OPTIMIZATION OF RECOMMENDATION SYSTEM USING MACHINE LEARNING APPROACH

Anmol Alawadhi[1], Vikas Thada[2] & Khushboo Tripathi[3]
[1]*M.Tech(AI&R), Deptt. of Computer Science & Engineering, Amity University Haryana, India*
[2]*Associate Professor, Deptt. of Computer Science & Engineering, Amity University Haryana, India*
[3]*Assistant Professor, Deptt. of Computer Science & Engineering, Amity University Haryana, India*

Abstract

Recommendation system plays important role in Internet world and used in many applications. It has created the collection of many application, created global village and growth for numerous information. This paper represents the overview of Approaches and techniques generated in recommendation system. Recommendation system is categorized in three classes: Collaborative Filtering, Content based and hybrid based Approach. This paper classifies collaborative filtering in two types: Memory based and Model based Recommendation.The paper elaborates these approaches and their techniques with their limitations. This survey shows the road map for research in this area.

In the very recent years, development of recommendation system has been a more heated problem due to a higher level of data consumption and the advancement of machine learning techniques. Some traditional approaches such as collaborator filtering, has been widely used in recommendation systems, have helped recommendation system to give users a quick access to the data. However, collaborative filtering or content based filtering have limitations in giving a better result with the ignorance of combination factor of lyrics and genre.

In my paper, I will propose an improved algorithm based on machine learning on hybrid approach using collaborative filtering, content based filtering and popularity based filtering. The proposed method will make it possible that it could make recommendations in a large system to make comparisons by "understand" the content of data. In this paper, I propose an end-end model, which is based on machine learning approach to predict user's next most possible data by similarity.Experiments made and evaluations based on Dataset and demonstrate how it outperformed the traditional methods.

Keywords **-** Recommendation, Collaborative filtering, Model based, Memory based, Content based, Hybrid.

I. INTRODUCTION

Recommendation System is part of Daily life where people rely on knowledge for making decision of their personal interest. Recommendation system is subclass of information filtering to predict preferences to the items used by or for users. Although there are many approached developed in past but search still goes on due to it's often usage in many applications, which personalize recommendation and deals with information overload. These demands throws some challenges so different approaches like memory based, model based are used. Recommender system still requires improvement to become better system.

Recommendation system is a sharp system that provides idea about item to users that might interest them some examples are amazon.com, movies in movielens, music by last.fm. In this paper different approached with their techniques are mentioned to compare the limitation of each technique in proper manner to provide proper future recommendations.

1.1 PROBLEM STATEMENT

This paper is based on recommendation system that recommends different things to users. This system will recommend movies to users. This system will provide more precise results as compared to the existing systems. The existing system works on individual users' rating. This may be sometime useless for the users who have different taste from the recommendations shown by the system as every user may have different tastes. This system calculates the similarities between different users and then recommend movie to them as per the ratings given by the different users of similar tastes. This will provide a precise recommendation to the user. This is a web based as well as android system where there is a movie web service which provides services to user to rate movies, see recommendations put comments and see similar movies.

1.2 AIM

To develop recommendation system using machine learning approach

1.3 Objective

- To study background related to recommendation and machine learning approach
- To design and implement methodology for recommendation system using machine learning approach
- To analyse the results for the parameters: Popularity based system, Content based system, Collaborative Filtering based system using hybrid approach

II. BACKGROUND

A variety of approaches has been used to provide recommendation like collaborative filtering, content based and hybrid approach. Different Algorithms and approaches are there to provide recommendation that may use rating or content information; however collaborative filtering and content based method suffer from same limitations. Several researchers have tried to overcome these limitations by combining both collaborative filtering and content based method as a hybrid approach that combined ratings as well as content information. Recommendation system will always remain active search area for researchers [1].

2.1 Approaches of Recommendation System

Recommendation system is usually classified on rating estimation

- Collaborative Filtering system
- Content based system
- Hybrid system

In content-based approach, similar items to the ones the user preferred in past will be recommended to the user while in collaborative filtering, items that similar group people with similar tastes and preferences like will be recommended. In order to overcome the limitations of both approach hybrid systems are proposed that combines both approaches in some manner [15].

2.2 Collaborative filtering system

Collaborative filtering systems work by collecting user remark in the form of ratings for items in a given field and exploiting similarities in rating actions amongst several users in determining how to recommend an item. Collaborative filtering systems recommend an item to a user based on opinions of other users. Like, in a movie recommendation application, Collaborative filtering system tries to find other like-minded users and then recommends the movies that are most liked by them. Although there are many collaborative filtering techniques, they can be divided into two major categories [2]:

:

- Memory Based approaches
- Model Based approaches

2.2.1 Memory based Approach

Memory-based techniques continuously analyze all user or item data to calculate recommendations and can be classified in the following main groups: CF techniques, Content-Based (CB) techniques and hybrid techniques. CF techniques recommend items that were used by similar users in the past; they base their recommendations on social, community-driven information (e.g., user behavior like ratings or implicit histories). CB techniques recommend items similar to the ones the learners preferred in the past; they base their recommendations on individual information and ignore the offerings from other users. Hybrid techniques combine both techniques to provide more accurate recommendations. A hybrid RS could combine CF (or social-based) techniques with CB (or information-based) techniques. If no efficient information is available to carry out CF techniques, it would switch to a CB technique [4].

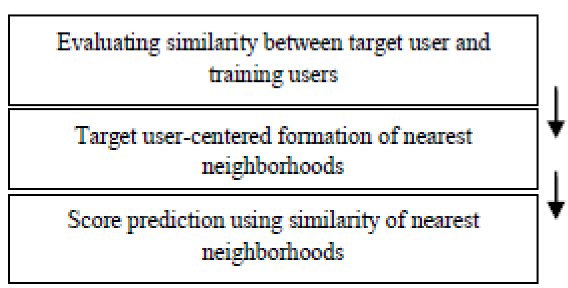

Fig 1 Block Diagram of Memory Based RS [4]

The prediction process in memory-based CF contains three steps. They are similarity evaluation, generation of nearest neighborhoods and score prediction. For evaluation of the performance, the CF system considers the mean absolute error (MAE), precision and recall. The CF performance varies according to the processing method of each step[5].

A) Existing Similarity Measures

The most important first step in memory-based CF is similarity evaluation. The CF system in this step evaluates the similarity between the target user and other users for common rating items. The similarity is used as a weight for predicting the preference score. Various similarity metrics have been proposed in previous studies. These are as follows [8][10][17]:

Tanimoto coefficient. It is similarity between two sets. It is a ratio of intersections. Assume that set X is {B,C, D} and set Y is {C, D, E}. The Tanimoto coefficient T of two set A and B is 0.5. This metric doesn"t consider the user rating but the case of a very sparse data set is efficient[8].

Fig 1 Block Diagram of Memory Based RS [17]

The prediction process in memory-based CF contains three steps. They are similarity evaluation, generation of nearest neighborhoods and score prediction. For evaluation of the performance, the CF system considers the mean absolute error (MAE), precision and recall. The CF performance varies according to the processing method of each step[17].

Cosine similarity. The Cosine similarity is known as the Vector similarity or Cosine coefficient. This metric assumes that common rating items of two users are two points in a vector space model, and then calculates cosΘ between the two points[10][8][18].

Person's Correlation. In Equation, SU1 is the standard deviation of user U1. The Pearson Correlation measures the strength of the linear relationship between two variables. It is usually signified by *r*, and has values in the range [-1.0,1.0]. Where -1.0 is a perfect negative correlation, 0.0 is no correlation, and 1.0 is a perfect positive correlation[4][10][8][18].

Spearman's Rank Correlation. The Spearman Rank Correlation also measures the strength of the linear relationship between two variables. Unlike the Pearson Correlation, this metric considers rank of scores. So this similarity measure has more general applicability than the Pearson Correlation, which isn't suitable outside a normalized preference range. Because the range of preference scores for CF is normalized, the Spearman Rank Correlation in the CF field shows comparable performance to the Pearson Correlation[8]

B) Formation of Nearest Neighbor

The second step after the similarity evaluation is generation of nearest neighborhoods. To improve performance, many methods have been proposed by CF researchers. The methods for selecting nearest neighborhoods include classification using K-means, a threshold for the number of common rating items and a graph algorithm. In general, it selects similar users greater than a given threshold or high rank users[8][10].

C) Prediction of Preference Score

The last step in memory-based CF is to predict the preference score of the target user for non-rating items. It predicts the preference score of non-rating items for the target user, based on the rating of nearest neighborhoods. Various methods have been proposed, and Weighted Mean is used as most general algorithm. PSU1,Ii is the predicted score of item i for U1 , and NNUi is the nearest neighbor i[8].

D) Performance Evaluation

In the CF system, there are two types of measure for the performance evaluation. The first type is prediction accuracy, which is evaluated by MAE. Pi is the real preference score of item i and qi is the predicted score of item i[8].

Merits and Demerits of Memory Based Approach

User-based techniques correlate users by mining their (similar) ratings and then recommend new items that were preferred by similar users. Item-based techniques correlate the items by mining (similar) ratings and then recommend new, similar items. The main advantages of both techniques are that they use information that is provided bottom-up by user ratings, that they are domain-independent and require no content analysis and that the quality of the recommendation increases over time. CF techniques are limited by a number of disadvantages. First of all, the so-called „cold start" problem is due to the fact that CF techniques depend on sufficient user performance from the past. Even when such systems have been running for a while, this problem emerges when new users or items are added. New users first have to give a sufficient number of ratings for items in order to get accurate recommendations based on user-based CF (new user problem)[9]. New items have to be rated by a sufficient number of users if they are to be recommended. Another disadvantage for CF techniques is the sparsity of the past user actions in a network. Since these techniques deal with community-driven information, they support well-liked tastes more strongly than unpopular tastes. The learners with an unusual taste may get less qualitative recommendations, and learners with common taste are unlikely to get unpopular items of high quality recommended. Another common problem is scalability. RSs which deal with large amounts of data, like amazon.com, have to be able to provide recommendations in real time, with the number of both the users and items exceeding millions[10].

2.2.2 Model Based Approach

K-MEANS CF: *k*-means clustering is applied to identify the segments. *k*- means is a clustering method that has found wide application in data mining, statistics and machine learning. The input to *k*-means is the pair-wise distance between the items to be clustered, where the distance means the dissimilarity of the items. The number of clusters, *k* is also an input parameter. It is an iterative algorithm and starts with a random partitioning of the items into *k* clusters. Each iteration, the centroids of the clusters is computed and each item is reassigned to the cluster whose centroid is closest.

CLUSTER MODEL: To find customers who are similar to the user, cluster models divide the customer base into many segments and treat the task as a classification problem. The algorithms goal is to assign the user to the segment containing the most similar customers. To find customers who are similar to the user, cluster models divide the customer base into many segments and treat the task as a classification problem[2].

The algorithm's goal is to assign the user to the segment containing the most similar customers. It then uses the purchases and ratings of the customers in the segment to generate recommendations. The segments typically are created using a clustering or other unsupervised learning algorithm, although some applications use manually determined segments. Using a similarity metric, a clustering algorithm groups the most similar customers together to form clusters or segments. Because optimal clustering over large data sets is impractical, most applications use various forms of greedy cluster generation. They then repeatedly match customers to the existing segments, usually with some provision for creating new or merging existing segments. Once the algorithm generates the segments, it computes the user's similarity to vectors that summarize each segment, then chooses the segment with the strongest similarity and classifies the user accordingly. Some algorithms classify users into multiple segments and describe the strength of each relationship. Cluster models have better online scalability and performance than collaborative filtering3 because they compare the user to a controlled number of segments rather than the entire customer base. The complex and expensive clustering computation is run offline. However, recommendation quality is low.1 Cluster models group numerous customers together in a segment, match a user to a segment, and then consider all customers in the segment similar customers for the purpose of making recommendations. Because the similar customers that the cluster models find are not the most similar customers, the recommendations they produce are less relevant[2].

III. LITERATURE REVIEW

Bonnin and Jannach in [7] review and compare the mostly used methods for automatic playlist generations and recommendations. As important issues in music playlists are the co-occurrence of songs and the smooth transition among them, Markov models using songs as states and association rules or sequential patters mining are among the techniques used in this domain. The usual recommendation approaches treating playlists as users and comparing them through cosine similarity measures or combined with rank prediction algorithms or content based approaches to find tracks with similar musical features, can also be used. The major limitations of these methods are that they are computationally expensive and sometimes work based on strong assumptions while their overall effectiveness depends heavily on the type of the used data. Finally, due to the long tail distribution present in the music domain, the authors propose two popularity-based recommendations approaches that also include some artist information.

Hyeong-Joon Kwon, Tae-Hoon Lee, etal.[8] base their reasoning on the implicit likeliness that can be found in the playlists generated by professionals, like music radio stations and Djs. Usually, the items placed into sets together with higher frequency have some common characteristics, like a particular genre, pairwise suitability, relative popularity etc. They model the transmissions between musical items based on a graph representation, where adjacent songs are represented as nodes and each arc has weight equal to the number of times that this transition was observed. The resulting graph is transformed into a Markov random field where a playlist can be generated as a random walk starting from a given song (node) and using the Markov transition probabilities.

Baccigalupo and Plaza in [9], present an interesting Case-Based Reasoning approach to music playlists' recommendation with aim to generate playlists of a desired length, being both varied and coherent. Every playlist is treated as a case and their relevance is computed based on their songs co-occurrences and a recommendation is formed as a combination of the items in the most similar lists. The authors also analyze the properties that may bias the effectiveness of their approach, namely songs' popularity and sub-lists' length. As the context within which a music item is consumed or a playlist is generated is of high importance,

Domingues et al. [10] present an interesting approach of incorporating the contextual parameters within the recommendation model. More than performing a pre or post filtering based on the actual context, they represent it as "virtual items". Their results show that these additional dimensions are able of improving the recommendations' accuracy when combined with the usual recommendation algorithms. In addition this contextual modeling may also enable the access to less popular or novel but however relevant to the active user, items, Finally a slightly different approach to music recommendations is presented by **Rosa et al. [11]**. The authors associate music songs with users' sentiments as these can extracted from the users' posts in their social networks by using lexicon-based sentiment metrics. However, these last papers focus on song recommendations and would possibly need to be reformulated or extended in order to be used for playlist recommendations.

Personalized recommendation is a typical way of personalized service, which actively to push targeted resources to users according to the user s preferences and the user s evaluation or feedback on the project, so as to achieve the purpose of decision support and information services. For music personalized recommendation, the commonly used methods include content-based recommendation technology, the collaborative filtering recommendation technology and hybrid recommendation technology. The content-based recommendation is concerned with some of the characteristics of the music itself, which mainly use the metadata related with the user s favorite music to match the information. First of all, obtain the metadata of user s favorite music through the user s historical records. Then the music with similar content is obtained by calculating the similarity between the metadata and recommend to the user. By comparing the acoustic features that are extracted from the music with the user s preferences, music that has similar acoustic features is recommendation to the user **[12]**. Collaborative filtering recommendation technology is used to explore the user s new interest points via user rating data for music. In order to fully mine the user s preferences, it often combined with music tags to study.

IV.METHODOLOGY

Hybrid approach-

Popularity based + Content based + Collaborative based

- It has been taken into account minimum rating while fetching our data i.e. **popularity based**.
- After that, took the loss function called WARP (Weighted Approximate Rank Pairwise)
- Warp helps us to create recommendations for each user by looking at the existing user rating pairs and predicting rankings for each.
- It uses the gradient descend algorithm to find the weights that improve our prediction over time. This takes into account both the users past rating history i.e. **content based** and similar users rating i.e. **collaborative based**.

Approach – 1

Popularity based-

It creates an instance of popularity based recommender class and feed it with our training data. This achieves the following goal: based on the popularity of each song, create a recommender that accept a user_id as input and out a list of recommended song of that user

The logic can be seen more clearly here **-** Based on the number of users or guests that rated place1 and place 2, we'd say that place 1 is more popular than place 2, so based on popularity, place 1 would be recommended over place 2

Approach – 2

Content Based-

Content based systems predict what you like based on what you have liked in the past.

Starting from an explicit set of music tracks provided by the user as evidence of his/her music preferences, it is infered high-level semantic descriptors, covering different musical facets, such as genre, culture, moods, instruments, rhythm, and tempo. On this basis, two of the proposed approaches employ a semantic music similarity measure to generate recommendations. The third approach creates a probabilistic model of the user's preference in the semantic domain.

Approach – 3

Collaborative filtering -

Collaborative systems predict what you like based on what other similar users have liked in the past.

It is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if a person *A* has the same opinion as a person *B* on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person.

Collaborative Filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc. Applications of collaborative filtering typically involve very large data sets.

V. CONTENT BASED APPROACH

Any Systems implementing a content-based recommendation approach analyze a set of documents and/or descriptions of items previously rated by a user, and build a model or profile of user interests based on the features of the objects rated by that user. The recommendation process basically consists in matching up the attributes of the user profile against the attributes of a content object. The result is a relevance judgment that represents the user’s level of interest in that object. If a profile correctly reflects user preferences, it is of tremendous advantage for the effectiveness of an information access process[11].

Methods for Content Based Feature Selection[12]

1) Wrapper methods evaluate different subsets of features by training a model for each subset and then evaluating each subset's contribution on a validation dataset. As the number of all possible subsets is factorial in the number of features, different heuristics are used to choose “promising" subsets (forward-selection, backward-elimination, tree-induction, etc.). Wrapper methods are independent of the prediction algorithm[13].

2) Filter methods are typically based on heuristic measures, such as Mutual Information or Pearson Correlation, to score features based on their information contents with respect to the prediction task. Similar to wrapper methods, filter methods are also independent of the algorithm in use. However, they do not require training many models and therefore scale well for large datasets. Yet, filter methods cannot be naturally extended to recommender systems, in which the prediction target varies and depends both on the user's history and on the item under consideration. This work proposes a framework and algorithms to address the above difficulties[14].

3) Embedded methods are a family of algorithms in which the feature selection is performed in the course of the training phase. Unlike wrapper methods, they are not based on cross-validation and therefore scale with the size of the data. However, since the feature selection is an inherent property of the algorithm, an embedded method is tightly coupled with the specific model: If the recommendation algorithm is replaced, features selection needs to be revisited[15].

Techniques of Content Based Approach

TF-IDF : The terms that occur frequently in one document (TF=term-frequency), but rarely in the rest of the corpus (IDF = inverse-document-frequency), are more likely to be relevant to the topic of the document. In addition, normalizing the resulting weight vectors prevent longer documents from having a better chance of retrieval.

NAÏVE BAYES: Naıve Bayes is a probabilistic approach to inductive learning, and belongs to the general class of Bayesian classifiers. These approaches generate a probabilistic model based on previously observed data.

Merits and Demerits of Content Based approach

The approval of the content-based recommendation paradigm has several advantages:

USER INDEPENDENCE - Content-based recommenders exploit solely ratings provided by the active user to build her own profile. Instead, collaborative filtering methods need ratings from other users in order to find the “nearest neighbors” of the active user[11].

TRANSPARENCY - Explanations on how the recommender system works can be provided by explicitly listing content features or descriptions that caused an item to occur in the list of recommendations. Those features are indicators to consult in order to decide whether to trust a recommendation[11].

NEW ITEM - Content-based recommenders are capable of recommending items not yet rated by any user. As a consequence, they do not suffer from the first-rater problem, which affects collaborative recommenders which rely solely on users‟ preferences to make recommendations. Therefore, until the new item is rated by a substantial number of users, the system would not be able to recommend it[11].

content-based systems have several shortcomings:

LIMITED CONTENT ANALYSIS - Content-based techniques have a natural limit in the number and type of features that are associated, whether automatically or manually, with the objects they recommend.

OVER-SPECIALIZATION - Content-based recommenders have no inherent method for finding something unexpected. The system suggests items whose scores are high when matched against the user profile; hence the user is going to be recommended items similar to those already rated. This drawback is also called *serendipity* problem to highlight the tendency of the content-based systems to produce recommendations with a limited degree of novelty.

NEW USER **-** Enough ratings have to be collected before a content-based recommender system can really understand user preferences and provide accurate recommendations. Therefore, when few ratings are available, as for a new user, the system will not be able to provide reliable recommendations[11].

VI. HYBRID APPROACH

Traditional recommender system techniques such as collaborative filtering (CF), content-based, and knowledge-based filtering, each have unique strengths and limitations. For example, CF suffers from sparsity and cold start problems, while content-based approaches suffer from narrowness and require descriptions. However, a hybrid approach can use one approach to make predictions where the other fails, resulting in a more robust recommender System[1][13].

Types of Hybrid

Weighted Hybrid. In this approach, a score for each recommended item is simply the weighted sum of the recommendation scores for each source. Weights for each context source are user-configurable through interactive sliders. Automatically optimizing the set of weights for each context source is desirable, but not trivial. Empirical bootstrapping can be used to calculate an optimal weighting scheme; however, historical data is needed for this approach[13].

Mixed Hybrid. In this approach, recommendations for each source are ranked, and then the top-n are picked from each source, one recommendation at a time by alternating the sources. This approach only considers relative position in a ranked list and does not include individual recommendation scores. In cases where a recommendation is produced by multiple context sources (i.e. was previously picked from another source) the algorithm simply selects the next recommendation from the ranked list for that source[13].

Cross-Source Hybrid. This approach strongly favors recommendations that appear in more than one source. Is is believed that if a recommendation is generated from more than one context source / algorithm, i.e. by both collaborative Filtering (Facebook) and content-based recommendation (Wikipedia), then it should be considered more important. To compute a final recommendation set, the weighted hybrid approach is first applied, then each recommendation's weight is multiplied by the number of sources in which it appeared. The following equation describes the the cross-source hybrid approach:

Wreci =Σsj2S(Wreci;sj _Wsj) *| Sreci |

where jSreci j is the number of context sources recommendation i was generated by (i.e. 1, 2, or 3)[13].

How Hybrid Approach Works:

In a Movie Recommender system, the content based part of the movie recommender is based on a naive Bayesian text classification method. The classifier creates a naive Bayesian model for every user, based on the content of the movies the user has rated.

Issue with Hybrid Approach

Reliable Integration*:* The first problem is to reflect the collaborative and content-based data when making recommendations. An easy solution is to use collaborative and content-based methods in parallel or in cascade. However, such an approach has drawbacks. Although Meta recommender systems have been proposed to select a recommender system among conventional ones on the basis of certain quality measures the disadvantages of the selected system are inherited. Moreover, the heuristics-based integration dealt with in other studies lacks a principled justification[5].

Efficient Calculation: The second problem, which has been scarcely dealt with, is to efficiently adapt a recommender system according to the increase in rating scores and users. An easy solution is to take a memory-based approach, which is originally free from this problem because the whole data is always used to make recommendations. However, these results in the late responses tried to overcome this disadvantage by using a probabilistic method in a pure collaborative filtering context. On the other hand, proposed an efficient method that incrementally trains an aspect model used for model-based collaborative filtering. To our knowledge, there are no studies on incremental adaptation of hybrid recommender systems. It need to carefully design hybrid architecture while considering whether the previous prominent methods can be applied or not[5].

RESULT AND DISCUSSION

Popularity Based-

We create an instance of popularity based recommender class and feed it with our training data. The code below achieves the following goal: based on the popularity of each song, create a recommender that accept a user_id as input and out a list of recommended song of that user

```
pm = Recommenders.popularity_recommender_py()
pm.create(train_data, 'user_id', 'song')
#user for the popularity model to make some
prediction
user_id = users[5]
pm.recommend(user_id)
```

Table 5.1: Popularity Based Output

	user_id	song	score	Rank
3194	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Sehr kosmisch - Harmonia	37	1
4083	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Undo - Björk	27	2
931	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Dog Days Are Over (Radio Edit) - Florence + Th...	24	3
4443	4bd88bfb25263a75bbdd467e74018f4ae570e5df	You're The One - Dwight Yoakam	24	4
3034	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Revelry - Kings Of Leon	21	5
3189	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Secrets - OneRepublic	21	6
4112	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Use Somebody - Kings Of Leon	21	7
1207	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Fireflies - Charttraxx Karaoke	20	8
1577	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Hey_ Soul Sister - Train	19	9
1626	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Horn Concerto No. 4 in E flat K495: II. Romanc...	19	10

Hybrid system-

Sample recommendation

We will call it at the end using the model data and a list of 3 or more random user ids as the parameters.

Each user has movies that they listed as well as movies that our system has recommended for them.

So basically recommendation algorithms help us make decisions by learning our preferences.

After entering the final step, we will get 2 results-

- It will print the top three known positive movies that the user has picked.
- It will print the top three recommended movies that our model predicts which is required.

```
User 25
     Known positives:
        Dead Man Walking (1995)
        Star Wars (1977)
        Fargo (1996)
     Recommended:
        Contact (1997)
        Fargo (1996)
        L.A. Confidential (1997)
```

VII. CONCLUSION

Several recommendation systems have been anticipated are based on collaborative filtering, content based filtering and hybrid recommendation methods and so far most of them have been able to resolve the problems while providing improved recommendations. However, due to information explosion, it is required to work on this research area to explore and provide new methods that can provide recommendation in a wide range of applications while considering the quality and privacy aspects. Thus, the current recommendation system needs enhancement for present and future requirements of better recommendation qualities.

References

[1] Arenas-García, J., Meng, A., Petersen, K. B., Lehn-Schioler, T., Hansen, L. K., & Larsen, J. (2007, August). Unveiling music structure via plsa similarity fusion. In

Machine Learning for Signal Processing, 2015 IEEE Workshop on (pp. 419-424). IEEE.

[2] Greg Linden, Brent Smith, and Jeremy York,"Amazon.com Recommendations Item-to-Item Collaborative Filtering", IEEE Computer Society 2003.

[3] Jun Wang, Arjen P. de Vries, Marcel J.T. Reinders," Unifying Userbased and Itembased Collaborative Filtering Approaches by Similarity Fusion", 2006 ACM.

[4] Panagiotis Symeonidis *, Alexandros Nanopoulos, Apostolos N. Papadopoulos,Yannis Manolopoulos," Collaborative recommender systems: Combining effectiveness and efficiency", 2007 Elsevier Ltd.

[5] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno," An Efficient Hybrid Music Recommender System Using an Incrementally TrainableProbabilistic Generative Model",IEEE 2008

[6] Akhmed Umyarov, Alexander Tuzhilin,"Improving Collaborative Filtering Recommendations Using External Data", 2008 IEEE.

[7] Zunping Cheng, Neil Hurley," Effective Diverse and Obfuscated Attacks on Model-based Recommender Systems" 2009 ACM.

[8] Hyeong-Joon Kwon, Tae-Hoon Lee, and Kwang-Seok Hong," Improved Memory-based Collaborative Filtering Using Entropy-based Similarity Measures", 2009 ACADEMY PUBLISHER.

[9] Asela Gunawardana, Christopher Meek," A Unified Approach to Building Hybrid Recommender Systems" 200X ACM

[10] Sergio Cleger-Tamayo, Juan M. Fern´andez-Luna, and Juan F. Huete, "A New Criteria for Selecting Neighborhood in Memory-Based Recommender Systems*",Springer-Verlag Berlin Heidelberg 2011.

[11] Pasquale Lops, Marco de Gemmis and Giovanni Semeraro," Content-based Recommender Systems: State of the Art and Trends", Springer 2011

[12] Yoav Bergner,Stefan Dr¨oschlery, Gerd Kortemeyer, Saif Rayyan, Daniel Seaton, and David E. Pritchard," ModelBased Collaborative Filtering Analysis of Student Response Data: MachineLearning Item Response Theory",2012.

[13] Svetlin Bostandjiev, John O"Donovan, Tobias Höllerer," TasteWeights:A Visual Interactive Hybrid Recommender System" 2012 ACM

[14] Jiankai Sun, Shuaiqiang Wang, Byron J. Gao, Jun Ma," Learning to Rank for Hybrid Recommendation", 2012 ACM

[15] Meenakshi Sharma and Sandeep Mann, "A Survey of Recommender Systems: Approaches and Limitations" ,ICAECE-2013.

[16] Royi Ronen, Noam Koenigstein, Elad Ziklik and Nir Nice," Selecting Content-Based Features for Collaborative Filtering Recommenders", ACM 2013

[17] Hendrik Drachsler, Hans G.K. Hummeland Rob Koper, "Personal recommender systems for learners in lifelong learning networks: the requirements,techniques and model".

[18] Mohammad Yahya H. Al-Shamri, Nagi H. Al-Ashwal," Fuzzy-Weighted Similarity Measures for Memory-Based Collaborative Recommender Systems", Journal of Intelligent Learning Systems and Applications, 2014.

[19] Michael J. Pazzani and Daniel Billsus," Content-based Recommendation Systems".

[20] Robin van Meteren and Maarten van Someren," Using Content-Based Filtering for Recommendation".

[21] Jaldert Rombouts, Tessa Verhoef," A simp le hybrid movie recommender system".