac.in

[3] Vasavi College of Engineering, India, mirzahdali@gmail.com

ABSTRACT

Music recommendation systems are developed to handle the information overloading present in music industry. Most of the research work carried out in this field is based on Collaborative filtering (CF) method. Basic CF is either user-based or item-based. This paper enhances general model-based user collaborative filtering method. K Nearest neighbor algorithm is combined with user-based CF method, to find the recommendations from the neighbors of a test user. Lift and Pearson correlation coefficient are used to find only positively related users for recommendation. Proposed methods are experimentally verified on a standard dataset, Last.fm obtained from Million Song Dataset (MSD). Proposed system performs well compared to the general user-based CF.

Key words: Collaborative filtering, Million Song Dataset, Music recommendation system, User profile information

1. INTRODUCTION

Online music streaming has become one of the prominent source for music lovers to listen to their favorite songs. Spotify, Apple Music, Pandora etc. are some popular online music service providers, are now able to collect the information about the likes and dislikes of users. As more and more users are using these services, the amount of information collected by the service providers is huge. This led to the development of music recommendation systems. There are a wide variety of recommendation systems such as for books, movies, restaurants etc. Music recommendation systems [1] are one part of recommendation system to give suggestions with respect to music. Some popular recommendation systems in use are Netflix movie recommendations or Pandora radio. Literature carried in the field of music recommendation system splits the recommendation systems into two classes. One of these two classes is Collaborative Filtering (CF) and Second one is Content Based Filtering.

The CF method [4][5] collects the information about listening behavior i.e. likes and dislikes of users. User data is collected from either implicit user feedback or explicit ratings of users about an item. History of users thus formed is used while performing recommendations. Suppose if user A listened to the songs X,Y,Z and for user B, if user A is the nearest neighbor, then CF method recommends the songs in his/her profile to user B. This type of CF is known as user-based CF. In case of item-based CF, users get recommendation of items similar to the ones already listened by him/her. k-nearest neighbor approach is the most commonly used method in CF. Cold-start, Sparsity and Longtail [6][7] are the three important challenges faced by CF methods.

Content based method [2][3] uses content of the item i.e. Timbre, Rhythm, Beats per minute etc. to find nearest neighbors for a target user. Most commonly faced challenges by content -based method is new user clod-start problem.

The research work carried in this paper is aimed to enhance basic user-based model for music recommendation. Enhancement is obtained by including user correlations into consideration.

The organization of the remaining paper is as follows. Section 2 describes related work; Section 3 explains different methods to enhance user based collaborative method and Results are shown in Section 4. Section 5 explains about Conclusion and Future scope. References are given in the last Section.

2. RELATED WORK

Collaborative Filtering (CF) is one of the most successful approaches to build recommender systems and uses the known preferences/ratings of a group of users to make recommendations or predictions of the unknown preferences for other users. A list of m users {U1, U2, ... , Um} and a list of n distinct items {I1, I2, ... , In} are present in a CF system, and each user, Ui,'s ratings of item Ij. These ratings show the implicit behavior of users. Collaborative filtering provides many advantages over Content-based filtering. Some of them are given below

No domain knowledge necessary: No domain knowledge is required because the system learns automatically from user's implicit or explicit feedback.

Serendipity: CF model help users to discover new items. New items are identified from the interests of similar users.

Great starting point: CF algorithms serve as simple and effective method based on the ratings of users.

In the process of music recommendation system users' implicit feedback is use to construct user-item rating matrix. There are few cells in the matrix where there are no values. These missing values indicated the items not preferred by that user. There are many challenges for collaborative filtering approach. CF algorithms need to deal with highly sparse data, to scale with large number of users and items, to make satisfactory recommendations in a short time etc.

The basic principle used in collaborative filtering (CF) is "If two users have similar taste in past, then they will have similar taste in future also". CF comes in two flavors, User-based CF and Item-based CF.

User-based CF works on the principle that "items of the similar users are used to provide recommendations to a test user" as shown in the Figure.1.

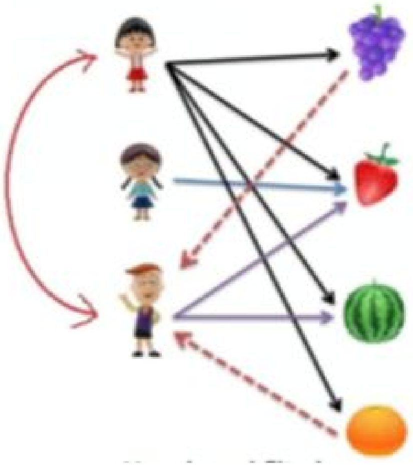


Figure.1 User-based CF

Item based CF:

Item-based CF [2] works on the basic principle that "items similar to the test user preferred items are provided as recommendations" as shown in Figure.2.

3. PROPOSED SYSTEM

Basic user-based Collaborative model shown in [8] is enhanced by combining k-nearest neighbor algorithm. Basic method also enhanced by including user profile information in the process of recommendation. User correlations are also combined to improve general model-based user collaborative filtering method.

3.1 Combining User-based Collaborative Filtering with KNN

In the process of music recommendation, user-based clustering model proposed in [8] is enhanced by combining nearest neighbors for a given test user. The algorithm used to combine user based CF model with KNN is given in Figure 3.

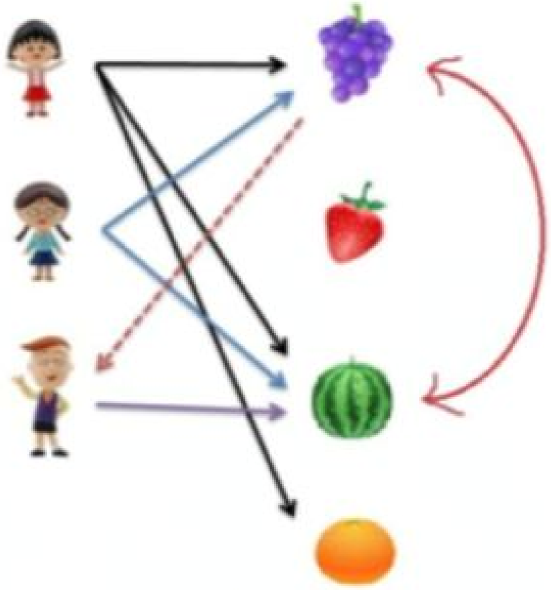


Figure.2 Item-based CF

One of the simplest supervised classification algorithm is K nearest neighbor (KNN). It is also known as lazy learner as it does not build any model when training data becomes available. As there is no model building, training will take less time and testing will take most of the time of the algorithm.

KNN works on labeled data i.e. the class label of each training sample is already known. Based on the neighbors it predicts the class label for any new unseen test data. In order to predict the label, KNN uses the nearest neighbors of test sample. K Nearest neighbors are found by using proximity measures such as Euclidean distance, Cosine similarity etc. K nearest neighbor's label is used to predict the class label of the test sample.

```
Algorithm KNN()
Input: Labeled training data, unlabeled test sample
Output: label predicted for each test sample
Method:
Begin
        1. For each test sample
        2. Find the Euclidean distance from each training
           sample
        3. Find K nearest neighbors based on the above
           calculated distances
        4. Count the number of votes for each label from K
           neighbors
        5. Predict the label of test sample as the label with
           maximum votes
End
```

Consider the following example to illustrate the working of KNN algorithm. The training sample consists of the attributes acid durability and strength to decide the quality of a special tissue paper as good or bad quality

Table.1. Sample data to demonstrate working of k-nn

S.No	Acid durability (seconds)	Strength (kg/ square meter)	quality of a special tissue paper
X1	7	7	Bad
X2	7	4	Bad
X3	3	4	Good
X4	1	4	Good

Suppose a new test sample is given with Xt=Acid durability=4 and strength=3 and K=3. To predict the quality of the tissue paper, Euclidean distance is calculated from test sample to all four training samples.

D(Xt, X1)= 5
D(Xt, X2)= 3.162
D(Xt, X3)= 1.414
D(Xt, X4)= 3.162

As K is given 3. Three nearest neighbors for Xt are X2,X3,X4.

No.of votes for (Bad)=1
No.of votes for (Good)=2
So test sample Xt quality is predicted as Good.

In the process of enhancing user-based CF recommendation system, KNN is combined with user clusters. Working of new approach is explained by using the sample user-item matrix in Table 4 and with user clusters in figure 4.

Test user, U10 is mapped to cluster UC2. To apply KNN with K=3, the proximity is calculated between U10 and the users in cluster UC2. Proximity measures used are Euclidean distance, Cosine similarity and Manhattan distance. According to these measures the nearest neighbors of U10 is given in the Table 2.

Table.2 Nearest neighbors of test user with various proximity measures

Test User	Nearest neighbor	Manhattan distance	Euclidean distance	Cosine similarity	Pearson correlation
U10	U3	8	4.47	0	2.13
	U5	9	5.92	0	3.24
	U7	10	9.06	0.1	4.54
	U9	15	9.64	0.25	6.12

As K=3, the three nearest neighbors of U10 are U3,U5 and U7. These users are used in recommending songs to U10. The list of songs recommended in case of all songs is S1, S2, S3, S4, S5, S6, S9 are recommended to U10. These songs are recommended to the test user. Alternate approach is to recommend top-n songs where n=4, only S1, S2, S3,S4 are recommended to U10. Evaluation measures are calculated based on these recommendations as given below.

Precision(U10_all songs)= 1/7=0.14
Precision(U10_top-4 songs)=1/4=0.25
Recall(U10_all songs)= 1/1=1
Recall(U10_ top-4 songs)= 1/1=1

```
Algorithm USER_CF_KNN()
Input: User clusters, test users
Output: K nearest neighbors of a given test user
Method:
Begin
    1. Let U = {U1,U2,………Ut} be the list of test users and UC={UC1,UC2,…….UCk} be the list of k clusters obtained by using user-based CF model
    2. for each test user Ut €U, find the closest user cluster from UC by using a proximity measure
    3. Let UCi be the closest user cluster for a test user Uj
    4. For each user Ul € UCi
    5. Find the distance from test user Uj as d(Uj,Ul)
    6. Sort the users in UCi in the increasing order of distance from Uj such that d(Uj,Ul)<d(Uj, Ul+1)<…..d(Uj,Ul+k-1)
    7. Consider the first K users from the sorted list
    8. Return these K users as the nearest neighbors for a test user Uj
End
```

Figure.3. Combining user-based CF model with KNN

K nearest neighbors obtained from the algorithm CF_KNN are used to find recommendations by using the recommendation algorithm defined in [8].

3.2 Enhance User-based Collaborative Filtering with user Correlations

One of the major challenges in user-based and item-based CF is Sparsity. Due to this, the recommendation system proposed in [8] with only user clusters is enhanced by considering other factors. This section discusses various methods to improve the performance of user-based CF model by considering user correlations.

In the process of recommendation user-based clusters are formed as mentioned in [8]. In order to improve the performance of recommendation system, other factors are also considered. One such feature considered in this research is correlation among users. Lift and Pearson correlation coefficient (PCC) are used to find correlation.

3.2.1. Lift: Lift is an objective measure of interestingness used in various fields such as data mining, statistics etc. It is one of the measures used to find interesting association rules. Lift is a numerical measure used to determine how items are correlated. Support and Confidence are the most commonly used measures but they fail to find association between negatively related items. Consider the example in Table 3. Here we are trying to find relation between Orange buyers and apple buyers.

Support (orange, apple)=2

$$\text{Confidence}(\{\text{orange}\}\text{->}\{\text{apple}\})=\frac{\text{Support(orange, apple)}}{\text{Support(orange)}}$$

$$= (2/7) \quad = 28\%$$

But support of apple buyers is 40%, which is contrasting the statement that 28% of orange buyers also buys apple. Here orange buyers and apple buyers are negatively related. In such cases support and confidence framework does not give useful information. Lift is used as an alternate measure.

Table 3. Sample Contingency table

	Buy apple	Not buy Apple	Row total
Buy Orange	2	5	7
Not Buy Orange	6	7	13
Column total	8	12	20

Lift is defined by using the equation 1. Lift measure contains values from -1 to 1.

$$Lift(U_i, U_j) = \frac{s(U_i \cap U_j)}{s(U_i)\, s(U_j)} \text{............}(1)$$

Lift value decides the relation between users as shown below

Lift(Ui,Uj) = <1 Ui, Uj are negatively related

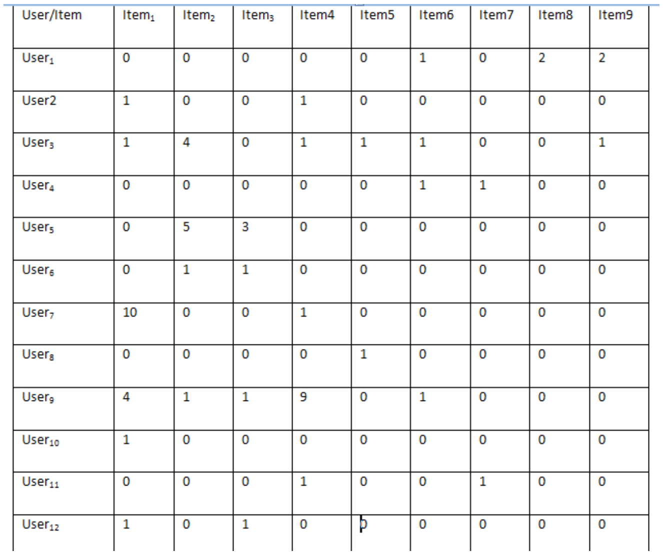
>1 Ui, Uj are positively related

=1 Ui, Uj are not related i.e. independent

Lift(orange, apple)= (2/20)/ (7/20)* (8/20) = (2 *20)/(56)=40/56=0.714 <1.

So orange buyers and apple buyers are negatively related.

Table 4. Sample user-item matrix



User/Item	$Item_1$	$Item_2$	$Item_3$	Item4	Item5	Item6	Item7	Item8	Item9
$User_1$	0	0	0	0	0	1	0	2	2
User2	1	0	0	1	0	0	0	0	0
$User_3$	1	4	0	1	1	1	0	0	1
$User_4$	0	0	0	0	0	1	1	0	0
$User_5$	0	5	3	0	0	0	0	0	0
$User_6$	0	1	1	0	0	0	0	0	0
$User_7$	10	0	0	1	0	0	0	0	0
$User_8$	0	0	0	0	1	0	0	0	0
$User_9$	4	1	1	9	0	1	0	0	0
$User_{10}$	1	0	0	0	0	0	0	0	0
$User_{11}$	0	0	0	1	0	0	1	0	0
$User_{12}$	1	0	1	0	0	0	0	0	0

In the process of enhancing user-based CF music recommendation, Lift measure is used to find only positively related users for any test user. User clusters are formed by using the algorithm [8]. Test user is mapped to the closest user cluster. Lift is used to calculate the correlation among test user and users in the mapped cluster. Users in the mapped cluster are categorized into three groups i.e. positively related users, negatively related users and independent users. Only users from positive group are considered in recommendation as shown in the example given below.

Consider the sample user-item matrix in Table 4 . First 9 users are used as training and last 3 users are used as test users. Clusters formed by applying the algorithm described in [8]. with K=2 are given Figure.4

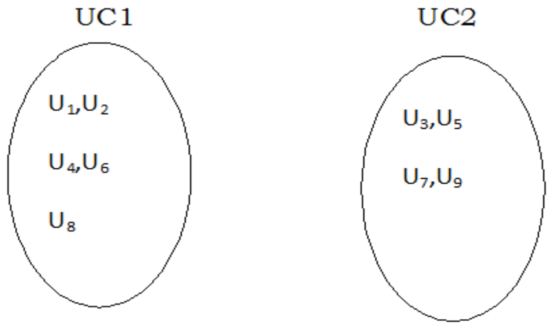


Figure.4. Sample clusters with K=2 for the data mentioned in Table 4

Algorithm USER_CF_LIFT()

Input: User clusters, Test user-item matrix

Output: Recommendation vectors for test users

```
Method:
Begin
1. Let U = {U1,U2,………Ut} be the list of test users and
       UC= {UC1,UC2,…….UCk} be the list of k
       clusters obtained by using user-based CF model
2. for each test user Uj €U, find the closest user cluster
       from UC by using a proximity measure
3. Let UCi be the closest user cluster for a test user Uj
4. For each user Ul € UCi
5. Find the correlation using Lift measure as L(Uj,Ul)
6. Find all positively related users for a test user Uj such
       that L(Uj,Ul) >1
7. Consider the positively related users and form the
       recommendation vector
End
```

Figure 5. Combining user-based CF with correlation based on Lift

Test users U10, U11 and U12 are mapped to these two clusters. U10 is closer to cluster UC2. Instead of using all users in the mapped cluster, Lift is calculated from U10 to all the users in UC2 and is shown in Table 5.

Table.5. Lift values for U10

User from nearest cluster	U3	U5	U7	U9
Lift	1.5	0	1.5	1.5

Here U10 is positively related to U3, U7 and U9. So only these users are participating in deciding the recommendation vector. U10 is provided with the items 1, 2,3,4,5,6,9.

In the process of recommendation, Lift is calculated for each test user from the users present in the mapped cluster. While performing recommendations positively correlated users are considered as shown in Figure 5. Precision, Recall and F-measure are used as evaluation measures to compare the performance with basic user-based CF model.

3.2.2. Pearson Correlation Coefficient: PCC is one of the most popular similarity measures for Collaborative filtering recommender system, to evaluate how much two users are correlated. PCC is defined as in equation 2

$$PCC(Ui,Uj) = \frac{\sum_k (R_{ki} - R_i^{-})(R_{kj} - R_j^{-})}{\sqrt{\sum_k (R_{ki} - R_i^{-})^2 ((R_{kj} - R_j^{-})^2}} \quad ....(2)$$

R_{ki} is the number of times k^th song listened by user Ui song, R_i^{-} is the average rating of user Ui. Similarly R_{kj} is the number of times k^th song listened by user Uj song, R_j^{-} the average rating of user Uj.

PCC values ranges from 0 to 1.

If PCC(Ui ,Uj) >0 Ui, Uj are positively related
<0 Ui, Uj are negatively related
=0 Ui, Uj are independent

Consider the clusters in the Figure.4. PCC is used to find the correlation for the test user U10 from users in the nearest mapped cluster UC2 as shown in Table.6.

```
Algorithm USER_CF-PCC()
    Input: User clusters, Test user-item matrix
    Output: Recommendation vectors for test users
    Method:
    Begin
        1. Let U = {U1,U2,………Ut} be the list of test
             users and UC= {UC1,UC2,…….UCk} be the
             list of k clusters obtained by using user-based
             CF model
        2. for each test user Uj €U, find the closest user
             cluster from UC by using a proximity measure
        3. Let UCi be the closest user cluster for a test user
             Uj
        4. For each user Ul € UCi
        5. Find the correlation using PCC measure as
             PCC(Uj,Ul)
        6. Find all positively related users for a test user Uj
             such that PCC(Uj,Ul) >0
        7. Consider the positively related users and form
             the recommendation vector
    End
```

Figure 6. Combining user-based CF with correlation based on PCC

Table.6. PCC values for U10

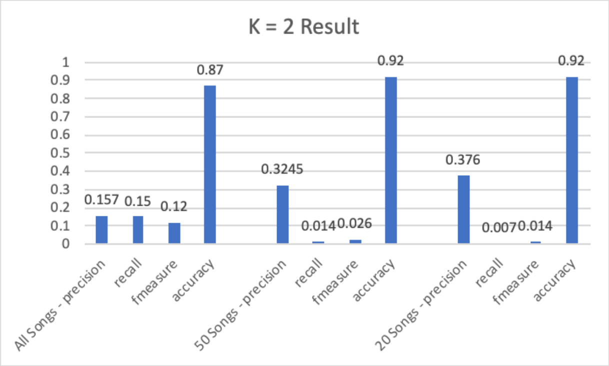
User from nearest cluster	U3	U5	U7	U9
PCC	0.75	-0.23	0.52	0.46

U10 is positively related to U3,U7 and U9, so songs are recommended from these users only as shown in Figure 6.

4. RESULTS

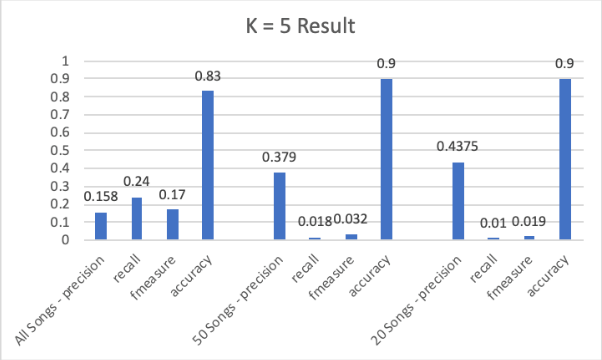
The system to enhance basic model-based user collaborative filtering is verified empirically on the dataset obtained from Last.fm internet portal for music. This data has been collected by Oscar Celma [6] as part of his research. The attributes in the user experience dataset are UserID, Timestamp, ArtistID, ArtistName, TrackID and TrackName. There is another dataset with the user demographic information such as age, location, gender and registration date. Data pre-processing is performed as shown in [8] and basic user clusters are formed for recommendations to be performed. These user clusters are used as the base for the research work carried in this paper. K-nearest neighbors are combined with user clusters to recommend only from the users, who are very much similar to the test user. Performance of the proposed recommendation system is evaluated by using the metrics, Precision, Recall and F-measure. Precision indicates the correctness of the recommendation system and its value varies from 0 to 1. Recall shows the extent to which recommendation system is able to fetch the relevant items. Recall value also lies in between 0 and 1. F-measure is a harmonic mean of precision and recall. For any test user, the recommendation vector is compared with actual user vector from the user-item matrix and finds True Positives (TP), False Positives (FP), False Negatives (FN), True Negatives (TN). Precision depends on the True Positives and False Positives and Recall depends on True positives and False Negatives.

Results obtained with k-nn are shown in Figures 7-10. Maximum precision obtained is 0.481 and is better compared to basic user-based CF model. User correlations are obtained by using Lift and PCC to find positively related users for any test user. Results with user correlations are shown in Figure 11 and Figure 12. User-based CF combined with PCC has given good precision and the obtained precision is 0.499. The results obtained with different enhancements shows the improvement over basic model user-based CF as shown in Figure 13.



Performance Evaluation of Results for User based Recommendation Using K=2 Nearest Neighbour

Figure 7. Performance of user-based combined with k-nn for K=2



Performance Evaluation of Results for User based Recommendation Using K=5 Nearest Neighbour

Figure 8. Performance of user-based clusters combined with k-nn for K=5

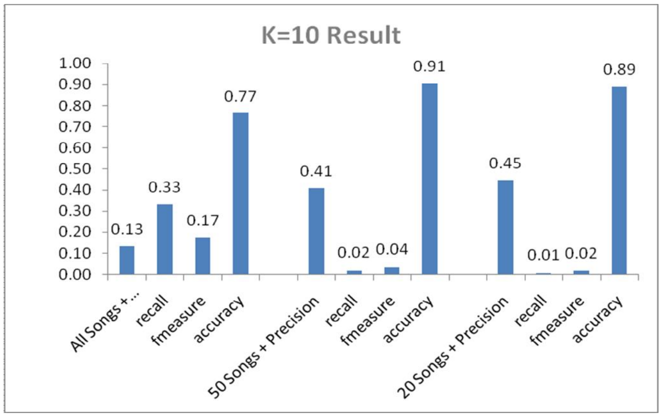
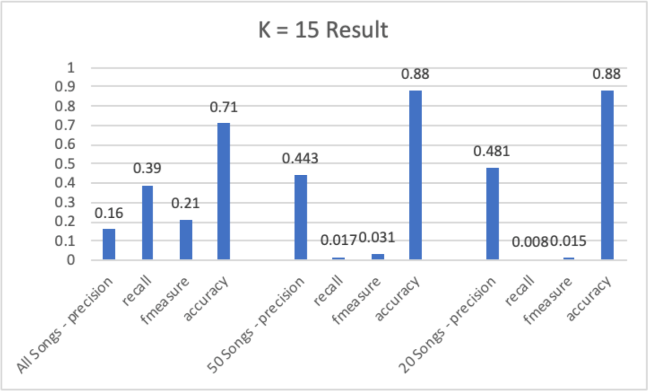


Figure 9. Performance of user-based clusters combined with k-nn for K=10



Performance Evaluation of Results for User based Recommendation Using K=15 Nearest Neighbour

Figure 10. Performance of user-based clusters combined with k-nn for K=15

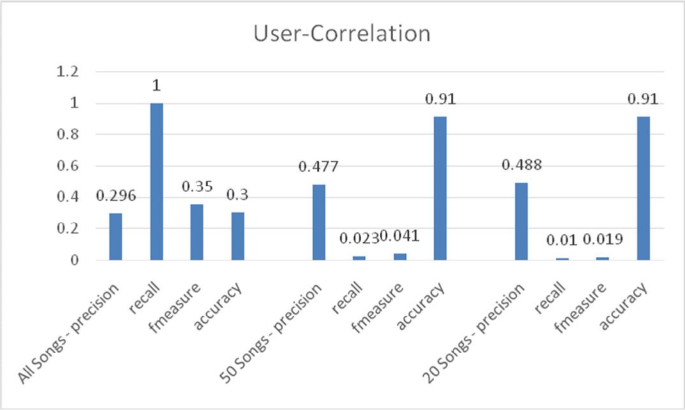


Figure 11. Performance of user-based clusters combined with correlation based on Lift

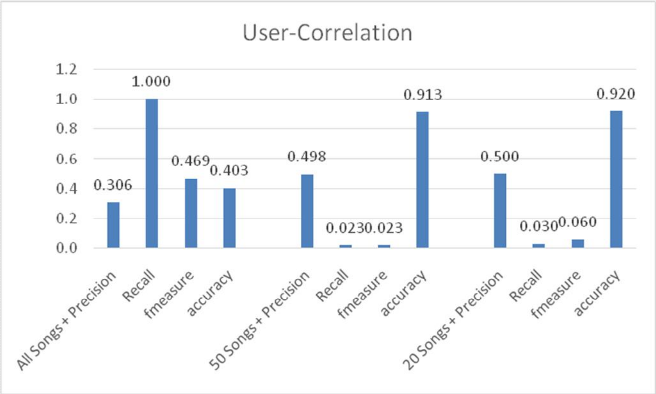


Figure 12. Performance of user-based clusters combined with correlation based on PCC

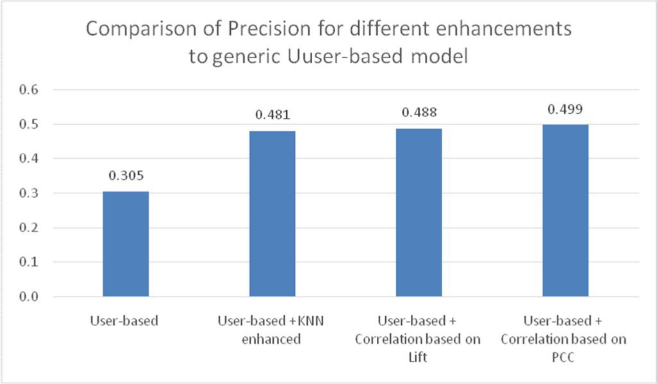


Figure 13 .Performance comparison of user-based clusters with different proposed enhancements

5. CONCLUSION

The research work carried out as part of this paper enhances basic user-based CF model. First method to enhance is combining k-nearest neighbors with user-based CF. k-nn is applied in different variations to find best recommendations for a given test user. User correlations are also obtained by using Pearson correlation coefficient and Lift. These correlations are used to find recommendations only from positively related users to any given test user. The results obtained shows that, proposed methods enhance basic user-based CF recommendation. Music recommendation is very versatile and depends on many features. Recommendation based on the user feedback and user correlations might not be able to capture all features to provide suggestions accurately. In future context of a user may also be combined as another parameter to enhance the recommendation system. Another important direction for future research is sequential songs recommendation as music recommendation is quite different from other recommendations where users might prefer to listen a sequence of songs in a row.

Acknowledgments

We would like to thank Vasavi College of Engineering, Hyderabad, India, for funding the publication of this paper.

REFERENCES

1. Aizenberg, N., Koren, Y., Somekh, O.: Build Your Own Music Recommender By Modeling Internet Radio Streams. In: Proceedings Of The 21st International Conference On World Wide Web, Pp. 1–10. ACM, 2012
2. Aucouturier, J.J., Pachet, F., Sandler, M.: “The Way It Sounds”: Timbre Models For Analysis And Retrieval Of Music Signals. IEEE Trans. On Multimedia **7**(6), 1028–1035, 2005
3. Bertin-Mahieux, T., Ellis, D.P., Whitman, B., Lamere, P.: The Million Song Dataset. In: Proceedings Of The 12th International Society For

Music Information Retrieval Conference, Pp. 591–596. Miami, USA (2011)

4. Bogdanov, D., Herrera, P.: How Much Metadata Do We Need In Music Recommendation? A Subjective Evaluation Using Preference Sets. In: Int. Society For Music Information Retrieval Conf. (ISMIR'11), Pp. 97–102 (2011)
5. Burke, R.: Hybrid Recommender Systems: Survey And Experiments. User Modeling And User-Adapted Interaction **12**(4), 331–370 (2002)
6. Celma, Ò.: Music Recommendation And Discovery – The Long Tail, Long Fail, And Long Play In The Digital Music Space. Springer, Berlin, Germany (2010)
7. Celma, O., Lamere, P.: Music Recommendation And Discovery Revisited. In: Proceedings Of The 5th ACM Conference On Recommender Systems (Recsys 2011), Pp. 7–8. ACM, New York, NY, USA (2011)
8. M. Sunitha, T. Adilakshmi, Comparison Of User-Based Collaborative Filtering Model For Music Recommendation System With Various Proximity Measures, International Journal Of Innovative Technology And Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-6S2, April 2019