INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD ISSN: 2455-0620 Volume - 5, Issue - 8, Aug – 2019
Monthly, Peer-Reviewed, Refereed, Indexed Journal with IC Value: 86.87 Impact Factor: 6.497
Received Date: 20/08/2019 Acceptance Date: 30/08/2019 Publication Date: 31/08/2019

Movie Recommender with Crow search and K-means using Python

[1] Eshetu Tesfaye, [2] Pooja, [3] Getaneh Ashenafi, [4] Abriham Atinafu
[1]M. Tech Scholar, [2]Associate Professor, [3]M.Tech Scholar, [4]M.Tech Scholar
Computer Science and Engineering, School of Engineering and Technology
Sharda University, Greater Noida, India
Email: [1]eshe384@gmail.com, [2] pooja.1@sharda.ac.in, [3] getaneh215@gmail.com, [4] abriham2009@gmail.com

Abstract: *Recommender frameworks are information filtering engine that seek to foresee the rating for clients and things, dominatingly from enormous information to suggest their preferences. Movie recommendation systems provide a mechanism to help users in arranging users with comparable interests. This makes recommender frameworks basically a focal piece of sites and internet business application. In this study, we have developed a scheme for a movie recommendation system named collaborative movie recommender system using crow search and K-means algorithm. This article centers on the movie suggestion proposal frameworks whose essential goal is to recommend a recommender framework through information bunching and computational insight. We have used Elbow method and Silhouette score to select right k number of clusters and calculate errors in each cluster respectively. To evaluate the performance of the proposed system we have used evaluation metrics such as standard deviation (SD), mean absolute error (MAE), root mean square error (RMSE). The experiment result shows 0.635 MAE and 0.758 RMSE which indicates that, our system achieved better performance compare to other existing methods [12, 14, and 16].*

Key Words: *Crow search optimization, E-commerce, K-means, Recommendation system, Movie.*

1. INTRODUCTION:

With the broad improvement of web advances and web-based business destinations in the previous years, an expanding number of online services getting to be well known, for example, Google News and Yahoo News for perusing news stories, Netflix and YouTube for watching recordings, and so on [2]. One other well-known utilization of the WWW is for web-based shopping, where the purchasing and selling of items and administrations are directed electronically [1, 9]

These online services lead to an expansion in the measure of data on the web which is called data over-burden issue. In this way, clients need to invest considerably more energy to locate their fascinating things among an enormous number of decisions. Recommender frameworks have demonstrated to be valuable methods to lead with this issue and can help the clients in finding their pertinent things in a sensible time. Recommender software filters the data using various calculations and prescribes the most applicable things to clients [6]. The primary thought behind the recommender framework is to utilize clients' past inclination to foresee future interests of clients. The level of clients' fulfillment relies upon the nature of results given by recommender frameworks [2]. Thusly, building up a ground-breaking method is a significant issue to improve the presentation of a recommender framework.

Recommender procedures can be classified into three fundamental gatherings including content-based, shared based, and mixture based techniques. However, the most widely used technique to build recommendation systems is a collaborative filtering method [3, 7].Every suggestion methodologies have its very own confinements. For instance, CB has over specialization issue, while CF has sparsity and cold start issues. The cold start problem is related to the situation when a user (item) in the system has expressed (received) a few number of ratings [15, 13, 17]. On the other hand, the data sparsity problem is related to sparsity of ratings that recommender system face, since the number of items is usually millions of users can provide ratings for small portions of the items [13, 4]. These issues happen when accessible information in the frameworks is lacking for distinguishing comparable clients or things as neighbors set. As it were, these issues happen when there is no crossing point at all between two clients or things dependent on accessible evaluations and subsequently comparability measure isn't good in any way. Notwithstanding when the calculation of likeness measure is conceivable, it might be is solid esteem as a result of deficient data handled.

The rest of this article is arranged as pursues: Section 2 gives a short clarification of the related work that was carried out on collaborative recommendation system and clustering based collaborative recommendation. The proposed methodology called as a k-mean-crow search approach for movie recommender framework is clarified in Section 3. In Section 4, experiment result performed on Movie lens dataset are described, and lastly summarizations of this article with future work are highlighted in section 5.

2. LITERATURE REVIEW:

In this chunk, clustering and optimization algorithms used for recommendation engine and the hybridizations of optimization algorithms with clustering algorithms are discussed in detail.

Several papers on recommender system surveys have been published in the last decades in order to analyze major problems of traditional recommender systems [3, 5]. The most widely used technique to build recommender systems is collaborative filtering [3, 7, and 9]. However, the issue of scalability and sparsity are faced by traditional collaborative filtering algorithms, there have been various approaches proposed to address this issues. Nitin pradeep and Zhenzhen [9] proposed hybrid user-item based CF methods to achieve a more personalized product recommendation for a user while addressing the traditional issues of data sparsity and scalability in collaborative filtering algorithms. They used case based reasoning (CBR) and average filling for sparsity reduction of the user-item matrix. For the sparsity issue, they used average filling method to fill the vacant cells in the matrix by the help of Euclidian distance similarity measure and complimented self-organizing map (SOM) with genetic algorithm (GA) optimization for user clustering to solve scalability problem. In 2018, Md. Akter Hossain and Mohammed Nazzin Uddin developed a neural engine for movie recommender system using artificial neural network (NN). The results they were obtained using a single NN was, MAE=3.92, MSE=6.02, and MRE=9.12%. Their simulation results show that their system achieved better performance compare to other methods. On the other hand, Vimala Vellacichamy and Vivekanandan Kalimuthu [20], have also proposed a model based collaborative recommender system to reduce the data sparsity and scalability issues. They apply FCM (Fuzzy C-Mean) clustering technique to cluster users into different groups and Bat optimization to obtain the initial position of clusters. According to [20], Fuzzy Bat Clustering collaborative recommender system is performed in two phases. In the first phase, Fuzzy C-Means method clusters the users into different groups based on their past history of ratings. Bat optimization is then used to find the optimum cluster center points for FCM. It provides better result in optimization than other optimization methods. For measuring the accuracy of the proposed recommender system, [9] they used MAE (Mean Absolute Error) as a statistical accuracy measure. According to their experimental result, MAE of traditional Item based CF is 0.22 and MAE for they proposed is 0.15 for each of 5-fold validation. This indicates their proposed method showing better prediction sensitivity and better prediction quality than the traditional item based CF algorithm.

Cluster analysis is one of the popular data mining techniques for knowledge discovery and it is defined as the process of grouping similar data. Clustering technique is widely used in machine learning, image segmentation, data compression, pattern recognition, statistical data analysis. In general [1, 10], the algorithms used in clustering methods are divided into two categories: hierarchical and partitioning. K-means is one of the clustering algorithms to cluster the numerical data. The features of K-means clustering algorithm are easy to implement and it is efficient to handle large amounts of data. The major problem with K-means is the selection of initial centroids. It selects the initial centroids randomly and it leads to a local optimum solution. Recently, nature-inspired optimization algorithms are combined with clustering algorithms to obtain the global solutions. Rahul Katarya and Om Prakash Verma [12] applied hybrid of K-means and cuckoo search to the movie lens data set to achieve an improved movie recommendation system. They followed two steps; initially, they applied K-means clustering algorithm to movie lens data set for clustering of users into different clusters. That means, clusters are selected randomly at first then users are inspected one by one by calculating the differences in their rating and the centroid of the cluster and if there difference is smaller, then the user gets allocated to the cluster to which they are closest. Next they applied Cuckoo search optimization algorithm to the resultant of the K-Means algorithm for optimizing the results. Their limitation was if the initial partition does not turn out to work well then efficiency may decrease. The K-means clustering algorithm is easy to implement and efficiently handles large datasets. The main drawback is that it produces local optimum solutions. To obtain the global optimization solution, K Lakshni, N Karathikayani and S shanthi [6] combined a metaheuristic global optimization algorithm with K-means. Generally, in Genetic algorithm, three operators namely selection, crossover and mutation need to be applied [6]. CSA needs only two parameters AP and FL. Each optimization algorithm has its own parameters and it is tedious to fix the optimum values for each parameter.

3. PROPOSED METHODOLOGY:

To overcome the limitations of a collaborative recommender system, we proposed a hybrid cluster and optimization based technique to improve movie prediction accuracy. Our motive is to build a cooperative model solution that incorporates user ratings from the movie lens dataset for predictions. Crow search algorithm (CSA) is a new population based met heuristic optimization algorithm. This algorithm is based on the intelligent behavior of the crows. In this paper, CSA is combined with the K-means clustering algorithm and applied to collaborative movie data set to obtain the global optimum solution. Fig 1 below illustrates the framework of the proposed system as follow:

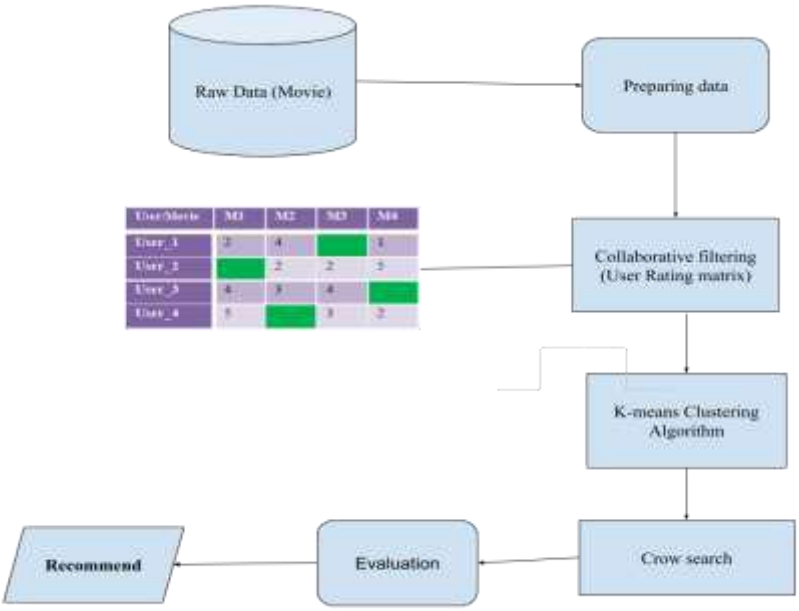

Fig. 1 Framework of the proposed system

A. K-mean Algorithm

Clustering method is popular data mining technique; currently famous in the fields of image processing, machine learning, pattern recognition, and other specific fields like deep learning. K-means is the process of organizing related data together. It is the most widely used and easy to implement clustering algorithms to cluster the datasets. The features of K-means are the selection of initial cantroids. It chooses the initial centroid randomly and it provides a local optimum solution. Initially, the clusters are selected randomly as a centroids and the difference between clusters and users are calculated. Then distance of users are compared to all centroids and grouped according to the smallest distance from any cluster's mean. The main function of k-means is to reduce sum of intra-cluster distance calculated as squared error function using Eq. (1) below.

$$\sum_{j=1}^{k} \sum_{i=1}^{N} \|Xi(j) - Ci\| \ldots\ldots (1)$$

A rating matrix consists of N number of data objects Xi, I = 1, 2, 3 …N with D number of movie types as a features. Dj, j=1,2,3, ……D

The K-means algorithm is made of the following steps given below [6, 10, 12, and 15].

1. Choose k number of points as Cj, j=1, 2... K from dataset.
2. Find the distance from each dataset points to k centroids using Eq. (2) below.

   Dif (Xi, Cj) = (∑(Xi - Cj)2)1/2 …….(2)
3. Assign the data points to the cluster with smallest distance.
4. Refresh the centroids by taking average of the cluster.
5. Repeat step 3 and 4 till the centroids no longer move or maximum number of iteration is reached.

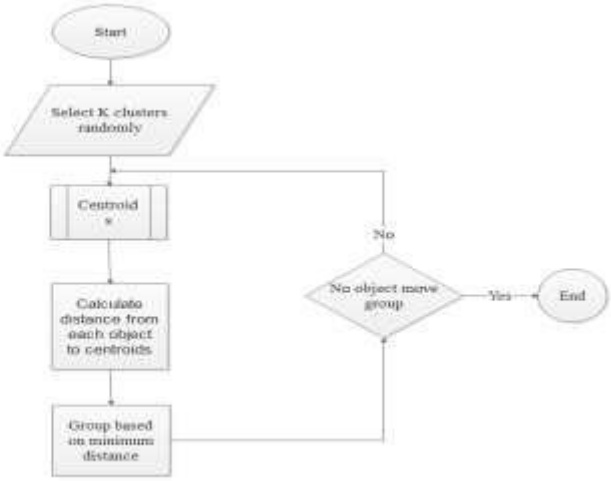

Fig. 2 Flow chart of K-means clustering

B. ***Crow Search Algorithm***

Next CSA is applied to the outcome of the k-means for optimizing the centroids. CSA is popular based metaheuristic algorithm, which is based on modeling the intelligent behaviour of crow [12]. It was proposed by Askarzadeh in 2016 for solving optimization problems [8, 22]. CSA endeavors to impersonate the social knowledge of crow rush and their sustenance get-together procedure. Crows are a generally conveyed family of winged creatures, which have been credited with knowledge all through fables. Curiously, a crow individual tends to take advantage of the nourishment asset of different species, including the other crow individuals from the herd. Indeed, each crow endeavours to conceal its abundance sustenance in an alcove spot and recover the put away nourishment in the desperate hour. The basic concept of crow search algorithm is each crow individual scans the decision space for hideout with the best assets.

In the standard CSA [8, 19, 22], the group of crows spread and quest throughout the decision space for perfect hideout spots. It includes three successive stages. Initially, the situation of each crow is made arbitrary then that position is initialized as the best hid spot memory of each crow. Next, a crow assesses the quality of its situation according to the objective function. Lastly, the crow arbitrary chooses one of the groups crows and tails it to get the position of the foods covered up by that crow. In the event that they found the situation of the nourishment is delicious, the crow refreshes its position. Something else, the crow remains in the present position and does not move to the created position [22, 19].

The standards of CSA are accompanying:

i. Crows live as gatherings.
ii. Keep in mind the situation of nourishment concealing areas.
iii. Searching the food source of the others members: and
iv. Protecting their food source.

The group of crows spread and search throughout the choice space for the perfect food source. The number of crows called flock size is assumed as P in the search space and the position of the crow at iteration time I in the search space as X(I, t), where I = 1,2,3,.....N; t=1,2,3,.....max_iter; where max_iter is the maximum iteration time. Each crow has a memory M to remember the position of food source. At the tth iteration , the position of the hideout spot of the it^{h} crow individual is represented by M(i,t) and it shows the best position obtained so far.

The flow steps of CSA is explained as follow;

1. Set the parameters such as flock size P, It_max, flight length FL, and Awareness probability AP.
2. Set the position of crows arbitrary in PD-dimensional choice space.
3. Initialize the memory of the crows with position of crows.
4. Check the position of the crows
5. While **t < max_iter**
   a. For all crows
   i. assume crow I follows crow j
   ii. If crow j don't sense that crow I is tailing it, new position of i is obtained using Eq. (3); if crow j senses that crow I is tailing it, position of I is randomly obtained;

$$X(I,t+1)= X(I,t) + ri * fl(I,t) * [m(j,t) - x(I,t)] \quad Rj >=AP(j,t)$$
$$\text{Otherwise } X(I,t+1) =random \quad \ldots\ldots\ldots(3)$$

   iii. Check the practicality of the new position; if the new position of crow is attainable, its position is refreshed; something else, the crow remains in the present position.
   iv. Check the new position of the crow using Eq. (3).
   v. Refresh the memory of the crows using Eq. (4) below.

$$M(I,t) = \begin{cases} x(i.t+1) & \text{if } f[x(I,t+1)] > f[m(I,t)] \quad \ldots..(4) \\ M(I,t) & \text{otherwise} \end{cases}$$

6. End of while loop.

In which, R_j is a random number distributed uniformly within the range of [0, 1]; and AP (j, t) is the awareness probability of the jth crow at the tth iteration.

C. *K-means-Crow search based collaborative filtering framework*

The K-Means is clustering algorithm whose main goal is to group similar data points into cluster which is simple to build and handles enormous data effectively. Its limitation is it produces local optimal solutions. To obtain the global solution and improve the performance of recommender system we combined K-means with global optimization algorithm CSA and apply to rating matrix dataset. The proposed collaborative recommendation system with K-means and crow search is described as follow:

1. Firstly, prepare your data as pivot table (users and movie types as row by column respectively). So your data will looks like this below.

Index	(no genres listed)	Action	Adventure	Animation	Children	Comedy	Crime
1	nan	4.32222	4.38824	4.68966	4.54762	4.27711	4.35556
2	nan	3.95455	4.16667	nan	nan	4	3.8
3	nan	3.57143	2.72727	0.5	0.5	1	0.5
4	nan	3.32	3.65517	4	3.8	3.50962	3.81481
5	nan	3.11111	3.25	4.33333	4.11111	3.46667	3.83333
6	nan	3.60938	3.89362	4.07143	3.61702	3.37008	3.28571
7	nan	3.25781	3.31481	3.39286	3.2	3.16327	3.30769
8	nan	3.33333	3.54545	5	4.25	3.20833	3.88889
9	nan	3.125	3.8	4	4	3.66667	3.14286
10	nan	3.5	3.58065	3.86667	3.60714	3.26582	3.11538
11	nan	3.54348	3.55556	nan	nan	3.41667	3.69231

Fig.3 prepared data in pivot table format

2. Set the values of flock size N as total users, movie types as pd, number of clusters K, maximum number of iterations max_iter, flight length FL, and awareness probability AP as tolerance in k-means algorithm.
3. Set the position of crows N and memory of crows M. while initializing the memory of the crows, set the memory of the crows with the values of the position of the crows because initially crows hid their foods at their initial positions.
4. Check the fitness of initial position of crows using squared error function (Eq. (1)).
5. Set the fitness of memory of the crows with the fitness position of the crows.
6. Refresh the position of the crows:
7.
   A. For **t < max_iter**
   I. For all crows
      a. Assume crow I follows crow j
      b. If crow j doesn't sense that crow I is tailing it, new position of I is obtained using Eq. (3)
      c. If crow j senses that crow I is tailing it, position of I is obtained randomly.
      d. Check the practicality of the new position; if the new position of crow is attainable, its position is refreshed; something else, the crow remains in the present position.
      ii. End of For
   B. Evaluate the fitness position of the crow using Eq. (1)
   C. Refresh the memory of the crows using Eq. (4).
8. Obtain best solution.

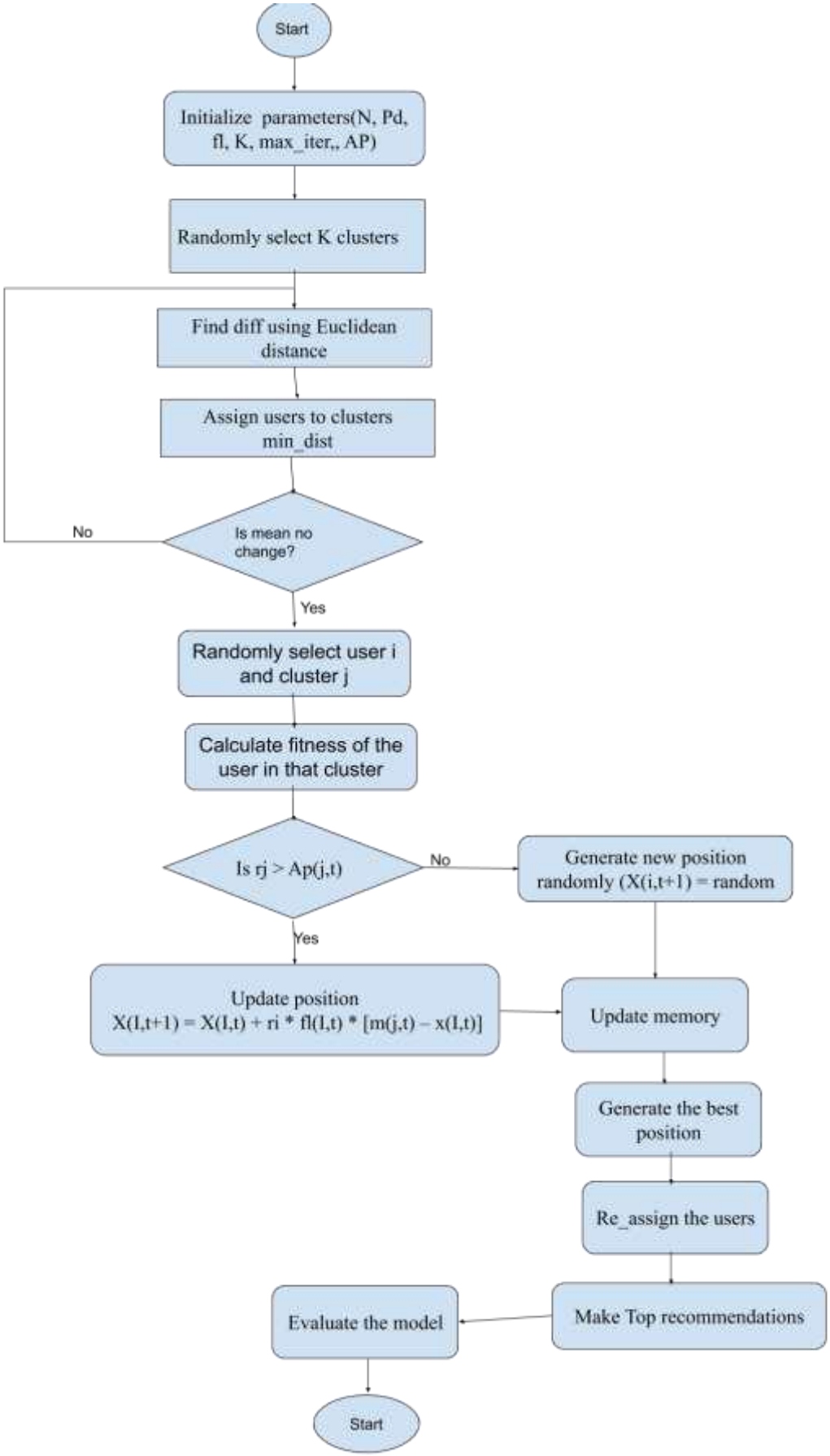

Fig. 3 flow chart that illustrates the process involved in the proposed framwork

```
1. Initialization step
N = no_of users (represents flock size (number of cros)),
Pd = no_of_movies or total movie types(represents
D-Dimensional environment)
K = number of clusters,
max_iter = maximum iteration,
fl = Flight length
AP = Awarness propability
M = memory represents each crow has a memory M to
 remember the position of hiding place (indirectly represents
centroid or cluster)
M(i,t) = crow's memory position initially
Users _in_container[N] = total number of users before
grouping
ungrouped_users =number of users without assigning to any
cluster (represents users left in the container)

2. Calculate Euclidian distance  and assign to closest cluster
     min_dist = max_it
     for each cluster J
     diff =Euclidian distance between user and cluster
      if diff < min_dist
       min_dist = diff
       min_inedx = j
     assign cluster min_index to i

3. find average rating for elements with in clusters
     while t > 0
      for each cluster i
        for each movie i rated by j
          for each user j belonging to i
           calculate mean rating for each movie

4. Repeat step 2 and 3 until termination conditions (no more
mean change, reaches maximum iteration)
     for each user i
      for each cluster j
       calculate diff(i,j)
        Assign user to cluster with min_diff

5. Apply CSA
     #from each user remaining in the container
      Randomly select user i
       Randomly select a cluster j
        calculate the fitiness of the user in that cluster
          if fitines[i,j] >no_of_elements_in_cluster(j)*including
factor
6.        m[N] =0, i=0
           while i < N
            if rj >= AP(j,t)
              X(I,t+1) = X(I,t) + ri * fl(I,t) * [m(j,t) – x(I,t)]
             else
               X(I,t+1) = random; #xnew(i,j)=l-(l-u)*rand; where l=0
(lower rating value) and U =5 (upper rating value)

7. find best memory (predict the user to the best cluster
              if f[x(I,t+1)] > f[m(I,t)]
                M(I,t) = x(i.t+1)
                    else
                  M(I,t) = M(I,t)
8. Re_cluster again
8. make recommendation
9. Calculate the predicted rating by each user
10. evaluate the accuracy of the model
```

Fig. 4 Pseudo-code of proposed framework

4. EXPERIMENT RESULTS AND ANALYSIS:

In this section, we are discussing data types, the process and steps involved in the data processing, outcome result analysis, and various metrics used to evaluate the performance of the proposed system framework. The goal of this paper is to find out similarities within groups of people in order to build a movie recommending system for users using crow search algorithm. We analyzed a movie dataset that we get from Group lens link to explore the characteristics that people share in movies' taste, based on how they rate them.

A. Datasets

We use movie dataset to assess the performance of our system. This dataset are collected from Movie Lens web site (http://movielens.org). This data has two files movies.csv and ratings.csv we used for the analysis. The dataset contains 100,836 ratings and 3,600 tag applications applied to 9,742 movies by 600 users. The ratings for the movies

are in the range of 0 to 5 and fig 6 rating matrix below describes ratings given by users initially as a sample and fig 7 shows movie data collected from Group Lens web site respectively.

Index	userId	movieId	rating	timestamp
0	1	1	4	964982703
1	1	3	4	964981247
2	1	6	4	964982224
3	1	47	5	964983815
4	1	50	5	964982931

Fig. 5 rating data

Index	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

Fig. 6 Movie data

B. Evaluation method for choosing right K number of clusters

Unlike supervised learning where we have the ground truth to assess the model's performance, clustering algorithms doesn't have a strong assessment metric that we can use to assess their result. Also, since K-means requires K as information and doesn't take in it from information, there is no correct answer regarding the number of groups that we ought to have in an issue. So, to find right number of clusters we used Elbow method. To calculate error in each clusters we used silhouette analysis method and the result we have obtained from our experiment looks like this below;

No. Of clusters (K)	Silhouette eror score
2	0.4112
8	0.4182
14	0.4028
20	0.4055
26	0.3807
32	0.3762
38	0.36
44	0.3473
50	0.3535
56	0.3431
62	0.3288
68	0.3163
74	0.2913
80	0.2573
86	0.2031
92	0.1719
98	0.1479
104	0.1039
110	0.0541

Fig. 7 Silhouette error score for various values of k number of clusters between 2 and 110

From the above figure 6, we have calculated error values for all k values we are interested in. totally we have 610 users who have rated the movie then we selected three movie as a sample and optimized it. Finally we obtained 110 users because most of users have rated with the same values so we biased the data. Fig.7 illustrates each value of k vs. the silhouette error score at that value.

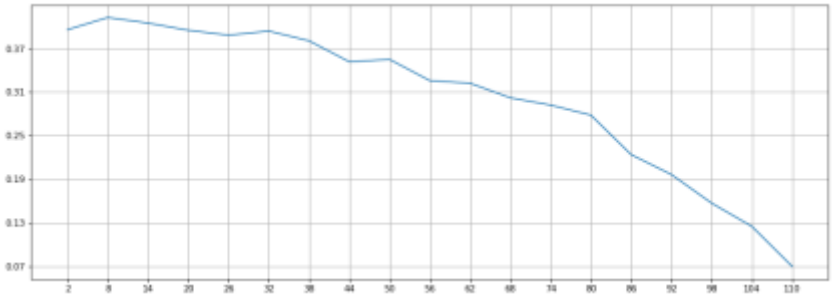

Fig. 8 Each value of K vs. Silhouette score error

The above graph clarifies that good choice for k are 8, 14, and 32 amongst other values. Therefore, the result shows that as we increases the number of clusters, total number of users in each cluster become less and similarities among them become high, then we results in worse clusters.

C. Prediction step

To predict unrated movie for the user, we pick user id from one cluster and find average of the votes for the movies in that cluster; that would be a prediction rate value for the user would enjoy.

According Elbow method, we pick *k* at the spot where sum of squared distance between data points and their assigned clusters' centroids starts to flatten out and forming an elbow. So, from fig.7, we choose k = 8 which indicates we will have eight number of clusters. As example we picked cluster number =3 and calculated the rating value for unrated movies in that cluster by taking means of rating values of users in that cluster which results 4.264705882352941for movie name "Pulp Fiction (1994)" as prediction rating value.

D. Recommendation step

For recommending movies to the user, firstly we picked a user Id =4 and get all this user's ratings shown below by fig. 8 and separately selected which movie did he not rated which is illustrated by fig. 9 below. Then we calculated the ratings of those unrated movie by taking mean of which cluster they are found and put in order which is suggested to the user for recommendation. Fig. 10 below show 10 most movies recommended for user Id 4.

Index	
Star Wars: Episode IV - A New Hope (1977)	5
Star Wars: Episode V - The Empire Strikes Back (1980)	3
Star Wars: Episode VI - Return of the Jedi (1983)	5
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)	nan
Silence of the Lambs, The (1991)	5
Pulp Fiction (1994)	5
Fargo (1996)	nan
E.T. the Extra-Terrestrial (1982)	4
Back to the Future (1985)	4
Alien (1979)	nan

Fig. 9 ratings given by user id -4

Index	
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)	nan
Fargo (1996)	nan
Alien (1979)	nan
Indiana Jones and the Last Crusade (1989)	nan
Jurassic Park (1993)	nan
Stand by Me (1986)	nan
Sixth Sense, The (1999)	nan
Terminator, The (1984)	nan
Terminator 2: Judgment Day (1991)	nan

Fig. 10 Unrated movies by user_id 4

```
In [34]: cluster.mean().head(10) # this can be used for recommendation
Out[34]:
Star Wars: Episode IV - A New Hope (1977)                                    4.426829
Star Wars: Episode V - The Empire Strikes Back (1980)                        4.387500
Star Wars: Episode VI - Return of the Jedi (1983)                            4.081081
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)  4.416667
Silence of the Lambs, The (1991)                                             4.400000
Pulp Fiction (1994)                                                          4.264706
Fargo (1996)                                                                 4.411765
E.T. the Extra-Terrestrial (1982)                                            3.867647
Back to the Future (1985)                                                    3.897059
Alien (1979)                                                                 4.062500
dtype: float64
```

Fig. 11 Ten top movies recommended for user Id-4

To evaluate the accuracy of the model we have used three metrics SD, MAE, and RMSE. From fig. 7 we have rated value of the selected user and we get predicted values for these movies on fig. 10, then we calculate metrics values for the recommended movies to the user id 4. Table 1 below shows the given ratings and predicted rating for which the user wants to enjoy.

Table 1 given rating and predicted value for recommended movies

Rating	Predicted
5	4.426829
3	4.3875
5	4.081081
5	4.4
5	4.264706
4	3.867647
4	3.897059

After we calculated the metrics values, we have obtained standard deviation value (SD) 0.7568 and 0.6357MAE and 0.7584 RMSE where MAE processes the deviation between actual given ratings and ratings of the predicted movies.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}| p_i - r_i | \qquad \text{...(5)}$$

Root Mean Square Error (RMSE) is like MAE, yet puts more accentuation on bigger deviation that is given by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2} \qquad \text{... (6)}$$

Where p_i= is the given ratings

r_i = is the predicted ratings

n = is the amount of ratings

5. CONCLUSION:

In this paper hybrid of crow search and k-means algorithm is applied to the collaborative Movie lens dataset to obtain an enhanced movie recommendation system. We evaluated the accuracy of our approach regarding Silhouette score error, Elbow method for selecting right k number of clusters and SD, MAE, and RMSE for evaluating the accuracy of recommended movies to measure the performance of the proposed framework. The experiment outcomes on the Movie lens dataset explained marked that the approaches that we discussed provide high performance regarding accuracy and efficiency. Since we have used standard CSA where the fitness function which depends on values of AP and flight length initialized at the beginning; therefore for the future work use an improved CSA and other nature inspired algorithms in place of crow search algorithms.

REFERENCES:

1. Alireza Balavand, Ali Husseinzadeh Kashan, and Abbas Saghaei, “Automatic clustering based on Crow Search Algorithm-Kmeans (CSA-Kmeans) and Data Envelopment Analysis (DEA)”. International Journal of computational intelligence systems, Vol. 11, 2018.

2. Daniar Asanov, "Algorithms and Methods in Recommender Systems". Berlin Institute of Technology Berlin, Germany, 2016.
3. Farida Karimova (2016), "A survey of e-commerce recommender systems." vol.12.No. 34 ISSN:1857- 7881. DoI:10.19044/esj.2016.v12n34p75
4. Guo G, Qiu H, Tan Z, Liu Y, Ma J, Wang X (2017), " Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems".
5. Hamidreza Kooh, Kourosh Kiani, "User based collaborative filtering using fuzzy C-means". http://dx.doi.org/10.1016/j.measurement.2016.05.059.
6. K Lakshmi, N Karthikeyani Visalakshi And S Shanthi, "Data clustering using K-Means based on Crow Search Algorithm". https://doi.org/10.1007/s12046-018-0962
7. Maryam Jalloulia, Sonia Lajimi, Ikram Amous, "Designing Recommender System: Conceptual Framework and Practical Implementation". Peer-review under responsibility of KES International 10.1016/j.procs.2017.08.195
8. Ms. Dipa Dixit, Mr Jayat Gadge, "Automatic recommendation for online users using web usage mining". DOI : 10.5121/ijmit.2010.2303
9. Nitin Pradeep Kumar, Zhenzhen Fan, "Hybrid User-Item Based Collaborative Filtering". Procedia Computer Science 60 / 1453 – 1461, 2015
10. Phongsavanh Phorasim and Lasheng Yu, "Movies recommendation system using collaborative filtering and k-means" http://dx.doi.org/10.19101/ IJACR.2017.729004.
11. Prajyoti Lopes, Bidisha Roy, "Dynamic Recommendation System using web usage mining for E-commerce users". International conference on Advanced computing Technology and Applications (ICACTA-2015).
12. Rahul Katarya, Om Prakash Verma, "An effective collaborative movie recommender system with cuckoo search". Egyptian Informatics Journal 18 105-112, 2017.
13. Sajad Ahmadian, Mohsen Afsharchi And Majid Meghdad "A novel approach based on multi-view reliability measures to alleviate data sparsity in recommender systems". https://doi.org/10.1007/s11042-018-7079-x
14. Sambhav Yadav, Vikesh, Shreyam, Sushama Nagpal "An Improved Collaborative Filtering Based Recommender System using Bat Algorithm". Procedia Computer Science 132 / 1795–1803, 2018.
15. Santosh Kumar Uppada, "Centroid Based Clustering Algorithms- A Clarion Study". (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , (2014), 7309-7313, ISSN: 0975-9646.
16. Sonali B. Ghodake and Ratnamala S. Paswan, "Efficient recommender system using collaborative filtering techcnique and distributed framework". Vol.03, 2016 Issue:09 e-ISSN:2395-0056
17. Suni, prof. M. N. Doja , "Recommender System Based on Web Usage Mining for Personalized E-learning platforms". International Journal of modern Compuuter Science (IJMCS) ISSN: 2320-7868 (online) vol 5, Issue 3, 2017.
18. T.K.Das, "Intelligent Techniques in Decision Making: A Survey". Indian Journal of Science and Technology, Vol 9(12), 2016, DOI: 10.17485/ijst/2016/v9i12/86063.
19. Vibhor Kant, Tanisha Jhalani, Pragya Dwivedi, "Enhanced multi-criteria recommender system based on fuzzy Bayesian approach". DOI 10.1007/s11042-017-4924-2.
20. Vimala Vellaichamy, Vivekanandan Kalimuthu, "Hybrid collaborative Movie Recommendaer System Using Clustering and Bat optimization". International Journal of Intelligent Engineering and systems, vol.10,no.5. DOI:10.22266/ijies2017.1031.05
21. Wikipedia. https://towardsdatascience.com.
22. Aboul Ella Massanich, Rizk M.Rizk Allah, Mohammed Elhoseny, "A hybrid crow search algorithm based on rough searching scheme for solving engineering optimization problems." (2018), https://doi.org/10.1007/s12652-018-0924-y
23. Md. Akter Hossain, Mohammed Nazzin Uddin , "A Neural Engine for movie Recommendation System.". Department of Computer Science and Engineering, East Delta University, Chattogram, Bangladesh, 2018
24. Eugene Seo and Ho-Jin Choi . "Movie Recommendation with k-means clustering and self-organizing Map Methods." ICAART 2010-2nd International Conference on Agent and Artificial Intelligence. Department of Computer Science, Korea Advanced Institute of Science and Technology.