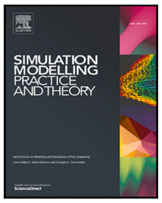
Simulation Modelling Practice and Theory 113 (2021) 102375

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat



Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network

Yassine Afoudi, Mohamed Lazaar [*], Mohammed Al Achhab

ENSIAS, Mohammed V University in Rabat, Morocco
ENSA, Abdelmalek Essaadi University in Tetuan, Morocco

ARTICLE INFO

Keywords:
Recommender systems
Collaborative Filtering
Content-based filtering
Hybrid system
Clustering
Neural network

ABSTRACT

Recommendation systems are information filtering tools that present items to users based on their preferences and behavior, for example, suggestions about scientific papers or music a user might like. Based on what we said and with the development of computer science that has started to take an interest in big data and how it is used to discover user interest, we have found a lot of research going on in the area of recommendation and there are powerful systems available. In the unsupervised learning domain, this paper introduces a novel method for creating a hybrid recommender framework that combines Collaborative Filtering with Content Based Approach and Self-Organizing Map neural network technique. By testing our system on a subset of the Movies Database, we demonstrate that our method outperforms state-of-the-art methods in terms of accuracy and precision, as well as improving the efficiency of the traditional Collaborative Filtering methodology.

1. Introduction

Nowadays, with the ever increasing volume, complexity and availability of online information, recommendation systems have been an effective solution to overcome such information overload [1][2].

The main objective of a recommender system is to predict user preferences, in another way system that offer specific items to users among a large number of choices according to their tastes. Suggestions for movies on Netflix, or products on Amazon, or Videos on YouTube, are real-world examples of using recommender systems in our life [3].

In the mid-1990s, recommendation systems became a famous topic of research. Awareness in recommendation systems has grown significantly in recent years, and recommendation systems now play a significant role in commercial websites and well-known businesses such as Spotify, Facebook, LinkedIn, and IMdb … because these systems help them to increase the number of items sold and sell more diversified items or even increase the user satisfaction. This demand for this type of systems has given the green light to researchers to develop powerful systems and several researches have been carried out in this field [4].

A recommendation method must demonstrate that a product needs to be recommended in order to find relevant products for the user. There are many kinds of recommendation algorithm approaches for this, the most prominent of which are collaborative filtering, content-based filtering, and hybrid systems.

Recommendations to the active client in the Collaborative Filtering approach are focused on items/products that other users with common preferences have enjoyed in the past. Many users' taste similarity is determined based on the similarity of ranking data (scale of 1 to 5 for movies) or the users' browsing history. The Content-Based model suggests products to the active user based

[*] Corresponding author at: ENSIAS, Mohammed V university in Rabat, Morocco.
E-mail addresses: yassine.afoudi@gmail.com (Y. Afoudi), mohamed.lazaar@um5.ac.ma (M. Lazaar), m.alachhab@uae.ac.ma (M. Al Achhab).

https://doi.org/10.1016/j.simpat.2021.102375
Received 5 April 2021; Received in revised form 17 June 2021; Accepted 18 July 2021
Available online 24 July 2021

on items he has already enjoyed. The resemblance of the products is determined by the characteristics associated with the compared items, such as the context, title, or even the product picture. For example, if a user gives a song in the HipHop genre a favorable rating, the machine can learn to recommend other songs in that genre. Hybrid methods [5][6] are built on combining existing strategies to benefit from the advantages of both and the fewest disadvantages. The most well-known and widely used solution is to incorporate Collaborative Filtering with other approaches [7].

We previously suggested a novel intelligent recommendation system [8] that integrates collaborative filtering and K-means clustering in our earlier work. Also, when items (movies) are grouped by genre attributes using K-means and users are categorized based on their item preferences and the genres they prefer to watch, we employed specific user demographic attributes such as gender and age to construct segmented user profiles [9]. Then The Collaborative Filtering technique is used to the cluster where the user belongs to recommend items to an active user.

In this paper, we propose a powerful recommendation method based on a Hybrid approach that blends Collaborative Filtering with Content-Based and is supervised by the ranking list provided by the self-organizing map, a well-known unsupervised learning artificial neural network technique.

The following is a summary of the paper's structure. The following section discusses related work in the recommendation area. The adaptive solution and the various steps to construct our system are presented in Section 2. Sections 3 and 4 describe the methods used to compare our system to other state-of-the-art approaches in the field of recommendation and analyze the results.

2. Related work

Several research efforts have been made to combine different recommendation approaches. Andreu Vall et al. [5] offers a hybrid recommender system combining two feature combination (profiles and membership) for the automated continuation of music playlists, they proposed a system that extends collaborative filtering by considering not only playlists organized by hand, but also by incorporating any type of song functionality vector. To boost prediction accuracy, R. Logesh et al. [10] suggest a customized context-aware hybrid travel recommender framework based on users' qualitative knowledge and opinion mining techniques. R. Kiran et al. [11] suggest a novel deep learning hybrid recommender algorithm that uses embeddings for describing users and items to learn non-linear latent factors and solves the cold start problem by incorporating certain users/items data into a very deep neural network. Masoumeh Riyahi et al. [12] offers a hybrid recommender system for discussion groups, their proposal comes from the observation that the majority of discussion group recommender schemes rely on mutual filtering mechanisms, while others use content-based or mixed filtering. As a result, they suggest a recommender mechanism that combines a Content-Based solution based on tagging with Collaborative Filtering that uses the users' implicit scores. Zafran Khan et al. [13] suggest a Deep Semantic based Topic focused Hybrid RS model for a hybrid recommender system that uses item definition semantics inspired by its topics content.

In our system, a new hybrid recommender system is represented, which is based on four parts, namely Collaborative, Content-Based, SOM Collaborative Filtering and Hybrid Model. Implicit user ratings are calculated using the singular value decomposition approach in the collaborative filtering part, we use also the items textual features to build a content-based model. Implicit ratings data from users and movie features are used in Self-Organizing Map with collaborative filtering to create the SOM CF model, then the results of these three parts are combined in the hybrid model part of the proposed system to recommend similar top-N movies to the active user selected.

3. Proposed work

Famous movie companies such as IMDB and Netflix have conducted extensive studies into the use of recommendation services in the film industry, taking into account both content and customer details in their recommendations. This section proposes a modern hybrid recommendation model with four key components: collaborative filtering, content-based filtering, SOM collaborative filtering and Hybrid filtering.

3.1. Collaborative filtering model

Collaborative filtering (CF) [14][15][16] is a method for providing suggestions based on correlations between users and products. In other words, it is the method of filtering items based on the opinions of other users and choosing a group of users with similar tastes to a specific user. The method analyzes their favorite products and integrates them into a categorized list of suggestions. The CF system tries to find similar items based on user feedback [17]. User feedback can be either explicit or implicit, explicit as a numerical rating to specify how much users liked a particular item, for example 1 means dislike or 5 if the user likes the item very much, or implicit like browsing history on the website or reading time a type of product … . Collaborative Filtering have two types of algorithms, Memory-Based and Model-Based, The first type saves products and user data in memory, then uses mathematical methods to make estimates based on the data. Different machine learning algorithms, such as the Bayesian network, rule-based, and clustering methods, are used to construct the model process. In our approach we will use the second category, Model-Based, because it can response user's request instantly. Singular Value Decomposition (SVD) is the approach we used to create a Collaborative Filtering model. SVD is a matrix factorization technique that reduces the number of features in a dataset by decreasing the space dimensions from A to B, where A is smaller than B. Since we are interested in the matrix factorization aspect of recommendation structures that maintains the same dimensionality, the key function of SVD is to decompose a matrix into three other matrices as seen below:

$$X = U \times S \times V^T \tag{1}$$

Where X is a $M \times N$ utility matrix and U is a $M \times R$ orthogonal left singular matrix, which shows the relationship between consumers and latent factors in our case, with the latent factors being the item characteristics. V is a $R \times N$ diagonal right singular matrix that represents the similarities between elements and latent factors. S is a $R \times R$ diagonal matrix that defines the strength of each latent factor. We use a matrix structure to construct our collaborative filtering model, where each row represents a user and each column represents a product, and the elements of this matrix are the ranking data that the user gives to the item. We then use the SVD method to get all the expected ratings by the users by taking the dot product of U, S, and V^T, as seen in Eq. (2). To put our model to the test, we take an user and rank all of his expected ratings to suggest the top-N products.

$$\hat{X} \approx U \cdot S \cdot V^T \tag{2}$$

3.2. Content-based model

The features or content of the items you want are referred to as "content" here. The aim of content-based filtering is to group products with similar attributes, consider the user's preferences, and then look for those terms in the dataset [18][19]. Finally, we suggest different items with similar attributes.

We want to add a weight to each word in the item description in our content-based model to assign the value of that word in the dataset. We use the TF–IDF algorithm to weight a keyword in any document (movie title) and attribute value to that keyword depending on the number of times it appears in the document; a higher TF–IDF score (weight) indicates that the phrase is rarer and more significant, and vice versa. As a result, each item will be interpreted by a TF–IDF vector dependent on title features at the end of the process.

Term Frequency (TF) is the total number of times a word appears in the current document. This means the occurrence of the word in a document, when the frequency of a word is higher, it gives higher weight, this why we should normalize the result using a division by the length of the document as shown in the formula below where $N(T_x, D_y)$ is the total number of times term T_x appears in a document D_y and $N(P_y)$ is the total number of terms in the document.

$$TF(T_{x,y}) = \frac{N(T_x, D_y)}{N(P_y)} \tag{3}$$

Inverse Document Frequency of a word is the cumulative number of records containing the word x; it indicates the scarcity of the word as the IDF decreases when the word occurs in the text. It aids in determining the significance of a word across the whole corpus. For more explanation of the IDF, we take an example, if we do a search on google on "the hybrid system", automatically the TF of the word "the" will be higher than "hybrid" and "system", here the role of the IDF came to reduce the weight of the word "The" to give more weight to the important words. IDF can be calculated by the formula below where $N(D)$ is the total number of documents and $N(D, T_x)$ is the number of documents containing term x.

$$IDF(T_x) = \log\left(\frac{N(D)}{N(D, T_x)}\right) \tag{4}$$

Ultimately, TF–IDF is a normalization measure used to evaluate the importance of a word for a document in a corpus of documents. The formula for calculating the TF–IDF is.

$$TF\text{-}IDF(T_{x,y}) = TF(T_{x,y}) \times \log_{10}\left(\frac{N(D)}{N(D, T_i)}\right) \tag{5}$$

We obtain a matrix containing each word and its TF–IDF score in our framework after computing the TF–IDF score for the title of each movie in our case, word by word, thus avoiding stop terms. We create a profile for each user in the dataset by selecting all of the products that the user has scored favorably, classifying the words by TF–IDF ratings, and finally generating a list of words with their scores that represents the user's profile.

Right now we have all the items represented in a vector space model containing each word with the TF–IDF score and the same as the user profiles, here we use the cosine similarity approach as indicated in the Eq. (6) between the user profile and the items vector space model to find items similar to the user profile and give him the recommendation after classified the results.

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{6}$$

3.3. Neural network model

Neural networks are a group of algorithms that detect patterns and are closely modeled after the human brain. They use a kind of machine vision to classify sensory data, naming or sorting raw data. Both real-world records, whether images, sound, text, or time series, must be converted into the patterns they understand, which are digital and stored in vectors. We should think of neural networks as a layer of clustering and classification on top of the data we store and handle, as they assist us in clustering and classifying data [20]. The self-organized map is a kind of neural network, and it is the technique we will be using in this article.

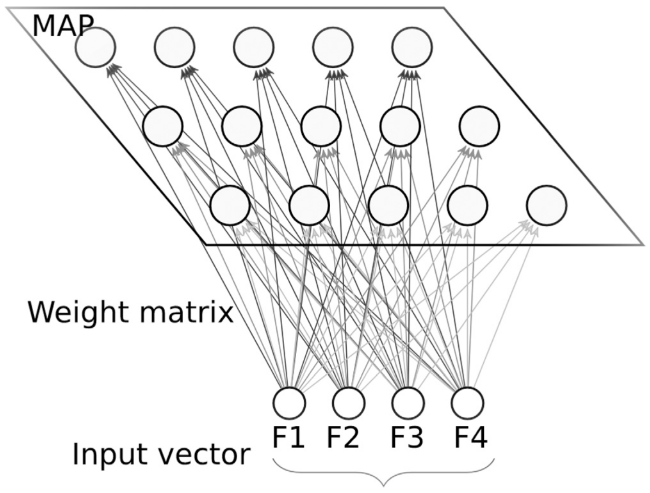


Fig. 1. SOM network architecture.

Table 1
Part of the movies dataset.

id	Title	Action	Adventure	Animation	Comedy	…	War
1	Toy Story	0	0	1	1	…	0
2	DoldenEye	1	1	0	0	…	0

As explained in Collaborative Filtering module, CF is the process of filtering items based on users' historical opinions and preferences on a set of items. Here, we will use the Self-Organizing Map (SOM) method to improve the traditional collaborative filtering system in order to build our hybrid system. A self-organizing map (SOM) is a form of artificial neural network (ANN) that is trained using unsupervised learning to generate a low-dimensional, usually two-dimensional. We will use the self-organizing map in our model to solve the issue of unsupervised clustering of the movie dataset. Clustering technology simplifies the structure of the dataset and divides it into different clusters, so the users can easily observe and analyze the data.

The map is made up of a standard grid of "neurons", which are processing units. Each unit is linked to a function vector that represents a high-dimensional observation model. Using a limited number of models, the map tries to reflect all available findings as accurately as possible. Neighbors are map groups that are close by on the grid. The model vectors are structured such that the local map units represent a common type of data and the distant map units represent various types of data after generating a map for a specific dataset. Fig. 1 shows the architecture of a SOM network.

Our system classifies movies using SOM based on genre of the movies as shown in Table 1. After initialization of our map dimensions and randomly initialization SOM weights we train the model. Once the map has been trained, its give us weights as results, then we use those weights as input data of K-means clustering model. Here, to find the appropriate number of clusters, we will use the Elbow method the best known method for choosing the number of clusters, this method says that by plotting the different number of clusters according to the variance, the point of the elbow is that of the number of clusters whose variance no longer decreases. To make recommendation, we classify the dataset of movies in a specific number of clusters, we choose for an active user all ratings of the positively rated movies considering only the movies with rating value greater than or equal to 2.5, after that, we would like to give each movie its cluster number, to do this, we calculate the distance between the items attributes and the centroids of k-means using the Euclidean distance approach as mentioned in Eq. (7). Then we choose the cluster with the smallest distance result. We measure the number of each movie cluster in the list after listing the positively scored movies with their cluster class, and then use the highest result class as our user's favorite class. After getting our user's favorite class, we get their age (demographic attribute) if the age greater than or equal to 35 years, we get all users with the same age interval and vice versa, then we build a matrix of all selected users and all selected movies and we use the SVD collaborative filtering approach to predict items ratings and sort them from best to worst to give recommendations.

$$d(a,b) = d(b,a) = \sqrt{\sum_{i=1}^{n}(b_i - a_i)^2} \tag{7}$$

3.4. The proposed hybrid model

To take advantage of the complementary advantages of two or more recommendation methods, hybrid recommendation models merge them in various ways. Table 2 shows some of the combination methods that have been used. In our architecture, we will use

Table 2
The most-known hybridization methods.

Hybridization method	Description
Weighed	The scores of different recommendation systems combined together to produce a single recommendation.
Switching	Depending on current situation, the system switches between recommendation techniques.
Mixed	Recommendation from several different recommenders are presented at the same time.
Cascade	One recommender refines the recommendations given by another.
Feature augmentation	Output from one technique is used as an input feature to another.

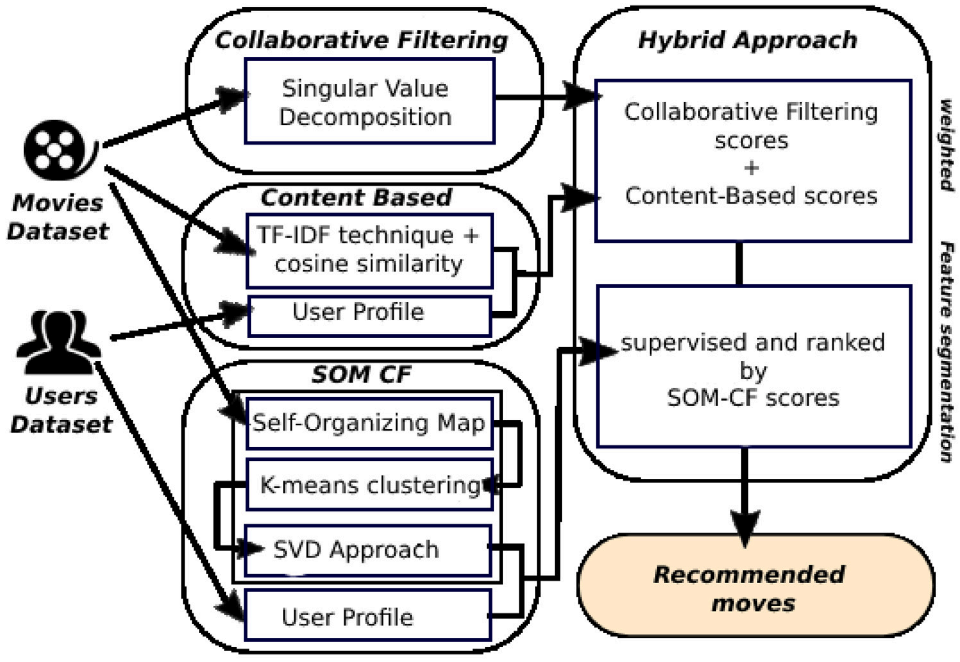


Fig. 2. The adopted architecture.

two methods in one, Feature augmentation and Weighted methods as shown in Fig. 2. After obtaining the results of the Collaborative Filtering model and the Content-Based model as presented in Table 3, we combine them with a linear combination of their scores, as determined in Eq. (8) (weighted method), then we classify the list and we take the first 200 recommended items, finally, we use the selected results and we classify them again according to the Self-organizing map CF scores (Feature augmentation) from best to worst and show the first N items as recommendation.

$$Hybrid_{weighted} = U + V \tag{8}$$

4. Result and discussion

4.1. The dataset

To evaluate our system, we use the Movielens100k dataset because it is publicly available and widely used in the evaluation of recommendation models.

The dataset contain 100.000 rating divided into 90570 rating for the train set and 9430 for the test set, the ratings are values from 1 to 5 scale, 1 mean movies negatively rated and 5 movies positively rated, all ratings exist in our dataset are distributed as shown in Fig. 3.

All ratings are given by 943 users on 1682 movies, the selected users must have rated at least 20 films and have some demographic attributes such as gender, age, zip code, occupation …

4.2. Implementation

4.2.1. Experimental steps

The first step is to import the dataset into our work space, which after we split the user interaction dataset (rating dataset) into two parts: 75% (90570) as a training set and 25% (9430) as a test set.

Table 3
The first three recommended movies for user id = 3.

Model name	Top 3 recommended movies	Scores
Collaborative Filtering (CF)	Titanic	2.281
	L.A. Confidential	2.253
	Air Force One	2.021
Content-Based	Paradise Lost	0.601
	Homeward Bound II	0.567
	The City of Lost Children	0.566
SOM CF	Chasing Amy	2.444
	Good Will Hunting	2.276
	The English Patient	1.952

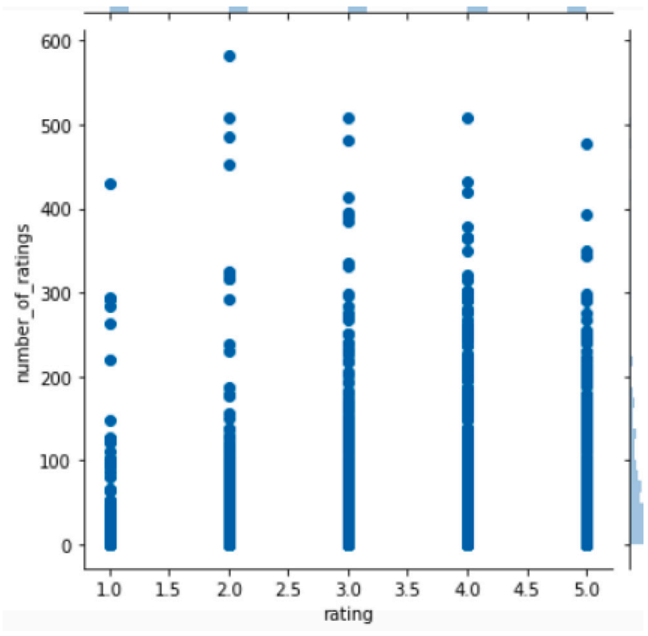


Fig. 3. Distribution of ratings data.

The second step, we clean the movies dataset by removing all unused features such as release date, imdb url . . . and we take just the id of the movie, the title and the genres attributes. After that, we prepare the collaborative filtering model by building a matrix containing all available users and movies and its values are the rating data given by each user to a movie, then we applied the singular value decomposition as explained above to create collaborative filtering scores for each user. The next step is to create the Content-Based model using the TF–IDF on the title of each movie after ignoring the English stop words to represent the movies in a vector space model whose max vector size is 60, composed of the main unigrams and bigrams found in the dataset, then we create a user profile by taking all of its movies positively rated (greater than or equal to 2.5) and build a TF–IDF matrix with selected movies, then get words with strengths weighted average to be represented words for a user, finally, we use the cosine similarity between the user profile and all the movie profiles (TF–IDF matrix) and use the similarity results as scores of our Content-Based model. To build our third model we use only movies genre attributes as input features of our 10×10 Self-organizing map and we train randomly the model with 1000 iterations and learning rate equals to 0.9, it should be mentioned that we also tried to use the k-means clustering algorithm, but we gave up this method because the classification results and the recommendation time speed obtained were not good enough as shown in Figs. 4 and 5. After forming our SOM model, we use its output as input to K-means clustering to classify all of the data into a certain number of clusters. We chose $k = 7$ as the number of our cluster by plotting the 8 different numbers of clusters according to variance, using the Elbow method discussed above and observing Fig. 6. After that, we represent each movie by the cluster with the smallest Euclidean distance between the movie features and the centroids of the clusters, then to build our model scores, we take for each user his favorite class according to the best cluster movies positively rated and we also use age segmentation to improve this model, then we build a SVD collaborative filtering model with movies in the same cluster and selected segmentation users for get a list of SOM CF scores.

Last step is to build our hybrid model combining the three models, in the first, we combine collaborative filtering with content based on a linear combination of their scores, and we classify the results from highest value to the smallest, then we take the first

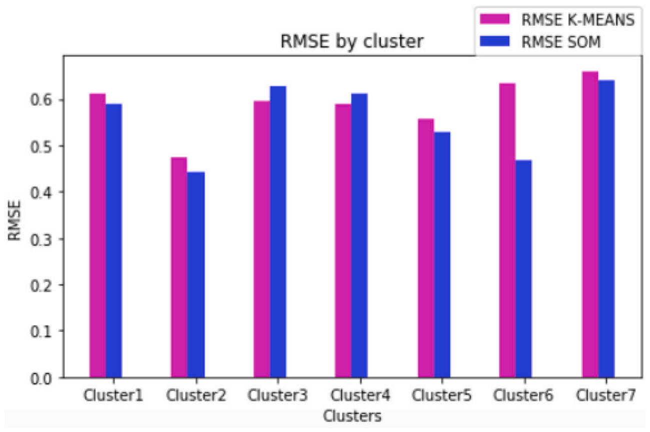


Fig. 4. The RMSE results for CF based on K-means and SOM.

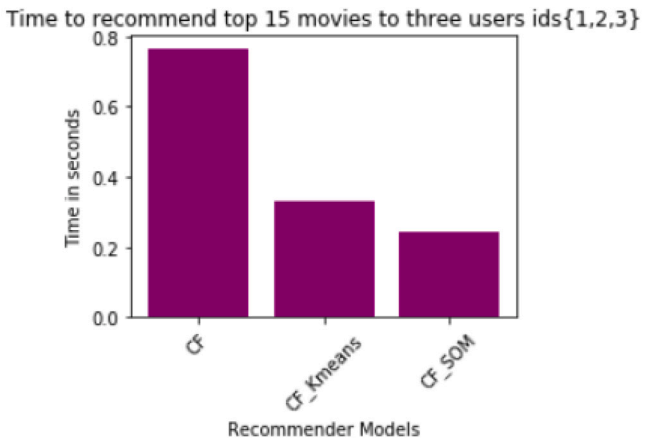


Fig. 5. Time comparison to recommend 15 movies to three users.

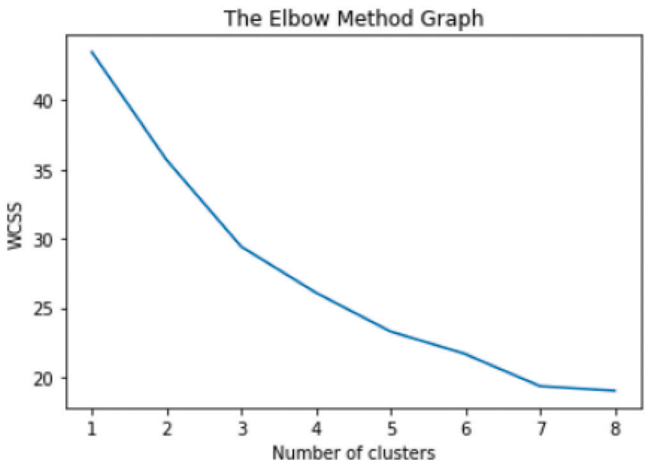


Fig. 6. The Elbow graph.

200 movies and we rank them again using SOM CF scores and finally we give the best recommended N movie to the active user after ignoring all previously viewed movies.

Table 4
The top-3 recommended movies for first three users using the proposed system.

User		Top 3 recommended movies	
Id	Age	Movie id	Title
1	24	191	Amadeus
		7	Twelve Monkeys
		100	Fargo
2	53	286	The English Patient
		313	Titanic
		275	Sens and sensibility
3	23	268	Chasing Amy
		272	Good Will Hunting
		286	The English Patient

After building our model, we want to test and evaluate the results of the recommendations by finding out whether the recommended items to an active user in the training phase are included in the list of items that the same user has seen and positively rated in the test set.

4.2.2. Experimental platform

Both of our experiences were compiled in a Jupyter notebook and implemented in Python because that includes several modules that make the implementation of various concepts easier. All of the tests were performed on a MacBook Pro with an Intel Core i7 processor running at 2.7 GHz and 8 GB of RAM.

4.2.3. Experiment results

Following the training of our three models, each framework offers us with a list of suggested movies along with their score values, which we then use to build our hybrid system. We measure the effects of our model on all users of the dataset using time-based recommendation speed and some current offline methods with sophisticated and state-of-the-art approaches to recommendation. Table 4 shows the results of the experiment for the first three users.

4.2.4. Evaluation

A recommendation system typically produces an ordered list of recommendations, from best to worst, for each user in the dataset. In fact, in many cases, the user does not much care about the order of recommended items, a few good suggestions are enough. But for us, we need to evaluate our system, because an evaluation technique provides insight into the relevance of the list of recommended items. Therefore, there are many evaluation approaches involved in the area of the recommendation system divided into online and offline methods [21]. For those online, we need a real-time user to give feedback and opinions on the recommended items, in our work we will use the offline technique from a dataset of real user interactions on movies.

We choose RMSE and Precision–Recall at k techniques from among several assessment approaches in this article. The Root Mean Squared Error is a well-known technique for evaluating the accuracy of a recommender system based on ratings data in order to look for low prediction errors. The principle of this technique is based on the use of predicted and real ratings, then calculating the average of the errors of the test set using Eq. (9), where P is the predicted rating and R is the true rating, and produce a final score; we then compare our model's result to that of others; if your result is less, it indicates that your model is more accurate.

$$RMSE = \sqrt{\frac{\sum_{ratings}(P-R)^2}{\#ratings}} \tag{9}$$

The other offline method used in our article is the Precision–Recall at k technique, where k is a number that can be defined by the user to much the objective of the Top-N recommendation. Recall is the capability of the model to find all relevant items and recommend them to the user while the Precision is the ability of the model to provide the relevant items with the fewest recommendations. The precision and recall to k are then determined as formulas (10) and (11).

Where R_i represents the number of relevant recommended items at k, Tr represents the total number of relevant items, and Nr represents the total number of recommended items at k.

$$Recall@k = R_i/Nr \tag{10}$$

$$Precision@k = R_i/Tr \tag{11}$$

After calculating the precision and the recall at k, to facilitate the analysis of the result, we normalize the techniques by defining the F-measure as below.

$$F\text{-}measure@k = (2 \times P@k \times R@k)/(P@k + R@k) \tag{12}$$

Once we have built our system, we will test it to see which model gives us the best performance and accuracy. We test our system on the movielens100k database and use the precision–recall approach at k as discussed below. Fig. 7 displays the plot of the average

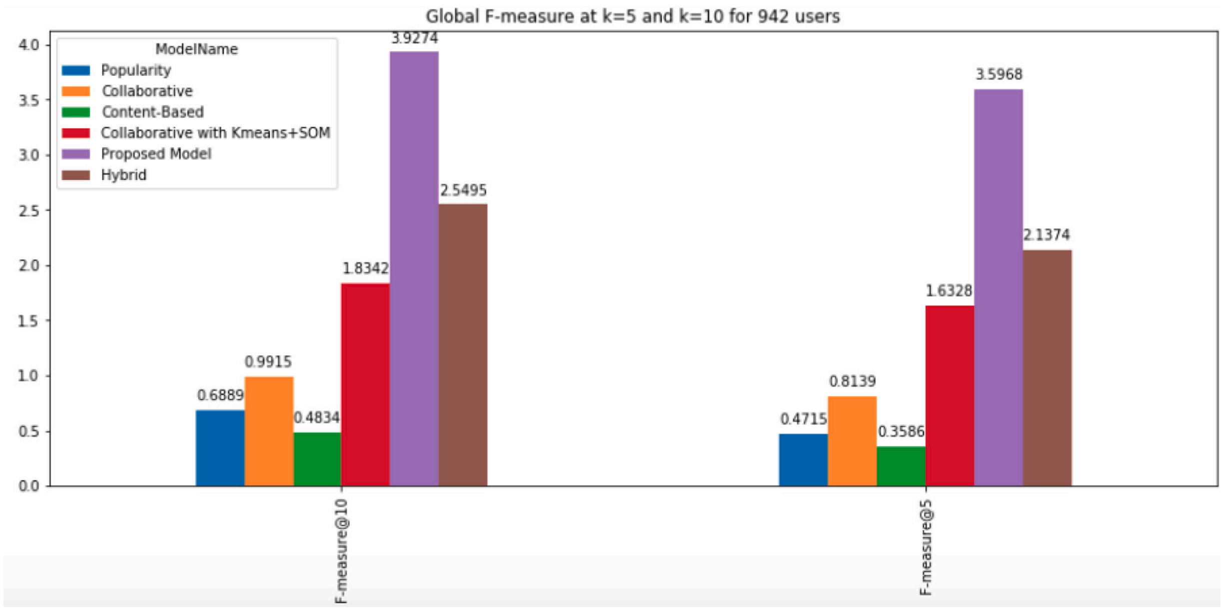


Fig. 7. The mean F-measure plot of all models at K = 5 and K = 10.

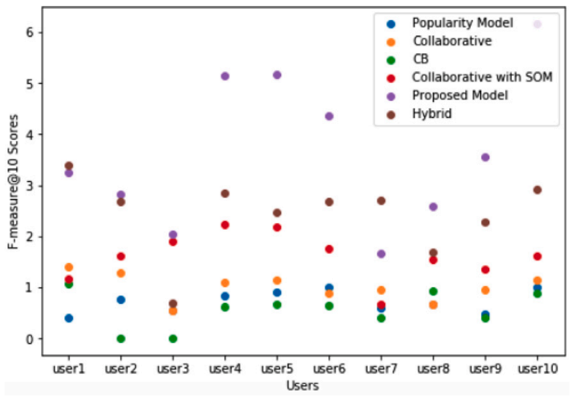


Fig. 8. The F-measure scatter plot of first ten users.

F-measure at $k = 10$ and $k = 5$, where k is the number of suggested movies for all users in the data collection, on the other hand, Fig. 8 represent the F-measure scatter graph of the first ten users, and Fig. 9 displays the accuracy curve of the first 20 users.

Finally, because of the high number of features and parameters, recommendation systems demand some space and time to generate suggestions. We can extrapolate from the findings that combining traditional collaborative filtering with a content-based technique, both supervised by our SOM-CF model's ratings, improves precision and accuracy greatly. Despite the fact that our hybrid model is slow in comparison to other structures, its accuracy necessitates its existence. Obviously, if an item lacks descriptive properties, it cannot be used in the Self-organizing map to determine which cluster it belongs to.

5. Conclusion and future work

In this paper, we propose a new hybrid model of a movie recommendation framework based on three models: collaborative filtering, content-based, and CF with a self-organizing map model that takes the age demographic attribute into account. The main advantages of our system is to combine all the scores of all models and benefit from the advantages of each of them. Even our system takes a lot of recommendation time speed compared to the other models but the precision and the performance improvement are very high.

The experiment shows that by using Self organizing map with collaborative filtering, the RMSE was reduced in the majority of clusters against of using K-means clustering with CF, then we combine this powerful system with the other state of art approaches to create a Hybrid system.

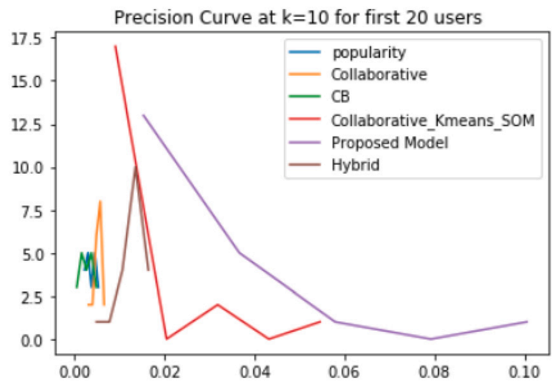


Fig. 9. The Precision curve at $k = 10$ for first 20 users.

Precision–Recall is an offline method to evaluating recommendation systems that uses actual user data obtained from an online movie website to test our system. The findings of this approach suggest that the proposed system will increase the quality and performance of movie recommendations. This indicates that the approach we recommend is feasible.

We see a variety of possible future paths for our work, in addition to the framework enhancements outlined in the results and discussion section. We will continue to refine and investigate other approaches in the field of recommendation, and machine learning and deep learning algorithms will be used to change and improve our model.

References

[1] F. Maxwell Harper, Joseph A. Konstan, The MovieLens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (4) (2016) 1–19.
[2] Shankar Vembu, Stephan Baumann, A self-organizing map based knowledge discovery for music recommendation systems, in: Computer Music Modeling and Retrieval, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2005, pp. 119–129.
[3] Meehee Lee, Pyungseok Choi, Yongtae Woo, A hybrid recommender system combining collaborative filtering with neural network, in: Adaptive Hypermedia and Adaptive Web-Based Systems, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2002, pp. 531–534.
[4] Duwairi Rehab, Hadeel Ammari, An enhanced CBAR algorithm for improving recommendation systems accuracy, Simul. Model. Pract. Theory 60 (2016) 54–68.
[5] A. Vall, M. Dorfer, H. Eghbal-zadeh, M. Schedl, K. Burjorjee, G. Widmer, Feature-combination hybrid recommender systems for automated music playlist continuation, User Model. User-Adapt. Interact. 29 (2019) 527–572.
[6] E. Çano, M. Morisio, Hybrid recommender systems: A systematic literature review, Intell. Data Anal. 21 (2017) 1487–1524.
[7] Jakomin Martin, Tomaž Curk, Zoran Bosnić, Generating inter-dependent data streams for recommender systems, Simul. Model. Pract. Theory 88 (2018) 1–16.
[8] Afoudi Yassine, Lazaar Mohamed, Mohammed Al Achhab, Intelligent recommender system based on unsupervised machine learning and demographic attributes, Simul. Model. Pract. Theory 107 (2021) 102198, http://dx.doi.org/10.1016/j.simpat.2020.102198.
[9] Yassine Afoudi, Mohamed Lazaar, Mohamed A. Achhab, Collaborative filtering recommender system, in: Advanced Intelligent Systems for Sustainable Development, AI2SD'2018, in: Advances in Intelligent Systems and Computing, Springer International Publishing, Cham, 2018, pp. 332–345.
[10] R. Logesh, V. Subramaniyaswamy, Exploring hybrid recommender systems for personalized travel applications, in: Cognitive Informatics and Soft Computing, Springer, Singapore, 2019, pp. 535–544.
[11] R. K, P. Kumar, B. Bhasker, DNNRec: A novel deep learning based hybrid recommender system, Expert Syst. Appl. 144 (2020) 113054.
[12] M. Riyahi, M.K. Sohrabi, Providing effective recommendations in discussion groups using a new hybrid recommender system based on implicit ratings and semantic similarity, Electron. Commer. Res. Appl. 40 (2020) 100938.
[13] Z. Khan, N. Iltaf, H. Afzal, H. Abbas, DST-HRS: A topic driven hybrid recommender system based on deep semantics, Comput. Commun. 156 (2020) 183–191.
[14] Z. Xian, Q. Li, G. Li, L. Li, New collaborative filtering algorithms based on SVD++ and differential privacy, Math. Probl. Eng. 2017 (2017).
[15] C.C. Aggarwal, Model-based collaborative filtering, in: Recommender Systems: The Textbook, Springer International Publishing, Cham, 2016, pp. 71–138.
[16] J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in: The Adaptive Web: Methods and Strategies of Web Personalization, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2007, pp. 291–324.
[17] Yassine Afoudi, Mohamed Lazaar, Mohamed Al Achhab, Impact of feature selection on content-based recommendation system, in: 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems, WITS, 2019, pp. 1–6.
[18] M.J. Pazzani, D. Billsus, Content-based recommendation systems, in: The Adaptive Web: Methods and Strategies of Web Personalization, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2007, pp. 325–341.
[19] D. Wang, Y. Liang, D. Xu, X. Feng, R. Guan, A content-based recommender system for computer science publications, Knowl.-Based Syst. 157 (2018) 1–9.
[20] Fadoua Oudouar, Mohamed Lazaar, Zaoui El Miloud, A novel approach based on heuristics and a neural network to solve a capacitated location routing problem, Simul. Model. Pract. Theory 100 (2020) 102064.
[21] F. Hernández del Olmo, E. Gaudioso, Evaluation of recommender systems: A new approach, Expert Syst. Appl. 35 (2008) 790–804.