Ensemble Based Hybrid Recommender Systems

T. Prathima, B. Anjana, V. Apoorva, B.R.Sridhar

Abstract: ***In the past few years, the advent of computational and prediction technologies has spurred a lot of interest in recommendation research. Content-based recommendation and collaborative filtering are two elementary ways to build recommendation systems. In a content based recommender system, products are described using keywords and a user profile is developed to enlist the type of products the user may like. Widely used Collaborative filtering recommender systems provide recommendations based on similar user preferences. Hybrid recommender systems are a blend of content-based and collaborative techniques to harness their advantages to maximum. Although both these methods have their own advantages, they fail in 'cold start' situations where new users or products are introduced to the system, and the system fails to recommend new products as there is no usage history available for these products. In this work we work on MovieLens 100k dataset to recommend movies based on the user preferences. This paper proposes a weighted average method for combining predictions to improve the accuracy of hybrid models. We used standard error as a measure to assign the weights to the classifiers to approximate their participation in predicting the recommendations. The cold start problem is addressed by including demographic data of the user by using three approaches namely Latent Vector Method, Bayesian Weighted Average, and Nearest Neighbor Algorithm.***

Keywords: Bayesian Weighted Average, Cold start, Hybrid recommender system, Ensemble hybrid models, Latent Vector method, Nearest Neighbor Algorithm

I. INTRODUCTION

With technology taking leaps and bounds in every field the amount of data being generated has crossed several limits. By using elaborate systems that exist, this humongous data can be used to help users and organizations make meaningful and informed decisions. Recommender Systems are the tools which come into play in this arena. By analyzing the data present, such as the types of products purchased over a period of time, behavior patterns of consumers, genres of movies and shows watched, the recommender systems present new products to the user which he is most likely to be interested in.

Traditional approaches use the content-based methods which recommend products based on the product content and an individual users profile. The content of each product is drawn from a descriptor set which describes the products.

The content-based methods give exceptional results because the probability of a user to like a product which is very similar to other products he likes is relatively high. The only drawback is that in case of a scenario where the user has just entered the scene and there are no products he has liked or purchased, it is almost impossible for the system to recommend, this is called the cold start problem from the user scenario. In this work we aim to address the user cold start through three techniques.

Another approach is the collaborative filtering where in the recommendations are made by using the recommendations made to other people. It works on the principle that if there are two people who have similar tastes, the products liked by one will most probably be liked by the other person, this is also gives it the name- Social Filtering. The catch here is that if there is a new product that hasn't been bought or reviewed yet, it takes some time for the machine to recommend it to the prospective users, this is called as the product cold start.

To combine the benefits of both and also to overcome individual drawbacks Hybrid Recommendation Systems have taken over now, which are a blend of the collaborative and content-based recommender approaches and definitely outperform the individual fundamental methods respectively. In this work we discuss a weighted approach which is applied to a combination of three different prediction algorithms and their results are compared.

A. Objective

The objective of this project is to increase the accuracy of hybrid recommender systems by performing comparisons between a linear combination and a weighted combination of the best performing algorithms like SVD, NMF and KNN by using different error measures and as well as address the cold start problem by using demographic information of users with the help of three approaches- Latent Vector Method, Nearest Neighbor Approach and Bayesian Weighted Average.

B. Organization of the Paper

The paper is structured as follows: Section II deals with the previous work that has been carried out in this area. Section III highlights the dataset being used. Section IV focuses on the methodology that is applied and used in the aforementioned approaches. Section V gives an insight into the testing and results followed by Section VI which gives the conclusions and Section VII which gives the future scope.

II. LITERATURE SURVEY

Recommendation Systems (RS) have seen a lot of growth since the explosion of the internet and the increase of the amount of data. These can be classified broadly into content based, collaborative, and hybrid filtering [1] [2].

Revised Manuscript Received on January 5, 2020

Correspondence Author*

T. Prathima*, Assistant Professor, Dept. of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana, India. (E-mail:tprathima_it@cbit.ac.in)

B. Anjana, Dept. of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana, India. (E-mail: anjana.bytha97@gmail.com)

V.Apoorva, Dept. of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana, India. (E-mail: apoorva1196@gmail.com)

B.R. Sridhar, Assistant Professor, Dept. of Mathematics & Humanities, Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana, India. (E-mail: sreedharcbit@gmail.com)

Collaborative filtering is based on the idea of similar user profiles- that is, user who have similar tastes tend to like the same things. Here recommendations are provided to the user from the products that were liked by another user who has liked the same things as the first user. Collaborative filtering is of two kinds: memory-based and model-based [3]. Memory-based approaches work by using the complete database and applying statistical methods to find the closely related users or products. Whereas in a model-based approach the model is built from the data and further used for making predictions [4].

Content-based recommendation systems provide users with recommendations by using the information that is attributed to the product [5] [6]. Products that have many common attributes with the products the user has already like are preferred and recommended. Hybrid recommendation systems are a combination of content-based and collaborative methods which provide better results and accuracy [7].

Good amount of work has been done in the field of recommendation systems and in specific on movie recommendations. These are performed on the MovieLens data up to a large extent and the same has been used in this paper as well. Clustering Algorithms [8] are used in Hybrid recommender systems for finding the similarities among the users who have already rated products and new users [9]. Similarity measures like cosine similarity, Euclidean distance and correlation coefficients are used. The experiments were conducted on the MovieLens dataset, by two modified versions of clustering algorithm kmeans [10]. These methods resulted in increased accuracy which was shown by Root Mean Square values when compared to centroid-based collaborative filtering methods. The results were depicted by lower values of RMSE, regardless of similarity coefficients. In [11] the authors propose to predict user ratings using a hybrid approach by means of trust which is based on three factors namely emotions, rating deviation and review helpfulness and context of users which is based on the features: companion, day, place, and priority. The experiments were performed on the movie data sources such as IMDB [12], Rotten Tomatoes [13] and the performance of this system was analyzed using the absolute error and RMSE. It was found that the rating prediction based alone on trust gave better results when compared to user-based and product-based filtering methods. Even context-based collaborative filtering approach performed better when compared to standard collaborative filtering approaches.

Recommendation systems that don't just use the information present about the movies but that of the users as well tend to give better results. In [14] the users propose a recommendation system which provides recommendations based on the information available about the users by analyzing their profiles, their watching history and movie scores from other websites. They used the similarity aggregate measures. The authors thus concluded that the results obtained by this approach provide improved accuracy than the straightforward content-based and collaborative techniques.

There are many techniques that are used to make predictions for user ratings to recommend movies. This has given rise to a lot of literature comparing these techniques and identifying the ones that work best. [15] is one such work which compares different techniques such as baseline predictor, KNN [16], Stochastic Gradient Descent [17], SVD [18], SVD++, asymmetric SVD, integrated model and NMTF. These are applied on the MovieLens dataset and the RMSE error is used for the evaluation of all of these techniques. The authors conclude that the methods based on matrix factorization perform better than most of the others. In [19] the authors proposed a parametric rating prediction with contributions from content based, collaborative filtering and demographic filtering.

Cold start is another area where most recommendation systems fail to work. A cold start is a situation in which the system fails to provide recommendations when a user or a product which is new is brought into system. It is only after a while when the system 'warms up' that it can perform well. This is addressed in [20], an approach where the relationship of user feature-scores obtained from user-product interaction from the ratings to optimise the prediction algorithm's input parameters used in the recommender system which further improves correctness of the predictions when the user records were considerably less. These methods were experimented on the MovieLens dataset and it showed to greatly reduce the drawback in collaborative filtering with an accuracy increase of 8.4% when compared with the standard collaborative approach. In [21] the authors proposed a method for recommending products that brings together content and collaborative data into a probabilistic framework. They benchmark their work against a naïve Bayes classifier on the cold start problem and recommend products no one has rated yet.

III. DATASET

In this work we use MovieLens 100k dataset [22] which is a stable benchmark dataset consisting of one lakh ratings from thousand users on one thousand seven hundred movies. Each user has rated a minimum of twenty movies and these movies belong to nineteen different genres. The MovieLens dataset contains attributes like age, sex, occupation, region and also provides demographic information of the users.

IV. METHODOLOGY

A. Weighted Average Method

The methodology followed in this paper consists of two parts: the weighted average method and the approaches for cold start. First consider the three algorithms we used for the weighted average methods namely: Singular Value Decomposition (SVD), N-Matrix Factorization (NMF), K-Nearest Neighbors (KNN). Using these algorithms we formed a hybrid recommender system by generating weights by using the standard error measure and assigning to each of them. The architecture of the weighted ensemble method is as shown in the Fig. 1. First the MovieLens Dataset is taken and the predictions are made for the ratings by using SVD, NMF, and KNN algorithms. Then by using the measure of standard error the coefficient of variance is calculated which is used to compute the weights.

These weights are multiplied respectively to each of the predictions from the three algorithms and then the final rating prediction is formed by adding all of them. The recommendations made are evaluated by using

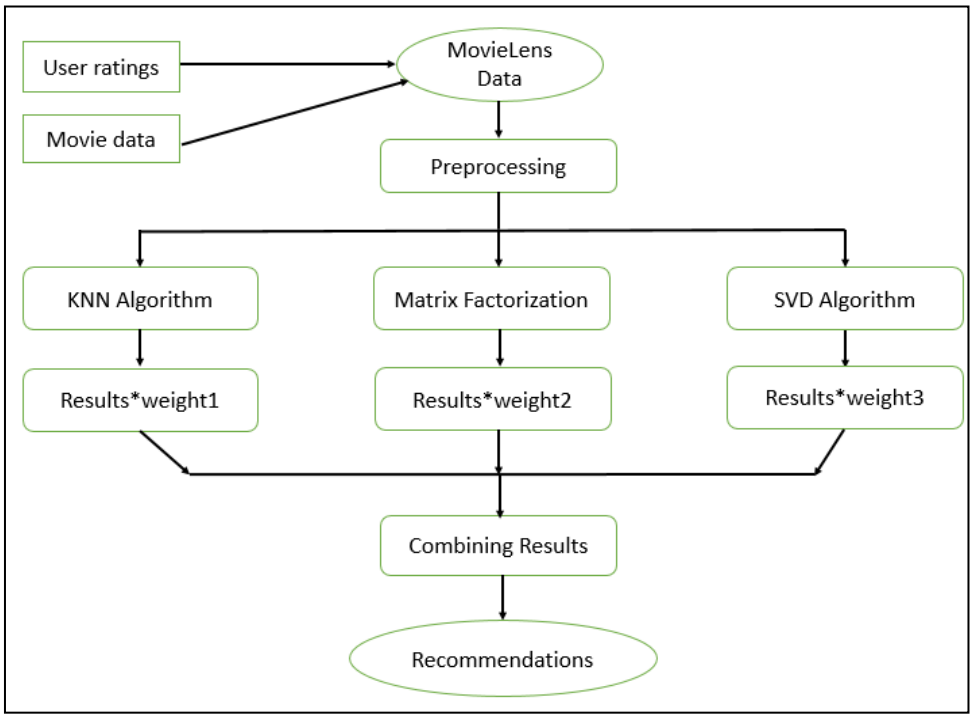

Fig. 1. Architecture of Weighted Model

the measure of RMSE, lower the RMSE value, the more accurate the model is.

The next section explains the algorithms used for predictions in brief followed by the approach for the computation of weights.

i. K-Nearest Neighbor Algorithm (KNN)

The KNN Algorithm is a simple and parameter less method used for classification. The algorithm determines the model from the data itself and no assumptions are made about the underlying data. KNN is also said to be a lazy algorithm which means that it does not draw any generalizations from the data and the training phase is minimal. In this approach all of the training data is used in the testing phase as well.

When a movie recommendation has to be made using the KNN algorithm, the distance between the target product and every other product in the data is calculated and the top K closest products are recommended.

ii. Non- Negative Matrix Factorization (NMF)

NMF is a matrix factorization method where we confine the matrices to be nonnegative. Factorizing a matrix V into two matrices W and H so that $V \approx WH$. Suppose that V consists of m rows $x_1,x_2,x_3,.....,x_m$, W consists of k rows $w_1,w_2,w_3,....,w_k$, H consists of m rows $h_1, h_2, h_3,.....,h_m$. Each row in V can be considered a data point. For instance, when decomposing an image, each row in V represents a single image, and each column is some feature. We can interpret x_i to be a weighted sum of some components, where each row of H is a component, and each row in W contains the weight of each component. Each row of the input matrix V is treated as a data point. In the case of NMF, we constrict the underlying components and weights to be non-negative. NMF decomposes each data point into an overlay of certain components. This reconstructed matrix serves as a basis to the recommendation. We can find attraction weight towards certain products in columns of the matrix. By sorting the values in descending order, we could determine which products should be proposed to the customer to match their preferences [23].

iii. Singular Value Decomposition (SVD)

SVD is the process in which a matrix is decomposed into three other matrices which is represented by the following formula:

$$X = USV^T \quad (1)$$

Where

- X : m x n matrix
- U : m x r orthogonal matrix
- S : r x r diagonal matrix
- V : r x n orthogonal matrix

The columns of U and V are called *left singular vectors* and *right singular vectors* respectively. We know that U and V are orthogonal, a matrix is said to be orthogonal if it satisfies the below equation:

$$U^TU = VV^T = 1 \quad (2)$$

Where I is the *identity matrix*. In an Identity matrix the diagonal elements are 1, with all other values being 0.

SVD reduces the dimensions of the utility matrix from its latent factors. Each user and product are mapped into a latent space with r dimensions. This way we can comprehend the users versus products in a better way. The main drawback of SVD is that there is little explanation as to why a product is recommended to a user [24].

iv. Predictions

Once the required data was obtained from the MovieLens dataset in the u.data file, which consists of the user id, movie id, actual rating given by the user, and timestamp, it was preprocessed and the timestamp column was removed as it was not useful in any way. After this, SVD, NMF and KNN algorithms were performed on it to predict the rating values. For the computation of weights the standard error measure was used that were assigned to each of the algorithms. Standard error can be given by:

$$SE = \frac{\sigma}{\sqrt{n}} \quad (3)$$

Where

- σ = standard deviation
- n = size (number of observations)

The Standard Deviation can be given by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}} \quad (4)$$

Where

- $\{x_1, x_2, \ldots, x_N\}$ are the observed values
- $\bar{x}$ is the mean value of the observations
- N is the number of observations in the sample

After finding the standard error, the coefficient of variance was found which is given by the formula:

$$cov = \frac{\text{standard error}}{\text{mean}} \text{ x } 100 \quad (5)$$

These values of covariance percentages were then used to calculate the weights assigned to ach algorithm which was given by:

$$\text{weight}_i = \frac{cov_i}{\text{sum of all covariances}} \quad (6)$$

The ratings predicted by the weighted approach can be given as:

$$\text{predicted rating}_i = (x_i \times \text{SVD pred}_i) + (y_i \times \text{NMF pred}_i) + (z_i \times \text{KNN pred}_i) \quad (7)$$

Where x, y and z are the weights calculated above

The weights are used to calculate the new predicted rating by multiplying each weight with the predicted value that is generated using that algorithm. Once these predicted weights are generated, we evaluate the RMSE values in two cases- one for the predicted rating of algorithm versus the actual rating and two, the weighted predicted values versus the actual rating.

B. *Approaches to Cold Start*

Cold start gets its name from a situation when a car engine fails to start initially on a cold morning but begins to function normally after a certain level of warmth is obtained. A similar scenario occurs in a recommender system when a previously unseen product or user is added to the system; this is where most recommendation systems fail. In cases where a new user is introduced in the system whose history is not known, it becomes close to impossible to make recommendations without knowing what the user has liked previously. This is called a User Cold Start. In cases where a new product is added to a system, it does not immediately occur in recommendations as there are not enough ratings for it but it slowly makes way after some people have rated it, this is called the Product Cold Start.

In this paper we address the User Cold Start the following approaches are proposed to provide recommendations in case of a User Cold Start scenario. They can be listed as follows:

i. Latent Vector Method
ii. Bayesian Weighted Average
iii. Nearest Neighbor Algorithm

i. Latent Vector Method

The Latent Vector Method is a method proposed in this paper to address the User Cold Start Scenario. It is carried out by using the Demographic data of the users which is available in the MovieLens Dataset. Demographic data stands for the information about the user like their age, sex, occupation and region. This data is used in identifying users similar to the new user, because intuitively people who are similar in age, sex, having the same working background are more probable to like the same movies. Vectors are computed from the demographic data and the movies and they are correlated to give the best recommendations. It is implemented as follows:

a. Preprocessing Demographic data
b. Generate the Most Liked Movie Genre matrix
c. Using Pearson's correlation coefficient to identify maximum correlated user
d. Picking the top three genres liked by this user
e. Suggesting top 5 movies from a combination of those genres

Step a: Preprocessing Demographic data

The demographic data of the users is available in the *u.data* file in the MovieLens 100k dataset. It consists of the user id, age, sex, occupation and zipcode. The zipcode is not useful in the approaches we apply and hence has been removed.

To enable the machine learning algorithms to perform prediction we need to convert qualitative variables into numerical variables using One hot encoding [19]. It gives a value 1 to the categorical variable that is present and a value 0 to all others. For example, in the MovieLens dataset on converting the categorical variable occupation by using One Hot Encoding, the occupation of the user becomes 1 and all other occupations becomes 0. In the case of age which has over a range of values we first bucketed them into 8 divisions, as the age ranged from 24 years to 73 years and dividing them into 8 buckets gave us almost equal samples in each, and then applied One Hot Encoding on the 8 buckets.

Step b: Generate the Most Liked Movie Genre matrix

The MovieLens dataset has a file *u.product* which consists of the movie id, movie title, release date and the number of genres it belongs to. All the columns that are not required are eliminated and only the movie id and the genres it belongs to are retained. Next, to compute the matrix of the genres which are most enjoyed by the user, the *u.data* file is taken which has the user id, movie id, actual rating given by the user and merge it with the above data file with respect to unique movie id so we get a list of all the movies one user has watched with the corresponding genres it belongs to.

And then all the genre values of the movies watched by one user are summed together which gives us a matrix of the user and all the corresponding movies that were liked by

him, the highest number in the genre indicating the genre he liked most.

Step c: Using Pearson's correlation coefficient to find most correlated user

In this neighborhood based approach users similar to the newly introduced user is determined and a prediction for the new user is considered. Pearson correlation coefficient has been used to compute the similarity. The ratings of person P and Q of the product k are written as P_k and Q_k, whereas $\bar{P}$ and $\bar{Q}$ are the mean values of their ratings. The correlation between P and Q is given by:

$$r(P,Q) = \frac{\sum_k (P_k - \bar{P})(Q_k - \bar{Q})}{\sqrt{\sum_k (P_k - \bar{P})^2 \sum_k (Q_k - \bar{Q})^2}} \quad (8)$$

Step d: Picking the top three genres liked by this user

After the most correlated user is picked by taking the top user after arranging the correlation values in descending order since higher the value, more closer the user is to the new user. The top three genres liked by him/her are picked by using the matrix mentioned in Step c.

Step e: Recommending top 5 movies from a combination of those genres.

Once the top genres are obtained, the top rated movies from a combination of these genres are picked. For example, if the top three genres are Romance, Mystery and Comedy, movies belong to each of the genres or a combination of the genres are picked.

ii. Bayesian Weighted Average

A conventional average method is easy to comprehend. Consider a bunch of students in a class and compute their average age. A newly admitted students age can reasonably be assumed from the average age of the class. If we go with weighted average, the weights are assigned to the ratings, where the weights of the lower ratings are less and the weights for the higher ratings are more. A weighted average is centered, around zero.

Unlike the approach in weighted mean where the values are moved close to 0, Bayesian average utilises the knowledge of the entire group to move the values closer to the average of a group. Consider a movie "A" with 100 ratings and the overall rating as 4.2 and another movie "B" with 1 rating and overall rating as 5. Based on these conditions one cannot predict that the movie with 4.2 rating is bad than movie with rating 5. Here the number of ratings is not being considered for the analysis of the best movie. So instead of considering the average of all ratings, one should consider the number of people rated it as well, and weights can be assigned to the movies while calculating average. This is done by the Bayesian weighted average method which calculates the overall weighted average of a movie.

Let there be n number of people in your dataset and m number of movies. The number of total ratings in the dataset is k then the Average overall rating of all the movies be R which is calculated by k/n. It is calculated by the following formula

$$S = wR + (1 - w)C \quad (9)$$

Where

S is the score given to the movie

R is the mean of user ratings for the movies

C is the mean of user ratings for all movies

w is the weight assigned to R and computed as $\frac{v}{v+m}$, where is the number of user ratings for that movie, and m is average number of ratings for all movies.

If relatively more populace rates a movie then its Bayesian average rating will be close to its original computed rating, otherwise, in the case of a new movie with fewer ratings Bayesian average rating will be closer to the average rating for all of a users movies.

c. Nearest Neighbor Algorithm

Many approaches exist on addressing the cold start problem, using the available demographic data like gender, preferences, hobbies, age and zip code is more appropriate than recommending same movies to everyone.

In this paper the categorical data or nominal data is converted to binary data using the method of one hot encoding as discussed previously. The zip code was removed because of the increase in the distance while calculating the Euclidean distance. The metric chosen is the Euclidean distance which is calculated by the formula:

$$d = \sqrt{\sum_{i=1}^{p} (v_{li} - v_{2i})^2} \quad (10)$$

Where

d is the distance between two vectors

v is the vector attribute

P is the total number of movies

The length of the vector is 24 (2+1+21) where 2 are gender, 1 is age, 21 are number of occupations. The user data is converted by one hot encoding to the form of vectors. The KNN algorithm can be used to solve classification and regression problems as well. The core of KNN algorithm presumes that things which are similar exist in close to each other.

The Nearest Neighbor Algorithm is implemented by calculating Euclidean distances from one user to all the other users in the dataset and by arranging then in the descending order, the top k users that match to the test user (new user) are collected. For example, take a new user who is of age 22, Male and a student then his latent vector would be of the form (0,1,22,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) and this is plotted in n-dimensional space and Euclidean distance between the remaining users in the dataset is computed.

After the distances are calculated to all the users now the nearest neighbors that match the persons demographic data can be drawn by considering the top k persons, in this paper the top 40 persons have been considered. This can suggest that the 40 persons think like the new person based on their demographic data. After sorting it was found that there are 23 students of age 22 and of gender male and have many common movies watched among them which gives evidence that the concept of demographic data can be used to calculate the recommendations in a cold start scenario. As we have the data of the movie ratings, taking the user ids of the user dataset and mapping it to the ratings data set and merging it with the movies dataset now gives all the movies that top 40 people have watched.

By using this method we recommend movies which have a high probability of

being liked by the user. Since our test data is taken from the original data itself by removing the movies they have watched, we conclude that if the movie that is recommended is present in the original database, which is a list of movies the user has watched, it is a hit.

Table-I: Estimated Linear and Weighted ratings

Index	User ID	Product ID	Actual Rating	SVD Rating	KNN Rating	Linear Rating	Weighted Rating
32236	1	1	5	3.836774	4.129971	3.983372	3.973045
23171	1	2	3	3.257877	3.193002	3.225439	3.227724
83307	1	3	4	3.631257	3.31734	3.474298	3.485355
62631	1	4	3	3.661979	3.785888	3.723934	3.719569
47638	1	5	3	3.09712	3.23935	3.168235	3.163225
5533	1	6	4	3.820044	3.800621	3.810333	3.811017
70539	1	7	4	4.713855	4.020766	4.36731	4.391723

V. TESTING AND RESULTS

A. Weighted Average Method

From the experiments conducted, the predicted values of ratings for each of the algorithms SVD, NMF, KNN, contrasted with the linear average and weighted approach are as shown in Table I. It can be seen that the values of the ratings predicted by the weighted approach are closer to the actual rating signifying that a weighted combination of algorithms works better.

From Table-II it can be seen that the RMSE values for the algorithms show that the weighted approach has the least value which means it is more accurate since lower the RMSE value of an algorithm the better it is.

Table-II: Estimated Linear and Weighted ratings

Algorithm	RMSE Score
SVD	0.936
NMF	0.963
KNN	0.980
Linear	0/732
Weighted	0.727

B. Results for Cold Start Approaches

i. Latent Vector Method

The most correlated user can be seen to be user 3 as this user has the highest correlation value as shown in Table-III. The order of the movie genres liked by him in descending order are as shown, and the top three can be seen to be Drama, Thriller and Comedy.

Table-III: Highest correlation value showing the most correlated user

User ID	Correlation Score
2	0.858800
15	0.790621
24	0.777400
25	0.776455
32	0.771355

ii. Bayesian Weighted Average

After generating the ratings as shown in Table-I, the ratings are sorted according to the score and the top movies can be recommended to the new user. Filters can also be applied to filter out specific genre like “Comedy”, “Action”, “Romance”, “Adventure” and the top movies from the resulting set can be given as recommendations. Note that the only limitation of this Bayesian-based scoring technique is that it assumes that the movie ratings come from a normal distribution centered at their average. The results in Table-IV are the outputs for the recommendations and after sorting, the results are shown in Table-V.

Table-IV: The Bayesian Weighted Averages of the Movies in the Movie Lens Dataset

Movie ID	Score
1	3.837484
2	3.307809
3	3.231944
4	3.545694
5	3.395833
6	3.544088
7	3.762813
8	3.895310

Table-V: Top movie Recommendations to the user after Bayesian method

Movie id	Score
318	4.309474
64	4.285107
50	4.281169
483	4.273240
12	4.228720
603	4.196251
98	4.188426

iii. Nearest Neighbor Approach

A list of movies the users have watched is obtained and merged to count the number of common movies among the people. As the ratings data is available as well, the user ids of the user dataset is taken and mapped to the ratings data set. It is then merged with the movies dataset which now gives all the movies that top 40 people have watched. For example, for a single user it was seen that more than 900 movies were recommended to the new user and he has watched 79 according to the original dataset. It was observed that out of these, 73 movies have matched our recommendation list and the accuracy is 93.6% which is obtained by taking a percentage value of 73 of 79 movies. The result is shown below in Table-VI:

Table-VI: Accuracy of 30 users using K Nearest Neighbors method

User ID	Recommendations	Actual Watched	Hits	Accuracy (%)
787	954	117	110	94.017
910	935	26	23	88.461
732	961	83	78	93.975
447	949	36	35	97.222
601	1064	89	87	97.753
791	957	231	215	93.073

VI. CONCLUSION

In this work we empirically evaluated a new ensemble method to predict the movie ratings by using SVD (Singular Value Decomposition), NMF (Non matrix factorization), KNN (K-Nearest neighbors). Two approaches for combining the three algorithms were proposed- the linear approach and weighted approach, which on observation from the results showed that the weighted approach gave better results as compared to the regular linear methodology. These approaches gave better results when RMSE is taken as the Key Performance Indicator (KPI). In this work we also addressed diverse approaches to deal with the problem of cold-start. Three approaches to recommend movies to the new users were proposed. The first is the Latent Vector method that uses Pearson's correlation to find the top probable genres that are liked by the user which can be mapped to predict movies, next the Bayesian average method that recommends top movies available in the database, the third approach is the KNN approach which finds the top nearest neighbors that are similar to new user and intersects the movies that are watched by them. Any of the approach can be used to recommend movies to a new user in the system.

VII. FUTURE SCOPE

Recommender system is a powerful technology to get additional value for a business from user databases. They assist the users by allowing them to find products they like. The proposed ensemble approach is not only confined to the algorithms used in this paper, the same approach can be used on different algorithms as well. The cold start approaches proposed in this work can be implemented to deal with the product cold start if the metadata of the product is available. This algorithm can be implemented in different domains such as recommending shopping products in e-commerce, suggesting optimal loans and schemes to customers in banking, predicting the problem based on various symptoms in healthcare and so on. The proposed approach has been implemented and evaluated using MovieLens dataset and can be implemented on large and other datasets like Netflix and Film Affinity.

REFERENCES

1. M. Pazzani, A framework for collaborative, content-based, and demographic filtering, Artificial Intelligence Review-Special Issue on Data Mining on the Internet 13 (5-6) (1999) 393–408.
2. Burke, Robin. (2002). Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction. 12. 10.1023/A:1021240730564.
3. Schafer, Ben & J, Ben & Frankowski, Dan & Herlocker, Shilad & Sen, Shilad. (2007). Collaborative Filtering Recommender Systems.
4. Xiaoyuan Su, Taghi M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in Artificial Intelligence archive, 2009.
5. Zisopoulos, Harry & Karagiannidis, Savvas & Demirtsoglou, Georgios & Antaris, Stefanos. (2008). Content-Based Recommendation Systems.
6. Aggarwal, Charu C. (2016). Recommender Systems: The Textbook. Springer. ISBN 9783319296579.
7. Burke, Robin. (2002). Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction. 12. 10.1023/A:1021240730564.
8. Xu, D. & Tian, Y. Ann. Data. Sci. (2015) 2: 165. https://doi.org/10.1007/s40745-015-0040-1
9. Kuzelewska, Urszula. "Clustering Algorithms in Hybrid Recommender System on MovieLens Data." (2014).
10. Byström, Hans. "Movie Recommendations from User Ratings." (2011).
11. Kumar Sejwal, Vineet & Abulaish, Muhammad. (2019). Trust and Context-based Rating Prediction using Collaborative Filtering: A Hybrid Approach. 1-10. 10.1145/3326467.3326491.
12. https://www.imdb.com/interfaces/
13. https://data.world/datasets/rottentomatoes
14. Chiru, Costin-Gabriel et al. "Movie Recommender system using the user's psychological profile." 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP) (2015): 93-99.
15. Bao, Zhouxiao and Haiying Xia. "Movie Rating Estimation and Recommendation." (2012).
16. B. Sarwar, G. Karypis, J. Konstan and John Riedl, Item-Based Collaborative Filtering Recommendation Algorithms, Proceedings of the 10th international conference on World Wide Web 2001: 285-295.
17. Bottou L. (2010) Large-Scale Machine Learning with Stochastic Gradient Descent. In: Lechevallier Y., Saporta G. (eds) Proceedings of COMPSTAT'2010. Physica-Verlag HD
18. Wall, Michael E., Andreas Rechtsteiner, Luis M. Rocha."Singular value decomposition and principal component analysis". in A Practical Approach to Microarray Data Analysis. D.P. Berrar, W. Dubitzky, M. Granzow, eds. pp. 91-109, Kluwer: Norwell, MA (2003). LANL LA-UR-02-4001.

19. B. Chikhaoui, M. Chiazzaro and S. Wang, "An Improved Hybrid Recommender System by Combining Predictions," 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications, Singapore, 2011, pp. 644-649.
20. Uyangoda, Lasitha Ahangama, Supunmali, Ranasinghe, Tharindu. (2019). User Profile Feature-Based Approach to Address the Cold Start Problem in Collaborative Filtering for Personalized Movie Recommendation.
21. Schein, Andrew I. et al. "Methods and metrics for cold-start recommendations." *SIGIR* (2002).
22. https://grouplens.org/datasets/movielens/100k/
23. X. Luo, M. Zhou, Y. Xia and Q. Zhu, "An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems," in *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273-1284, May 2014.
24. Golub G.H., Reinsch C. (1971) Singular Value Decomposition and Least Squares Solutions. In: Bauer F.L. (eds) Linear Algebra. Handbook for Automatic Computation, vol 2. Springer, Berlin, Heidelberg

AUTHORS PROFILE

Ms. T. Prathima, is an Assistant Professor in Chaitanya Bharathi Institute of Technology (CBIT),Hyderabad with M.Tech in Computer Science, her areas of research include Machine Learning and Data Analytics.

Ms. B. Anjana, graduated from the Dept. of Information Technology, Chaitanya Bharathi Institute of Technology. Her research interests include Machine Learning.

Ms.V.Apoorva, graduated from the Dept. of Information Technology, Chaitanya Bharathi Institute of Technology. Her research interests include Machine Learning.

Dr. B. R. Sreedhar, is an Assistant Professor in statistics of Department of Mathematics with 20 years of teaching experience for under Graduate and Post Graduate Courses . He did his Ph.D from Sri Krishna Devaraya University, Anathapur, with the specialization of "Reliability Theory". His Research Specializations are: Analysis of Reliability indices, fitting of Probability Distributions of Agricultural Data, Rainfall Data, Earthquake data and Data Envelopment Analysis (DEA) for Engineering Application