AN EFFICIENT CONTENT, COLLABORATIVE – BASED AND HYBRID APPROACH FOR MOVIE RECOMMENDATION ENGINE

Rajeev Kumar[*1] ; Guru Basava[2] ; Felicita Furtado[3]

[1,3] PG student, Department of Master of Computer Application, Jain Deemed-to-be university, Bangalore
[2] Assistant professor, Department of Master of Computer Application, Jain Deemed-to-be university, Bangalore

Abstract — The purpose of the project is to research about Content and Collaborative-based movie recommendation engines. Nowadays recommender systems are used in our day to day life. We try to understand the distinct types of reference engines/systems and compare their work on the movies datasets. We start to produce a versatile model to complete this study and start by developing and relating the different kinds of prototypes on a minor dataset of 100,000 evaluations. The growth of e-commerce has given rise to recommendation engines. Several recommendation engines exist within the market to recommend a wide variety of goods to users. These recommendations support various aspects such as users' interests, users' history, users' locations, and more. Away from all the above aspects; one thing is common which is individuality. Content and collaborative-based movie recommendation engines recommend users based on the user's viewpoint, whereas many things are there within the marketplace that are related to which a user is uninformed of. This stuff should also be suggested by the engine to clients; But due to the range of " individuality ", these machines do not suggest things that are out of the crate. The Hybrid System of Movie Recommendation Engine has crossed this variety of individuality. The Movie Recommendation Engine will suggest movies to clients according to their interest and be evaluated by other clients who are almost user-like. Additionally, for this, there are web services that are capable of acting as a tool adornment.

Keyword — Simple recommenders, Content-based recommenders, Collaborative filtering engines, hybrid recommenders, dataset and prototypes.

1. INTRODUCTION

A movie recommendation engine / system can be an information sorting system that works to estimate ratings or preferences [1] and will give the user an item and set up a simple or similar language "recommendation engine / system to attract the user something. Suggests important supported ".

Recommendatory [2] systems can also enhance the experience for:

- News websites
- computer games
- Knowledge base
- Social media platform
- Stock trading support system

A content and collaborative-based recommendation engine / system can also be a method of information sorting system that works to predict user preferences and provide suggestions that support them. The content on some platforms extends from movies, music, books and videos to friends. And to produce stories on social platform and on e-commerce websites, for persons on professional and dating websites, returned to see search results [3, 4, 5, 6, 7]. Two critical approaches are mainly used for recommendation engines. First, content-based filtering, where we attempt to profile client interests utilizing gathered information and recommend items that support that profile, and second, collaborative filtering [8] continues where we try and identify together and use information about identities to create recommendations systems. Every user has a different mindset to decide their likes and dislikes.

Additionally, even a customer's taste can look at different aspects, such as mood, seasons, or different activities performed by the user. As an example, the type of music you want to focus on during exercise is severely different from that in which he listens to music while making dinner. They have to find new areas to see more about the customer, while still determining the majority of what is already known about the customer.

Evaluation

We use the root mean squared error metric to calculate the predictions made by our system.

This is calculated as the root mean squared error equation: --

$$\sqrt{\frac{\sum_{(i,j)\in T} r_{ij} - r_{ij}^{*}}{N}}$$

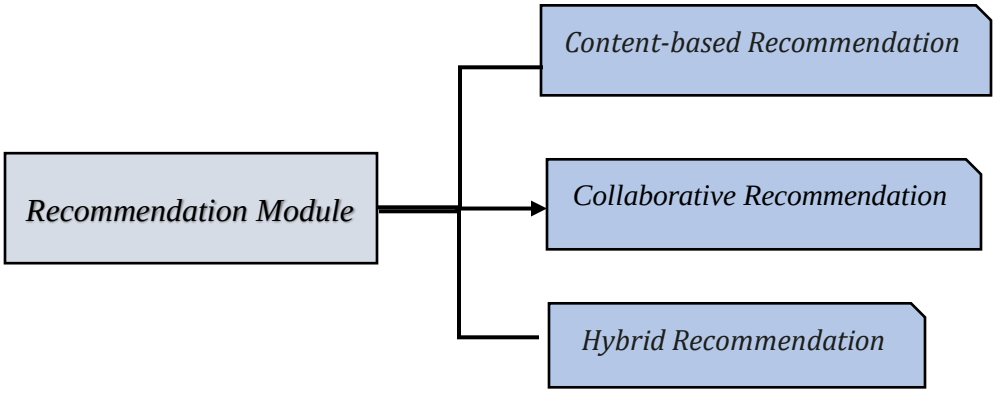
Where, N = no. of ratings in the test division,

p_{ij} = predicted rating for user i and movie j and

r_{ij} = actual rating.

2. EXISTING RECOMMENDATION MODELS

The recommendation module is a different kind of data filtering. Preferences and ratings are used to estimate or an item will be given to a user. The recommendation module uses content-based, collaborative and hybrid filtering technique.



2.1 Introduction to Simple recommenders

The simple recommendation system process provides generalized recommendations to each user, supported movie popularity and / or genre [9]. The basic approach behind this technique is that more popular and critically acclaimed movies will be better likely to be liked by the general audience. For example, IMDB Top 250 is an example of this technique.

2.1.1. Introduction to Content Based Filtering

The content-based recommendation [10, 11, 12] system process checks for user ads and changes and creates a user-based profile. Suggest similar items supported selected items. This engine uses item metadata for films, such as genre, director, description, actor, etc., to make these recommendations. The final idea behind these recommendations is that if a person sorts out a particular object, he is going to like the object that is related to it. The content-based filtering [10] technique has two approaches, such as the Top-N approach and the rating scale approach.

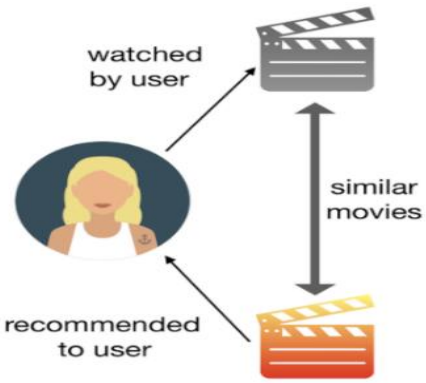


figure.1 Content-based filters

Content-based (cognitive filtering) recommendation systems recommend using items that support comparisons between the content and user profiles of those things. Otherwise we will get a notification for recommendation for supported movie layouts of other movies.

Advantage of content-based filtering: -

- This has the capability of mentioning unrated objects
- It is easy to explain the work of recommender system by catalog the content features of an item.
- Content-based recommender systems require only the rating of the concerned client, no one else's.
- Operator of the system.

Disadvantage of content-based filtering: -

- A new user who has not rated any item will not work for them, as enough ratings are required content-based recommender assesses the client inclinations and gives exact proposals.
- No proposals of unexpected objects.
- Restricted Content Study - If the system fails to distinguish the items the recommender does not work.
- A client likes items that he/she does not like.

2.1.2. Introduction to Collaborative Filtering

Collaborative filtering [8] methods make recommendations and demonstrations for user-supported ratings and data from multiple users' preferences. These systems attempt to rating or preference and

obtain that the client will provide an object based on the previous rating and preferences of different clients. These methods do not require metadata for objects like content-based equivalent.

Collaborative filtering [13] methods approached two filtering techniques, such as model-based or memory-based. These techniques define a model based on user-item interactions where the user or item representation has to learn from the interaction matrix. Memory-based filtering techniques do not interact with user-items [14] and depend on match among clients or objects in terms of observed interactions.

Model based filtering technique has several approaches, such as clustering technique, association technique, Bayesian network [15], neural network and so on. Memory based filtering technique has two approaches, such as client based and object based [16].

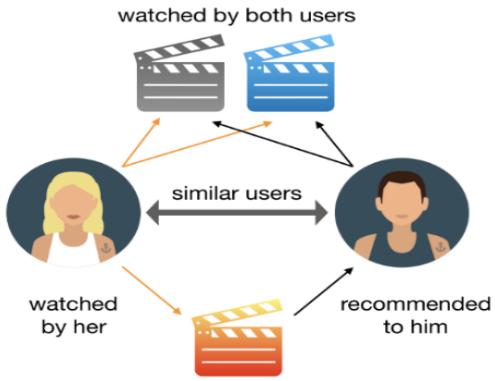


figure.2 Collaborative-based filters

The collaborative uses recommended filtering in association with user or movie IDs. suppose an example, there are two users 'Sunil' and 'Veer', Sunil likes movies O Teri, flash, Batman, Welcome and Veer likes movies flash, Batman, Welcome, Super Singh. Such as common movies flash, Batman and Welcome. User prefers; therefore, O Teri movie is being recommended for Veer and Super Singh movie is recommended for Sunil.

We see collaborative filtering technology in action on various kinds of online video platforms like Amazon Prime Video, Netflix, Zee5, YouTube and Facebook. We have suggested those supported ratings and buy the data that these platforms collect from their user and item base. We explore two procedures for collaborative filtering, the k-nearest neighbor algorithm and also the latent factor procedure.

Advantage of collaborative filtering-based* [17] *filtering: -

- It relies upon the connection between clients which infers that it is content-autonomous.
- Collaborative Filtering recommender systems can propose unexpected items by detecting like-minded person's behavior.
- They can make genuine quality evaluation of objects by considering other person's experience.

Disadvantage of content-based filtering: -

- Early rater problem: Collaborative filtering systems cannot suggest for fresh items as there are no client ratings on which to base a forecast.
- Gray sheep: In order to Collaborative filtering-based system to work, group with related features are required. It will be very tough to suggest clients who do not steadily agree or disagree to these groups, if such groups exist.
- Sparsity problem: Mostly, the number of items exceeds the number of clients by a huge margin which makes it tough to find items that are rated by ample individuals.

2.1.2.1. Nearest Neighbors Collaborative Filtering

The concept of nearest neighbors [18] collaborative filtering that users with rating behavior similar to this point share similar tastes and possibly exhibit similar rating behavior further. The algorithm first calculates relationship between clients using a row vector within the rating matrix, treating the client as a demonstration for that client. Parity is calculated by either cosmic similarity or Pearson correlation. So, to guess the rating for a selected client for a specified film, we search for users who are similar to the highest for the current specific client, then take the same average weighting of users' ratings as k, which is the value of equality.

2.1.2.2. Latent Factor Methods

The latent factor method [19, 20] appears to decompose the rating matrix M into two long and thin matrices U and V, in which the dimension of matrix P is num_users × k and V, whose size is num_items × k, where k is the number of implicit latent factors. The decomposition of M is specified in U and V

We get,

M = u × v ^ t

Any rating r_{ij} within the rating matrix can be calculated by taking the scalar product of row U_i of matrix U and V_j of matrix V. Matrix U and V are initialized randomly or by performing SVD on a rating matrix. Then, the algorithm solves the case of minimizing the error between the special rating values r_{ij} and simultaneously taking the scalar product of the values U_i and V_j. These algorithm displays the probability gradients to exclude matrices U and V with minimal error or ranging from early matrices.

2.1.3. Introduction to Hybrid System

The hybrid recommendation [21, 22] system is a blend of content-based or collaborative filtering methods. Content-based or collaborative filtering methods are executed individually and the outputs

of both methods are recommended to jointly suggest better. The spread of distance between feature vectors can also be used to calculate the similarity of two objects: -

- Content-based similarity
- Weight from collaborative network

Hybrid methods used a combination of two approaches, such as a content-based and collaborative filtering approach.

Table No.1: The steps of hybrid algorithm shown below: -

Sr. No.	Algorithm of hybrid recommendation module
Step No. 1:	Use the content-based predictor and to analyse the pseudo user-rating vector 'x' for every user 'z' in the database. Whereas, $x_{z,k} = r_{z.k}$: is user 'z' rated item 'k' $x_{z,k} = r_{z.k}$: Otherwise
Step No. 2:	The weight of all users in relation to equality with the active user. Likeness between users is measured as the Pearson correlation among their rating vectors.
Step No. 3:	Select the users 'm' that have the maximum likeness with the active user. These users create neighbours.
Step No. 4:	Predict from a weighted combo of the selected neighbours ratings. In step 2, the similarity between two clients is calculated using the Pearson correlation coefficient, defined below: $$P_{a,x} = \frac{\sum_{k=1}^{m}(r_{x,k} - \overline{r_a}) \times (r_{x,k} - \overline{r_x})}{\sqrt{\sum_{k=1}^{m}(r_{a,k} - \overline{r_a})^2 \times \sum_{k=1}^{m}(r_{x,k} - \overline{r_x})^2}}$$ Where, $r_{a,k}$: given rating to item k by active user a; $\overline{r_a}$: mean rating given by active user a; m : the total number of items. In step 4, predictions are computed as the weighted averages of deviations from the neighbours mean: $$P_{a,k} = \overline{r_a} + \frac{\sum_{x=1}^{n}(r_{x,i} - \overline{r_x}) \times P_{a,x}}{\sum_{x=1}^{n} P_{a,x}}$$ Where, $P_{a,k}$: the prediction for the active user a for item k; $P_{a,x}$: the similarity between users a and x; n : the number of users in the neighbours.

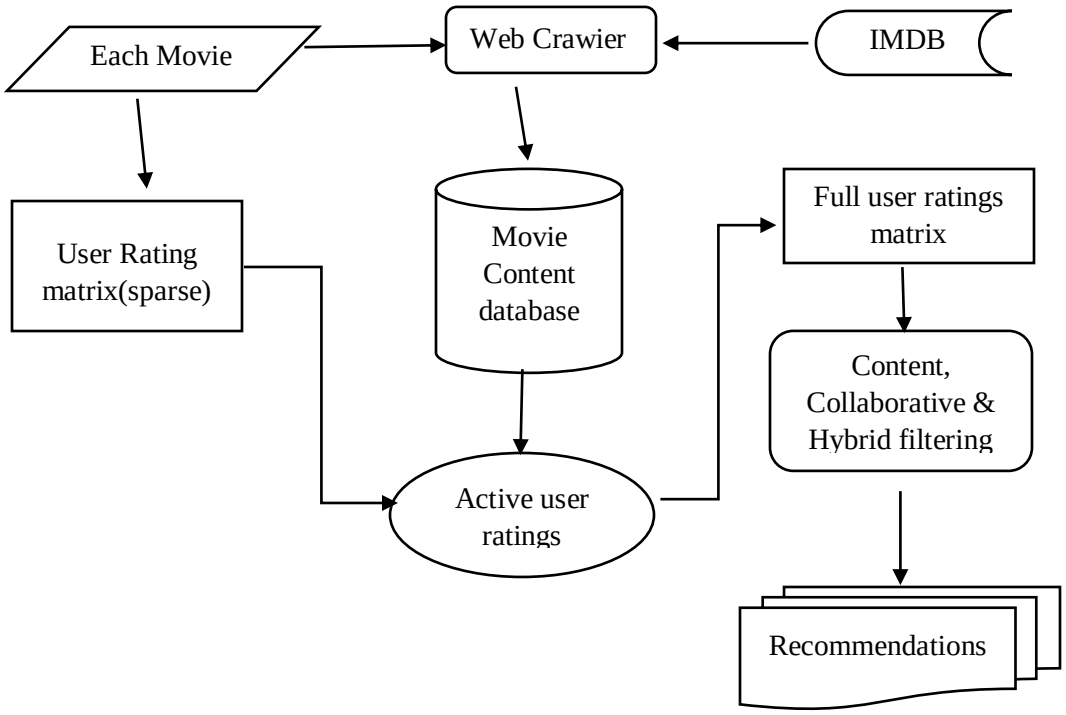


figure.3 Hybrid method

3. LITERATURE REVIEW

In the context of a review of the literature, a recommendation system using a content-based collaborative and hybrid approach [1] by a previous researcher is a different approach to the development of recommendation-based engines. In 2007 a web-based and knowledge-based intelligence movie recommendation system has been offered using the hybrid filtering method [23]. In 2017, a movie recommendation system supported style and rating coefficient of correlation purpose [24] by the authors. In 2013 a Bayesian network and trust model-based movie recommendation engine [25] have been recommended to predict ratings for users and items, primarily from datasets to recommend users their choice and vice versa. In 2018, the authors built a recommendation engine by analyzing the ratings dataset collected from Kaggle to recommend movies for a user selected from Python. In 2018 movie recommendation engines provide a process to help users categorize users with similar k-mean [1] cuckoo values and reinforcement learning based recommender systems, which are using bicycling techniques [26, 27]. Initial research mainly concentrated on the content of the recommendation system that examined the features of the object to complete the recommendation task. Experiments verified that their approaches were more elastic and precise. Bayesian networks are employed for model-based preferences based on their context. In 2007 Salakhuddinov and Minh proposed a collaborative filtering method [28], the probabilistic matrix factor, which can handle large-scale datasets. The collaborative filtering algorithm was distributed into portions for deeper study in the movie recommendation by Hurlkartal. When clients adopt new behavior, it is difficult for collaborative filtering to react instantly. Therefore, both

researchers and practitioners have a desire to align collaborative filtering method and content-based methodology to solve the issue [24, 29]. Ternary implemented Unplugged Learning of Machine Learning to examine the polarity of machine reflectivity [13, 30].

3. SYSTEM ARCHITECTURE

This section covers the architecture, process flow, DFD diagram and ER diagram of the system architecture.

Architectural Diagram: -

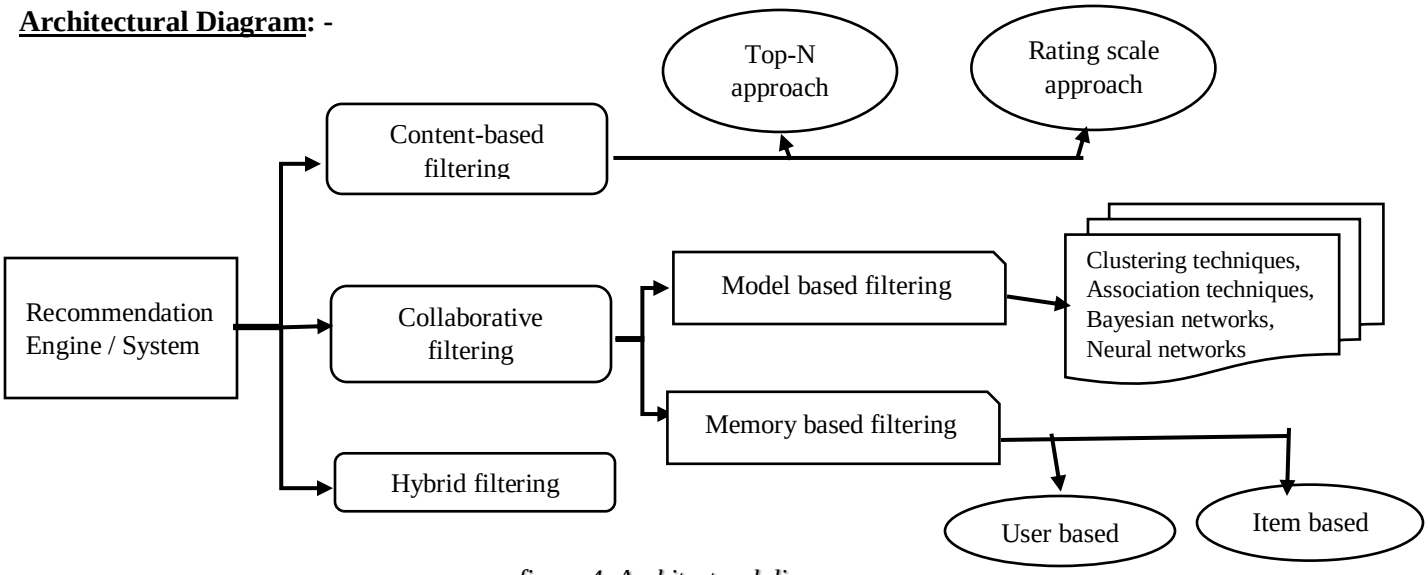


figure.4: Architectural diagram

Process Flow Diagram: -

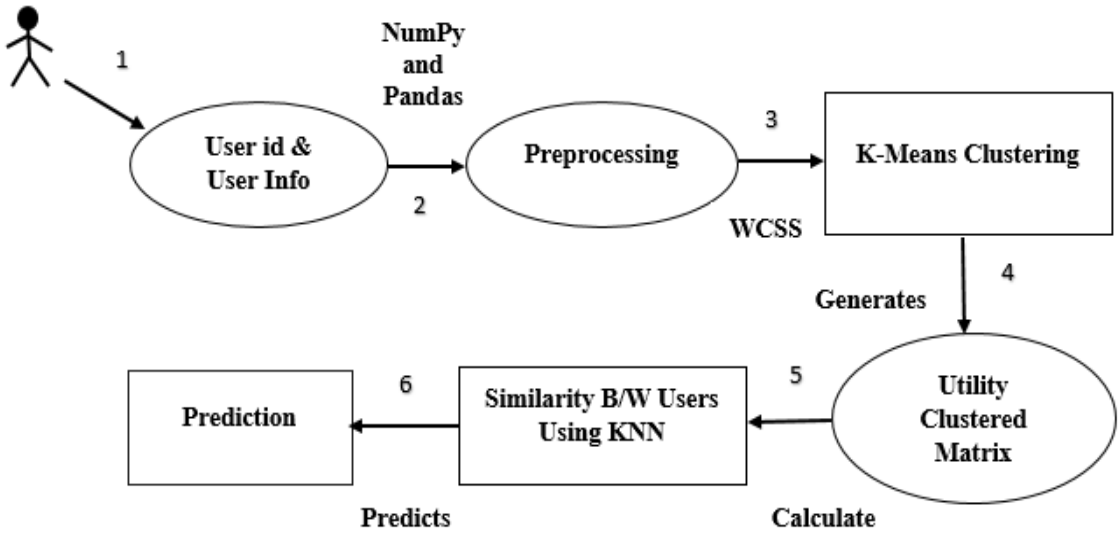


figure.5 Process flow diagram [1]

Table No.2: Description of the process flow diagram shown below: -

Sr. No.	Proceedings of methods
Step 1:	The respondent provides with the user Id and facts such as gender, age, pin code.
Step 2:	Using the NumPy and pandas' libraries; the raw data is pre-processed into separate data frames.
Step 3:	Used to find the correct no clusters within the clustered sum of the squared method to apply K-mean clustering to the movie.
Step 4	After applying the K-means a utility clustering matrix is created which defines the average rating given by the user to each cluster.
Step 5:	Using a utility clustered matrix and Pearson correlation similarity between users is calculated.
Step 6:	Finally, the KNN input uses the utility cluster matrix and parity to predict movies for the user.

DFD Diagram: -

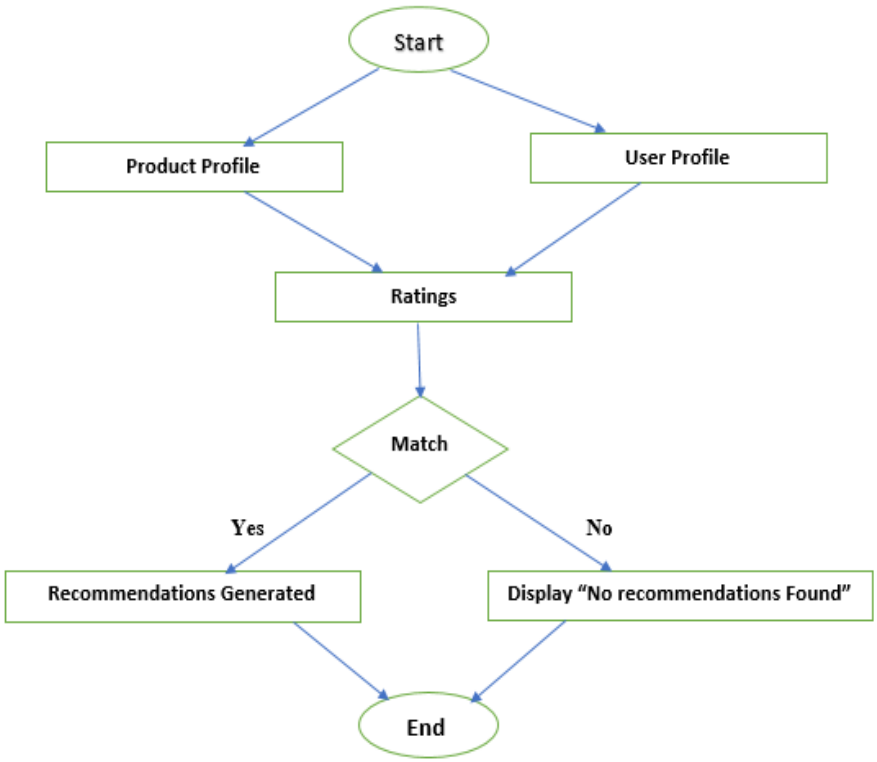


figure.6 DFD diagram

ER-Diagram: -

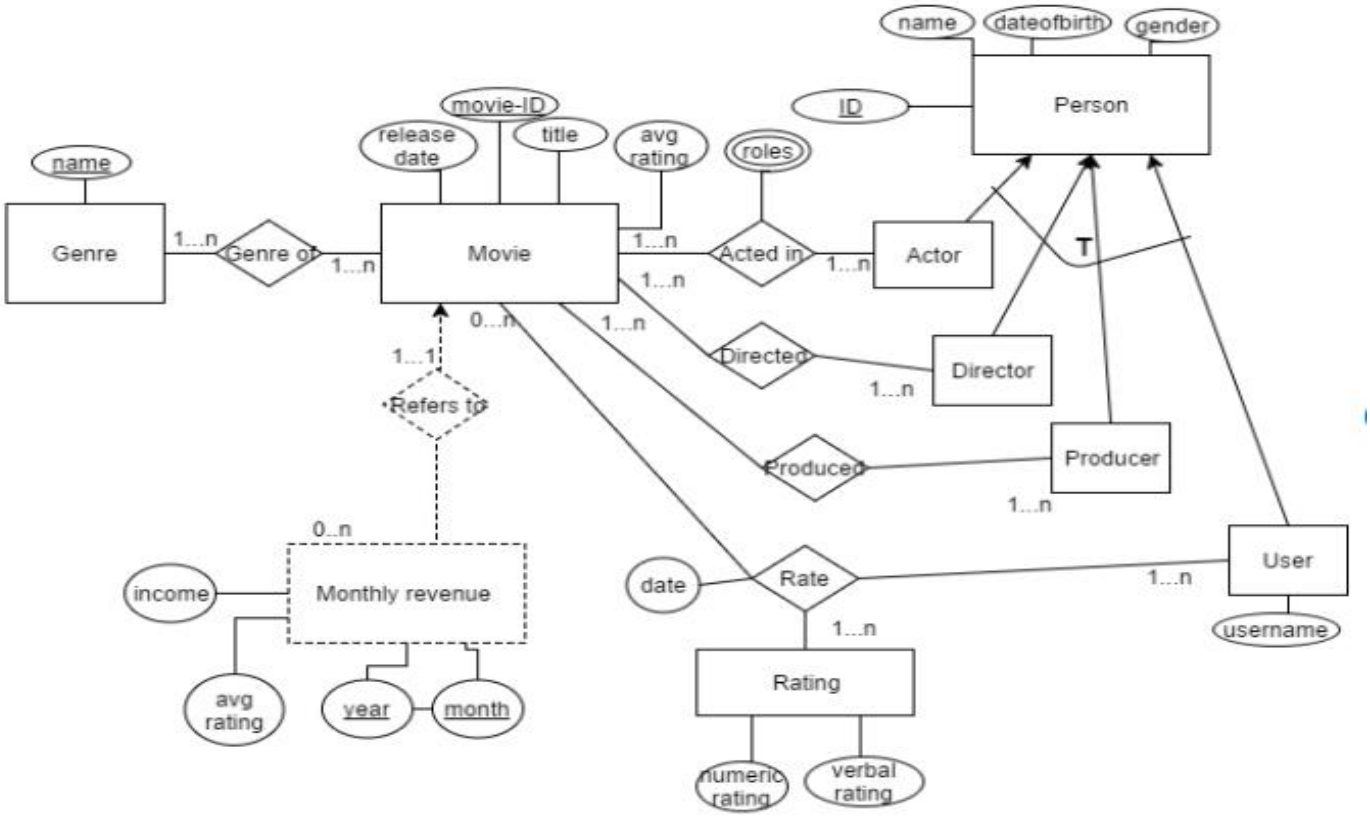


figure.7 ER diagram

4. PROPOSED SYSTEM

Basically, the purpose of a recommendation system is to search for material that will be interesting to a person. In addition, it includes many factors to create a personalized list of useful and interesting content useful to each user. Recommendation systems are both machine learning and artificial intelligence-based algorithms that skim through all possible options and create a personalized list of items that are interesting and relevant to an individual. These results are based on their profile, browsing history, watching others with similar traits / demographics and how you can watch those movies. This is entire through predictive modeling and heuristics with available data.

4.1. Proposed Algorithm

The following steps are included for the proposed algorithm is shown in Table 3 below:

Table 3: The proposed algorithm

Steps	Methodology
Step 1:	Import python libraries: NumPy, Pandas, Matplotlib, sklearn.
Step 2:	Read all information CSV as data frame within user and item variable ratings.
Step 3:	Split as information sets and variable ratings into training sets and test sets and set as information frames in rating tests.
Step 4:	Create a utility matrix or name utility that tells which user rates belong to which movie.
Step 5:	Using the WCSS method and selecting the appropriate number of clusters, then the K-means clustering technique is often applied to categorize flicks to keep them within the number of clusters.
Step 6:	Characterize the utility cluster matrix after implementing the K-means clustering algorithm.
Step 7:	Apply the Pearson correlation metric to the utility cluster matrix to calculate the similarity matrix between users.
Step 8:	To specify a value stored within the utility matrix.
Step 9:	The guess() function takes the two parameters as input userID and TopN users employed by KNN that estimate the movie ratings for TopN similar users.
Step 10:	The ratingtest data frame is used for rating comparisons when using the guess() function to estimate users' ratings.
Step 11:	RMSE (Roots stands for Mean Squared Error) is applied to calculate to rate the accuracy of the model.
Step 12:	Finally, we find the results of the recommendation model.

We use some sort of recommendation algorithm for our research paper which is simple recommender, Content-based filtering, Collaborative filtering & Hybrid System.

- Metric and score will have to be chosen to extract the rate of movies.
- We are trying to find out the score and value of every movie.
- The highest results are needed to extract scores and output.

4.2. Problem Statement

In the problem statement, providing content related to the collection of relevant and irrelevant items for users of online service providers. The purpose of the movie recommendation system is to

recommend movies to users based on user / item base movie ratings. Given a set of users with their previous rating for a set of movies, can we infer the rating they would assign to a movie not already rated?

For example. “Which movie will you like”, given that you have seen Ironman, Ironman II, Ironman III; the last stand and users who saw these movies also liked “Ironman origins: Tony Stack”?

Table 4: Prediction algorithm should try to estimate the rating on MsDoni for Felicita Furdato.

User / movie name ➡ ⬇	Good News	Ms Dhoni	Dhoom3	Ek tha Tiger
Rajeev kumar	5	5	2	4
Guru Basava	3	4	3	3
Rajesh Budihul	4	4	3	3
Ranjan kumar	4	5	3	2
Felicita furdato	5	?	3	4

.

5. IMPLEMENTATION

In this segment, we have explained the way to be implemented in our movie recommendation engine / system using python programming language using K-Means clustering [31] library and K-Nearest Neighbor. The implementation of the system consists of many sub-sections which are standard processes to be followed while solving any machine learning [32, 33, 34] problem.

These are as follows:

1) Dataset

2) Data Cleaning

3) Model Analysis

4) Model Building

A. Dataset

The movie dataset is hired in our research paper and collected from the Kaggle database. The Kaggle [35, 36] database provides datasets in the form of several varieties of movie content. The user rating data consists of 211231 records and has a user ID, movie ID, rating, and timestamp. The Characterization of the movie's content information includes over 54058 records and includes movie ID, title, genre, director, actor, and more.

B. Data Cleaning

Description of movies content data consists approximately 40 columns. Most of those columns were not necessary for our research paper and were therefore removed. The dataset contains plenty of empty values and duplicate values that need to be resolved. Additionally, there have been some entries for movies within the user rating files from movies, which did not correspond to any movies within description of movie content data. These entries were erased for easy processing.

C. Data Analysis

We evaluated datasets to gain insight into movie datasets from Kaggle database that can help to develop and perform our engine / system using Matplotlib libraries in python programming. We acknowledged patterns for most rated movies, most rated genres, the number of movies in each genre, and the number of rated movies in each rating category as shown below.

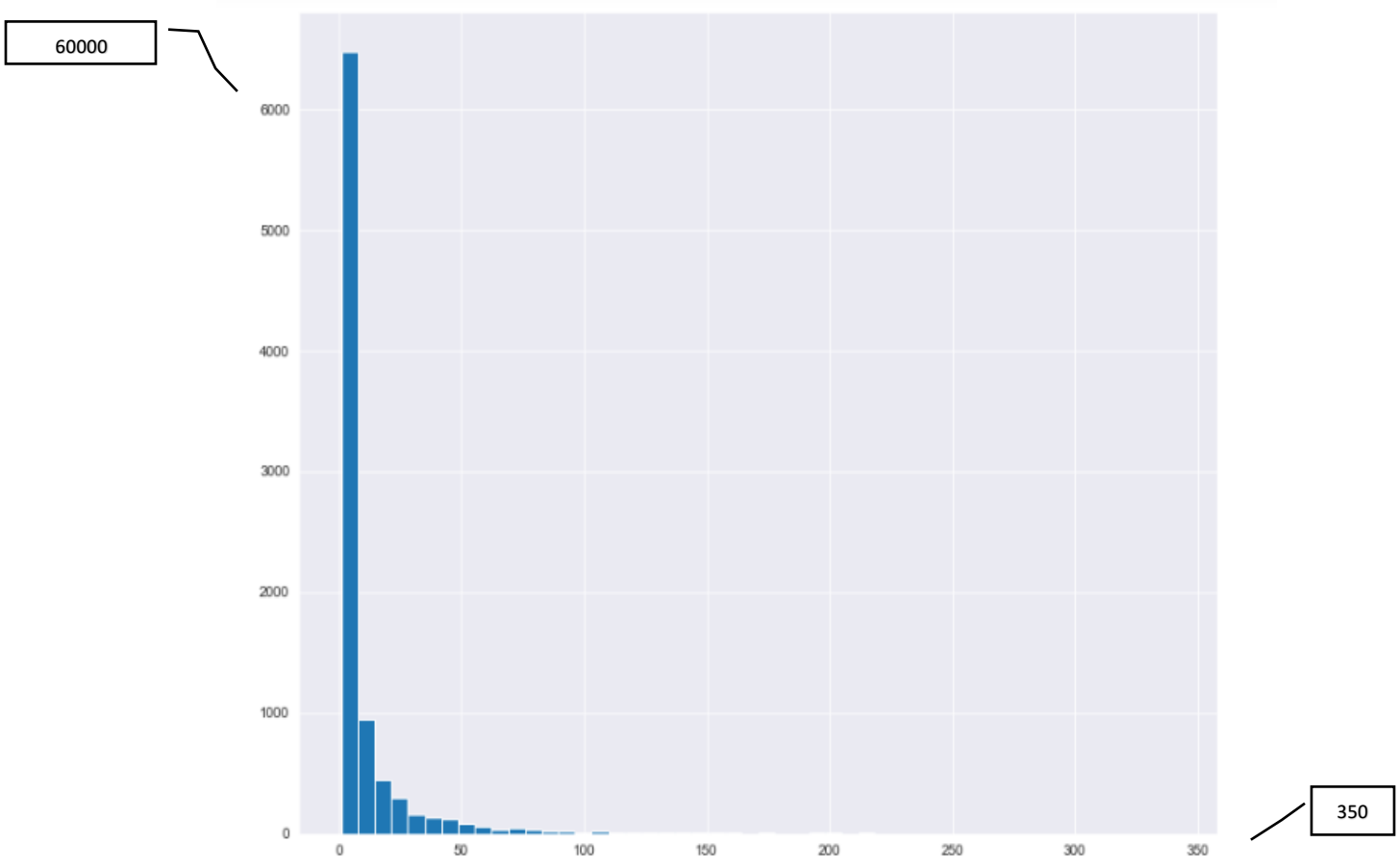


Figure.8

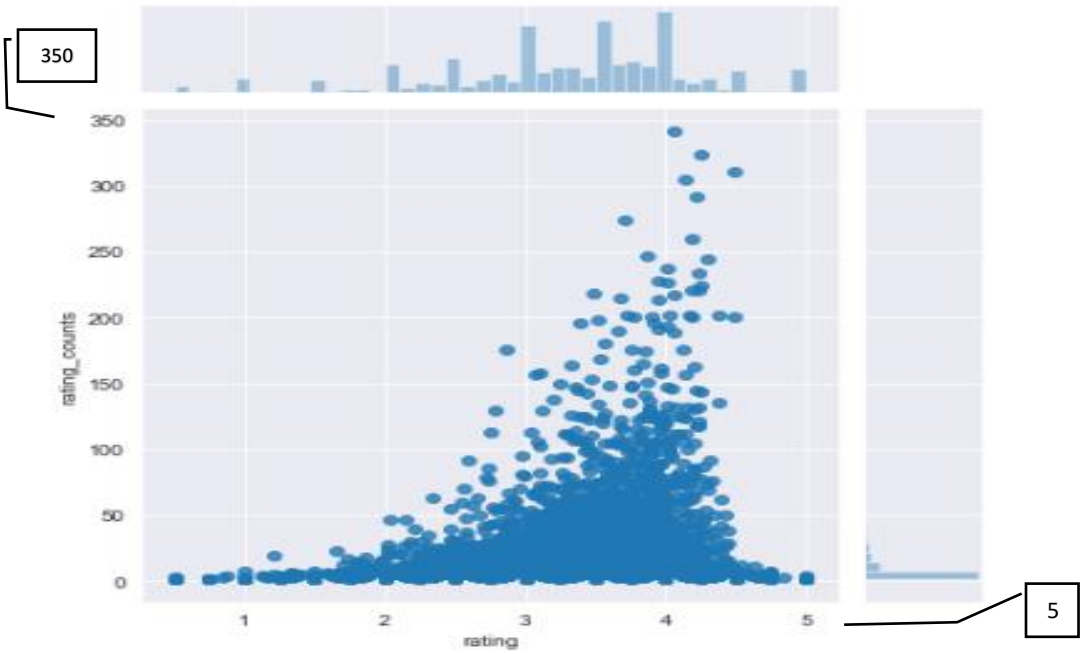


Figure.9 graphically represent of rating vs rating_column

We try to import the Seaborne library and use the combined plots (x = 'rating', y = 'Rating_counts', data = Rating_mean_count, alpha = 0.8). This can be important in the order that we are able to see the link between a movie's specific rating and therefore how much the movie got. It is very possible that a 5-star movie is rated by only 1 person. It is therefore statistically inaccurate that a 5-star movie is included to classify the move. Therefore, we should set a limit for the minimum number of ratings while constructing the recommended system.

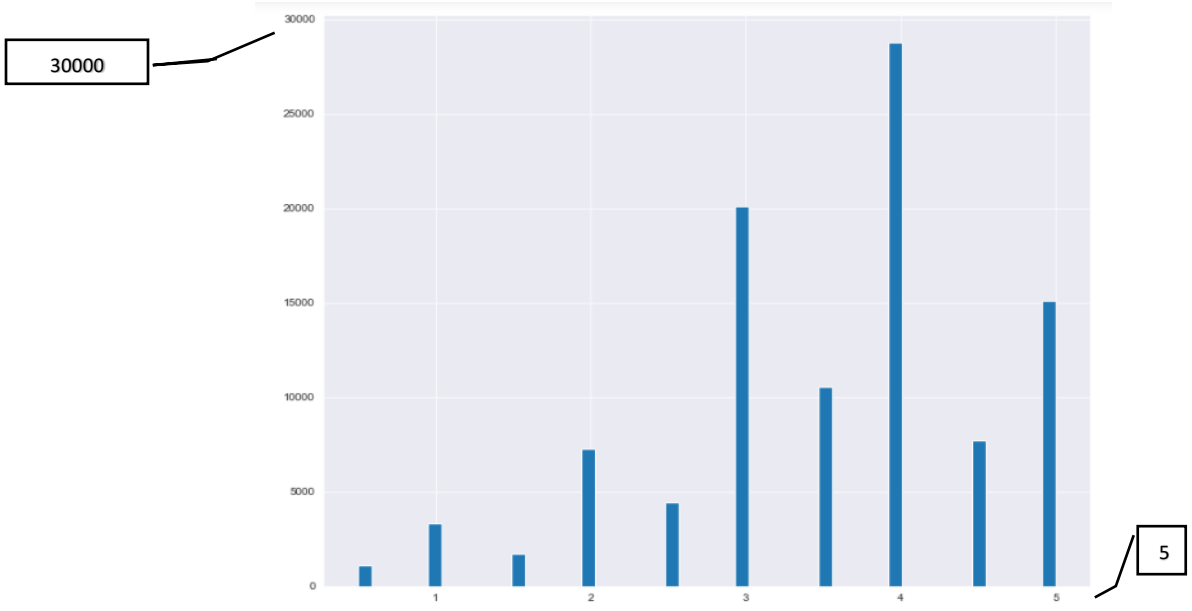


Figure.10 the histogram representation of the rating count vs rating mean

From the histogram above, it is clear that most of the movies have low ratings. Most rated movies are the ones that are most famous.

	userId	movieId	rating	timestamp	title	genres
0	1	31	2.5	1260759144	Dangerous Minds (1995)	Drama
1	7	31	3.0	851868750	Dangerous Minds (1995)	Drama
2	31	31	4.0	1273541953	Dangerous Minds (1995)	Drama
3	32	31	4.0	834828440	Dangerous Minds (1995)	Drama
4	36	31	3.0	847057202	Dangerous Minds (1995)	Drama
5	39	31	3.0	832525157	Dangerous Minds (1995)	Drama
6	73	31	3.5	1255591860	Dangerous Minds (1995)	Drama
7	88	31	3.0	1239755559	Dangerous Minds (1995)	Drama
8	96	31	2.5	1223256331	Dangerous Minds (1995)	Drama
9	110	31	4.0	840100695	Dangerous Minds (1995)	Drama

Figure.11(a) Using head command and showing old movies talk over with the genre

	movieId	title	genres
9115	161830	Body (2015)	Drama|Horror|Thriller
9116	161918	Sharknado 4: The 4th Awakens (2016)	Action|Adventure|Horror|Sci-Fi
9117	161944	The Last Brickmaker in America (2001)	Drama
9118	162376	Stranger Things	Drama
9119	162542	Rustom (2016)	Romance|Thriller
9120	162672	Mohenjo Daro (2016)	Adventure|Drama|Romance
9121	163056	Shin Godzilla (2016)	Action|Adventure|Fantasy|Sci-Fi
9122	163949	The Beatles: Eight Days a Week - The Touring Y...	Documentary
9123	164977	The Gay Desperado (1936)	Comedy
9124	164979	Women of '69, Unboxed	Documentary

figure.11(b) using tail command and showing new movies talk over with the genre

Therefore, we must set a threshold for a minimum number of ratings while constructing a system that recommends. So, to create this new column we use the utility of pandas' groupby. We groupby the title columns, so use the calculation method to calculate the number of ratings each movie received. Later, we look at movie_data using the head() function and calculate the average rating on the Sort_values of all movies.

```
title
Burn Up! (1991)                                  5.0
Absolute Giganten (1999)                         5.0
Gentlemen of Fortune (Dzhentlmeny udachi) (1972) 5.0
Erik the Viking (1989)                           5.0
Reality (2014)                                   5.0
Name: rating, dtype: float64
```

figure.12 groupby number of rating

Figure 12 shows the title of the movie with the highest rank as a descending order and those movies have been recommended by users

title	rating	rating_counts
"Great Performances" Cats (1998)	1.750000	2
$9.99 (2008)	3.833333	3
'Hellboy': The Seeds of Creation (2004)	2.000000	1
'Neath the Arizona Skies (1934)	0.500000	1
'Round Midnight (1986)	2.250000	2
'Salem's Lot (2004)	3.500000	1
'Til There Was You (1997)	2.625000	4
'burbs, The (1989)	3.052632	19
'night Mother (1986)	5.000000	3
(500) Days of Summer (2009)	3.755556	45

.

figure.13 find out number of rating with apply threshold

We will create the value of movie_data 'rating' using movie_title. We will create a data_frame and then store the movie_data, calculate rating_count-in 'title or rating rating' using the rating_mean_count.head() method and get the result. The value of rating_count in figure 13 compares with the title and rating.

D. Model Building

We used the Python library to create a recommendation system [4]. For user-based filtering and item-based filtering, we used the user similarity class for Pearson correlation similarity, which uses Pearson correlation to work out similarity between users' ratings; Hence priority [14].

Table 5: An excerpt of the survey is given below: -

Name	Cast/Actor	Genre	Rating	Director	Year
Rahul Kumar	2	5	1	4	3
Pritam Roy	5	3	1	2	4
Alok Singh	1	3	2	5	4
Farah Khan	3	2	1	4	5
Salam Khan	2	3	1	4	5
Diljit Singh	1	4	2	5	3
Gippy Agarwal	3	2	1	5	4

In our survey, I have found that just about every user has given movie-related ratings the maximum priority, so I have given our rating feature the highest priority. Also, I have come to know from our

poll that the movie has more weight in its ratings if that movie has got a higher number of votes. Therefore, I have got ratings and votes divided into three categories i.e. minimum, medium and maximum votes.

6. CONCLUSION AND FUTURE WORK

In this research paper, we have implemented a movie recommendation engine / system using simple recommendations, content-based filtering, collaborative filtering, and hybrid systems. In addition, a movie recommendation engine has been developed using different method prediction methods. This model is implemented in the python programming language. We have observed that the RMSE value of the proposed technique is healthier than the current technology after implementing the system with the help of python programming language.

In future, we can try and test the system using more data and improve the accuracy of the system. In addition, we can try users better to increase the accuracy of the recommendation system.

ACKNOWLEDGEMENT

I would like to express my gratitude and obligation to Professor Guru Basava, Professor Rajesh Budihul and Dr. Ranjan Kumar for his effective conduct and constant motivations during his analysis work. Their timely direction, full cooperation and minute observation have made my work valuable. I would also like to thank my mentor Professor Guru Basava, who wanted to provide me all the facilities that were required. Finally, I would like to thank my partner and friends for their support and encouragement throughout my studies.

REFERENCES

[1] R. Ahuja, A. Solanki and A. Nayyar, "Movie recommender system using K-means clustering and K-nearest neighbor," in *9th International Conference on Cloud Computing, Data Science & Engineering*, 2019.

[2] N. Severt, "An Introduction to Recommender Systems," Iterators, 6 Nov 2018. [Online]. Available: https://www.iteratorshq.com/blog/an-introduction-recommender-systems.

[3] Chavarriaga, Oscar, Florian-Gaviria, Beatriz, Solarte and Oswaldo, "A recommender yystem for students based on social knowledge and assessment data of competences," in *School of Systems and Computing Engineering (EISC),*, Colombia , 2014.

[4] Katarya, R. Verma and O. Prakash, "An effective collaborative movie recommender system with cuckoo search," *Egyptian Informatics Journal*, vol. 18, no. 2, pp. 105--112, 2017.

[5] Drachsler, H. Bogers, R. V. T Vuorikari, K. Duval, N. B. G. L. S. S. H. F. E Manouselis and Martin, "Issues and considerations regarding sharable data sets for recommender systems in technology enhanced learning," *Procedia Computer Science*, vol. 1, no. 2, pp. 2849--2858, 2010.

[6] T. Lorente, P. Carlos and S. Eduardo, "A quality based recommender system to disseminate information in a university digital library," *Information Sciences*, vol. 261, pp. 52--69, 2014.

[7] T. Tran, T. N, Atas, M. Felfernig and A. e. al, "An overview of recommender systems," in *Journal of Intelligent Information Systems*, 2018.

[8] Wu, C.-S. M. a. Garg, D. a. Bhandary and Unnathi, "Movie recommendation system using collaborative filtering," in *9th International Conference on Software Engineering and Service Science*, Beijing, China, 2018.

[9] C. Basu, H. Hirsh and W. Cohen, "Recommendation as classification : Using social and content-based information in recommendation," in *Proceedings of the 15th National Conference on Artificial Intelligence*, New Jersey, 1998.

[10] H. W. Chen, Y. L. Wu, M. K. Hor and C. Y. Tang, "Fully content-based movie recommender system with feature extraction using neural network," in *International Conference on Machine Learning and Cybernetics*, Ningbo, China, 2017.

[11] B. R. Cami, H. Hassanpour and H. Mashayekhi, "A content-based movie recommender system based on temporal user preferences," in *Iranian Conference on Intelligent Systems and Signal Processing* , Shahrood, Iran, 2017.

[12] C. Davidsson and S. Moritz, "Utilizing implicit feedback and context to recommend mobile applications from first use," in *Proceedings of the Workshop on Context-Awareness in Retrieval*

and Recommendation, California, Palo Alto, USA, 2011.

[13] W. Kebin and T. Ying, "A new collaborative filtering recommendation approach based on naive Bayesian method," *International Conference in Swarm Intelligence*, vol. 6729, pp. 218--227, 2011.

[14] Z.-D. Zhao and M.-s. Shang, "User-based collaborative filtering recommendation algorithms on hadoop," *Proceedings of Third International Workshop on Knowledge Discovery and Data Mining*, pp. 478--481, 2010.

[15] Munoz-Organero, Mario, A. R.-G. Gustavo, J. M.-M. Pedro and C. D. Kloos, "A collaborative recommender system based on space-time similarities," *IEEE Pervasive Computing*, vol. 9, no. 3, pp. 81--87, 2010.

[16] A. Jain and S. K. Vishwakarma, "Collaborative Filtering for Movie Recommendation using RapidMiner," *International Journal of Computer*, vol. 169, no. 6, pp. 0975 - 8887, 2017.

[17] R. Hande, A. Gutti, K. Shah, J. Gandhi and V. Kamtikar, "MOVIEMENDER-A movie recommender system," *Internal Journal of Engineering Sciences & Research Technology*, vol. 5, no. 11, p. 686, 2016.

[18] P. Symeonidis, A. Nanopoulos, Papadopoulos and A. e. al, "Nearest-biclusters collaborative filtering based on constant and coherent values," vol. 11, p. 51–75, 2007.

[19] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris France, 2009.

[20] H. Langseth and T. D. Nielsen, "A latent model for collaborative filtering," *International Journal of Approximate Reasoning*, vol. 53, no. 4, pp. 447 -- 466, 2013.

[21] N. T. N. R. R. S. C. Jain, "Hybrid filtering methods applied in web-based movie recommendation system," *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 4692, pp. 206--213, 2007.

[22] S. Grant and G. I. M. ., "A hybrid approach to making recommendations and its application to the movie domain," in *Conference of the Canadian Society for Computational Studies of Intelligence*, 2001.

[23] N. T. N. R. R. S. C. Jain, "Hybrid filtering methods applied in web-based movie recommendation system," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2007.

[24] Y. Zhang, Z. Tu and Q. Wang, "TempoRec: Temporal-topic based recommender for social network services," *Mobile Networks and Applications*, vol. 22, no. 6, pp. 1182--1191, 2017.

[25] W. Du and J. Chen, "The Bayesian network and trust model based movie recommendation system," in *Intelligence Computation and Evolutionary Computation*, Springer, Berlin, Heidelberg, 2013, pp. 797--803.

[26] H. H. U. H. C. K. J.-W. H. S. Y. Sungwoon Choi, "Reinforcement learning based recommender system using biclustering technique," in *International Conference on Information and Communications Technology*, Los Angeles, USA, 2018.

[27] M. L. L. Sebastian Thrun, "Reinforcement Learning: An Introduction," vol. 21, no. 1, p. 103, 2000.

[28] A. M. Ruslan Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, Canada, 2008, pp. 1257--1264.

[29] P. C. George Lekakos, "A hybrid approach for movie recommendation," *Multimedia tools and applications,* vol. 36, no. 1-2, pp. 55--70, 2008.

[30] T. D. Peter, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th annual meeting on association for computational linguistics*, Stroudsburg, PA, United states, 2002.

[31] J. Qi, Y. Yu, L. Wang and J. Liu, "K*-means: An effective and efficient k-means clustering algorithm," in *International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking*, 2016.

[32] I. Czarnowski and P. Jedrzejowicz, "Data reduction algorithm for machine learning and data mining," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2008.

[33] T. Howley, M. G. Madden, M.-L. O'Connell and A. G. Ryder, "The effect of principal component analysis on machine learning accuracy with high dimensional spectral dat," in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, London, 2005.

[34] M. Nick, "An Introduction to Machine Learning Theory and Its Applications: A Visual Tutorial with Examples," 2016.

[35] R. Sushmita, "Movie Recommendation System using Simple Recommender-Based Approach," in *International Journal of Innovative Science and Research Technology*, 2019.

[36] R. Banik, "The movies dataset- movie recommendation system," [Online]. Available: https://www.kaggle.com/movie-recommender-systems.