Demographic information combined with collaborative filtering for an efficient recommendation system

Sana Nabil, Mohamed Yassin Chkouri, Jaber El Bouhdidi
SIGL Laboratory, ENSATE, Abdelmalek Essaadi University, Tetuan, Morocco

Article Info

Article history:

Received Jan 20, 2024
Revised Jul 4, 2024
Accepted Jul 9, 2024

Keywords:

Clustering
Collaborative filtering
Demographic information
Factorization matrix
K-means
Recommender system
User's reviews

ABSTRACT

The recommendation system is a filtering system. It filters a collection of things based on the historical behavior of a user, it also tries to make predictions based on user preferences and make recommendations that interest customers. While incredibly useful, they can face various challenges affecting their performance and utility. Some common problems are, for example, when the number of users and items grows, the computational complexity of generating recommendations increases, which can increase the accuracy and precision of recommendations. So, for this purpose and to improve recommendation system results, we propose a recommendation system combining the demographic approach with collaborative filtering, our approach is based on users' demographic information such as gender, age, zip code, occupation, and historical ratings of the users. We cluster the users based on their demographic data using the k-means algorithm and then apply collaborative filtering to the specific user cluster for recommendations. The proposed approach improves the results of the collaborative filtering recommendation system in terms of precision and recommends diverse items to users.

Corresponding Author:

Sana Nabil
SIGL Laboratory, ENSATE, Abdelmalek Essaadi University
Tetuan, Morocco
Email: sana.nabil@etu.uae.ac.ma

1. INTRODUCTION

The advent of the internet, social media, big data, and predictive algorithms has profoundly influenced the collection and application of demographic information. Today's consumers generate substantial data, gathered extensively by social media platforms, third-party data collectors, retailers, and financial transaction processors. Integration with artificial intelligence enables the prediction and targeted marketing of consumer choices and purchasing preferences with exceptional accuracy, based on demographic traits and historical behaviors. In recent years, recommender systems have gained increasing importance with the growth of social media, YouTube, Amazon, Netflix, and various other web services. E-commerce platforms use recommender systems to suggest items that may interest customers. Today, recommender systems are an integral part of our daily online experiences.

Recommender systems can utilize three well-known approaches: collaborative filtering, content-based filtering, and hybrid methods. Additionally, they can use a demographic approach, categorizing users based on their attributes and recommending movies or other items using their demographic data [1]. In the collaborative filtering method, recommendations are made for each user by comparing their preferences with those of other users who have shown similar preferences in the past. Collaborative filtering (CF) operates on the principle that individuals who have shared similar ratings or evaluations of items in the past are likely to continue agreeing in their future evaluations. CF methods are grouped into two general methods:

neighborhoods-based and model-based [2]. Content-based filtering methods rely on detailed descriptions of items and profiles of users' preferences [3]. Recommendations are made by comparing the content of the items with the user profiles. Each item's content is represented by descriptors, terms, or feature vectors, such as a genre for a movie or a frequent term in a document. The content-based filter analyzes these descriptors for items previously rated by the user. This information is then used to construct a user interest model, which generates recommendations [4]. The hybrid recommender system combines two or more of the mentioned approaches in the same system [5] and combines the content-based approach with collaborative filtering. The two approaches can be used independently [6].

Many other approaches are used to build recommender systems, including sentiment analysis, which utilizes users' opinions [7], [8] and reviews. Many works incorporate sentiment analysis into hybrid recommendation systems, such as those using collaborative filtering, to improve the system's performance [9]–[11]. Recommender systems can also be based on demographic information, which categorizes users according to their attributes and recommends movies using their demographic data [1]. This approach uses the user's demographic profile, such as nationality, age, gender, and other factors, to classify users and provide personalized recommendations.

Traditional approaches such as collaborative filtering have limitations, including the cold start problem for new users or items. Additionally, as the number of users and items increases, it becomes challenging to find similar users or user groups. The computational complexity of generating recommendations can increase the accuracy and precision of recommendations, but it can be time-consuming due to the diversity of opinions among many users. In this context, combining collaborative filtering with other approaches, such as the demographic approach, has been shown to help solve the cold start problem [12]. Additionally, combining collaborative filtering with the demographic approach enhances the precision of collaborative filtering [13], [14].

Many studies in the field of recommender systems focus on improving results achieved through collaborative filtering. Aljunid and Manjaiah [15] propose a movie recommender system based on the alternating least squares (ALS) algorithm using Apache Spark. Their work emphasizes selecting ALS parameters that can influence the performance of a recommendation system. By using collaborative filtering, they predict user ratings for specific movies based on users' historical ratings. The system is evaluated using the MovieLens dataset, achieving the best root mean squared error value (RMSE) of 0.9167. Silva *et al.* [16] present a study comparing three algorithms: non-negative matrix factorization (NMF), singular value decomposition (SVD), and stacked autoencoders (SAE). They evaluated the algorithms using the MovieLens 100K dataset. The study found that the best RMSE results were achieved with NMF (0.923), SVD (0.901), and SAE (0.723). For mean squared error (MSE) values, the results were NMF (0.724), SVD (0.708), and SAE (0.213) [16]. Chen *et al.* [17] propose a recommender system based on collaborative filtering, which utilizes user correlation and evolutionary clustering. In the first stage, the score matrix undergoes pre-processing, including dimensionality reduction and normalization. A clustering principle is employed to create a dense score matrix, followed by dynamic evolutionary clustering. Similar users are grouped into clusters based on their interests and similarities, and the correlation between users is measured to calculate distances based on user satisfaction and potential score information. Finally, each group uses collaborative filtering to predict ratings. When using the MovieLens 100K dataset, the system achieves RMSE values between 0.95 and 0.96, and mean absolute error (MAE) values between 0.74 and 0.75. For the MovieLens 1M dataset, the RMSE values are around 0.94, and MAE values range between 0.73 and 0.74 [17]. Zarzour *et al.* [18] propose a collaborative filtering recommendation system that integrates k-means clustering and singular value decomposition. The system operates in two main phases. In the first phase, it clusters users' ratings based on their preferences using k-means, reduces the data's dimensions with SVD, and calculates similarities between users. In the second phase, the model created in the first phase is used to produce recommendations for the given active user. The study utilizes the MovieLens dataset, achieving RMSE values between 0.6 and 0.7 [18].

Ziani *et al.* [9] propose a recommendation system that integrates sentiment analysis with collaborative filtering. For sentiment analysis, they employ the support vector machine (SVM) algorithm to classify user reviews, extracting statistical features such as word count, emotionalism, addressing, and reflexivity. These features are then used to compute a polarity score that informs collaborative filtering during the recommendation phase. The system's performance metrics include a MAE of 0.52, precision of 0.96, and recall of 1.0 for English reviews. For French reviews, the results are MAE of 0.50, precision of 1.0, and recall of 1.0. For the unspecified language, the results are MAE of 0.60, precision of 0.90, and recall of 1.0 [9]. Nassar *et al.* [19] propose a recommendation system that enhances collaborative filtering performance through multi-criteria recommendation and deep learning. The proposed model operates in two stages. In the first stage, user and item features are extracted and fed into a deep neural network to predict criteria ratings. In the second stage, the deep neural network learns the relationship between the overall rating and the criteria ratings. The system's performance is evaluated using the TripAdvisor dataset, yielding a mean absolute error (MAE) of

0.7552 ± 0.0050 [19]. Sallam *et al.* [20] integrated collaborative filtering with a sentiment analysis approach to enhance the results of collaborative filtering. They utilized Lexon sentiment analysis to obtain sentiment ratings and then applied collaborative filtering using two methods: k-nearest neighbors (KNN) item-based and SVD. This proposed approach enhanced the accuracy of the Arabic recommendation system, reducing the average error values, and achieving a RMSE of 0.5583 and a MAE of 0.1558.

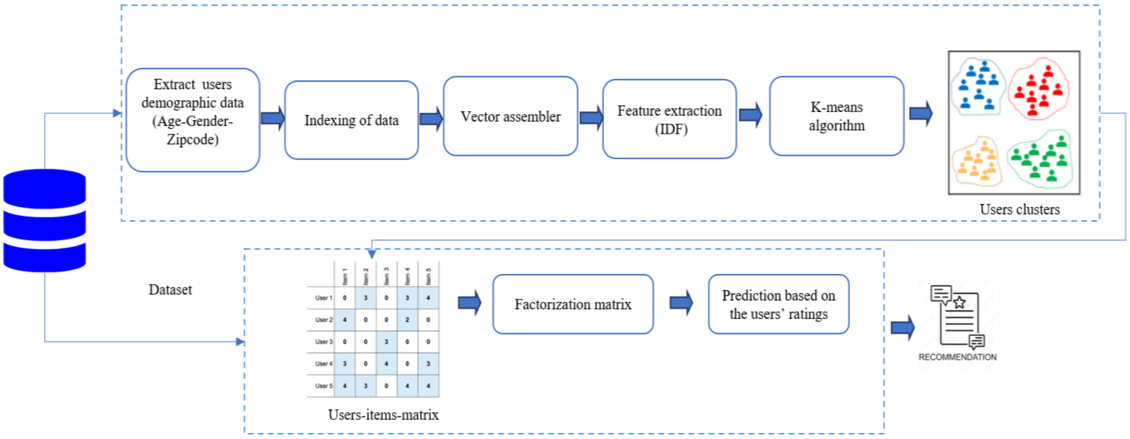
From the previously mentioned studies, we observe that systems that combine collaborative filtering with other approaches such as demographic, content-based, or sentiment analysis tend to yield better results than systems relying solely on collaborative filtering. For this reason, our work proposes a new recommender system that utilizes multiple user demographic attributes. We use a vector assembler and inverse document frequency (IDF) to integrate various user attributes, aiming to achieve improved results.

In this article, we propose a novel approach that combines the demographic method using the k-means algorithm with the collaborative filtering method based on the factorization matrix. As mentioned earlier, hybridizing collaborative filtering with other methods can enhance recommendation results in terms of precision and diversity. The novelty of our approach lies in integrating several demographic attributes, including age, gender, and zip code, through a vector assembler and IDF for feature extraction. The results demonstrate that utilizing multiple user demographic information improves the precision of the recommendation system. This article is structured as follows: In section 2, we describe the detailed architecture and methods of our proposed recommender system. In section 3, we present the experimentations and results, and finally, we conclude with a summary and discussion of the findings.

2. METHOD

2.1. The detailed architecture

Our proposed architecture comprises two main modules: the demographic model, which employs a demographic approach for constructing user clusters or profiles, and the recommendation model, which utilizes a collaborative filtering approach. In the demographic filtering model, we used each user's age, gender, and zip code from the dataset, and we clustered the users based on this information using the k-means algorithm, grouping them into several clusters based on their demographic attributes, the algorithm first clusters the users into male and female groups, then further clusters them based on age and zip code. After constructing user profiles based on their demographic information, we apply the collaborative filtering approach using matrix factorization of user item ratings data for recommendations. Figure 1 illustrates the architecture of our recommender system.



	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	4
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	4

Figure 1. The architecture of our recommender system

2.2. Clustering and profiles construction module

The first part of our module involves constructing profiles using a demographic approach based on users' age, gender, occupation, and zip code. First, we extract users' demographic information from the dataset (age, gender, and zip code) and index the data using a data indexer. For instance, gender data are represented as 'M' for male and 'F' for female, and we index these as 0 and 1, respectively. We combine the data (age, gender, and zip code) into a single vector using a vector assembler. Then, we apply feature extraction using IDF and finally use the k-means algorithm to generate different clusters.

2.2.1. Vector assembler

Vector Assembler is a transformer used to combine multiple columns into a single vector column. This allows for the merging of raw features and features generated by various feature transformers into one unified feature vector, which is useful for training machine learning models such as logistic regression and decision trees [21].

2.2.2. IDF

Inverse document frequency (IDF) is an estimator that fits on a dataset to produce an IDF model. The IDF model processes feature vectors, typically generated by methods such as *HashingTF* or *CountVectorizer*, and scales each feature accordingly. Essentially, it down-weights features that occur frequently across the dataset [21].

2.2.3. The clustering

Clustering involves dividing data points or enabled data into many clusters or groups based on their similarities and differences. Each data point is similar to the data point of the same cluster and different from the data points in the other groups. In our module, we use clustering to segment users into several clusters based on their demographic information, such as age, gender, and zip code, using the k-means algorithm. Each cluster contains users with similar attributes. Based on the user clusters, we generate user-items-ratings clusters. This clustering helps work with a reduced dataset and improves the efficiency of the recommendation system.

2.2.4. K-means algorithm

The k-means algorithm is one of the most effective and widely used approaches in unsupervised learning [22], it teaches a computer to use unlabeled, unclassified data and enables the algorithm to operate on that data without supervision. We chose the k-means algorithm because it is one of the most popular unsupervised learning algorithms. Due to its speed, it is relatively easy and efficient to apply, even to large datasets. Without any previous data training, this algorithm divides a set of n observations into k clusters. Each observation is assigned to the cluster with the nearest mean, which serves as the cluster's center or centroid. The centroid represents the prototype of the cluster.

Given a set of observations ($x1$, $x2$, ..., xn), where each observation is ad-dimensional real vector, k-means clustering aims to partition the n observations into $k(\leq n)$ sets S = {$S1$, $S2$, ..., Sk} to minimize the within-cluster sum of squares (WCSS) (i.e., variance). Formally, the goal is to find:

$$\underset{\mathrm{S}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{\mathrm{x} \in S_i} |\mathrm{x} - \mu_i|^2 = \underset{\mathrm{S}}{\operatorname{argmin}} \sum_{i=1}^{k} |S_i| Var S_i \quad (1)$$

where μ_i is the mean (also called centroid) of points in S_i.

$$\mu_i = \frac{1}{|S_i|} \sum_{\mathrm{x} \in S_i} \mathrm{x} \quad (2)$$

k-means clustering minimizes within-cluster variances based on squared Euclidean distances between two features x_1 and x_2.

$$\mathrm{d}(x_1, x_2) = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}} (x_{1n}, x_{2n})^2} \quad (3)$$

2.3. The recommendation

After constructing a profile based on the user's attributes, we assign the user to the relevant cluster which contains users with similar demographic data. We use each user's item ratings from the same cluster and apply matrix factorization for recommendations. Matrix factorization is a class of collaborative filtering algorithms commonly employed in recommender systems. This technique involves decomposing the user-item interaction matrix into the product of two lower-dimensional rectangular matrices [23].

In this work, we used the ALS; ALS [24] is a collaborative filtering technique developed in Spark MLlib; we use this technique because it solves the problem of overfitting in sparse data and increases the value of precision [25]. ALS observes the user’s item rating through matrix decomposition. The main idea is to find two low-dimensional matrices, V and U, that approximate the rating matrix R.

$$\mathrm{R}_{(\mathrm{m} \times \mathrm{n})} \approx \mathrm{U}_{(\mathrm{m} \times \mathrm{n})} \mathrm{V}_{(\mathrm{n} \times \mathrm{K})} \quad (4)$$

The objective is to determine the vector for each user (xu) and item (yi) in the feature dimensions, aiming to minimize the following loss function:

$$\operatorname{argmin} \sum_{r_{\mathrm{ui}}} \left(r_{\mathrm{ui}} - x_u^T y_i\right)^2 + \lambda\left(\sum_u \ |x_u|^2 + \sum_i \ |y_i|^2\right) \tag{5}$$

with λ is a regularization parameter used to avoid overfitting, r_{ui} represents the observed rating of item i by user u, xu is the feature vector for user u and y_i is the feature vector for item i. This regularization scheme to avoid overfitting is called weighted-λ regularization.

By fixing one of the matrices U or V, a quadratic form emerges that can be solved directly. This solution ensures a monotonic decrease in the overall cost function. Iterative application of this process to the matrices U and V leads to continuous improvement in the matrix factorization. The matrix R is represented in its sparse format as a tuple (i, j, r) where i represents the row index, j represents the column index, and r is the value of the matrix at position (i, j) [26].

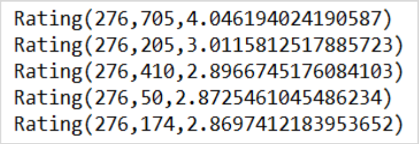
3. RESULTS AND EXPERIMENTS

3.1. Dataset

The MovieLens datasets were gathered by the GroupLens research project. These datasets contain 100,000 ratings from 943 users on 1,682 movies [27]. In this work, we will work with three files: the movies file contains all information about the movies, the *i.data* file contains all the users' ratings on movies, and the user file contains demographic information about users, such as their age, gender, occupation, and zip code.

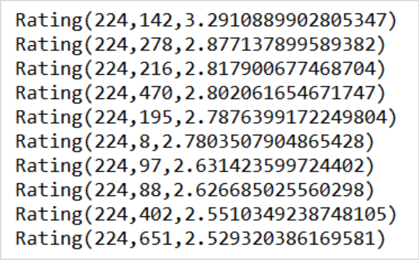
3.2. Results

Our system proposes to specific users, in this example, users with ID=224 and ID=276, the top 10 elements or the top 5 items based on the predicted rating for each item. Figures 2 and 3 present the results of our recommender system. Additionally, Table 1 shows the details of the proposed items and demonstrates that our recommender system suggests a variety of movie genres to the users.



```
Rating(276,705,4.046194024190587)
Rating(276,205,3.0115812517885723)
Rating(276,410,2.8966745176084103)
Rating(276,50,2.8725461045486234)
Rating(276,174,2.8697412183953652)
```

Figure 2. Results of the top 5 recommendation for the user 276 from the system



```
Rating(224,142,3.2910889902805347)
Rating(224,278,2.877137899589382)
Rating(224,216,2.817900677468704)
Rating(224,470,2.802061654671747)
Rating(224,195,2.7876399172249804)
Rating(224,8,2.7803507904865428)
Rating(224,97,2.631423599724402)
Rating(224,88,2.626685025560298)
Rating(224,402,2.5510349238748105)
Rating(224,651,2.529320386169581)
```

Figure 3. Top 10 recommendations for the 224 users from our system

Table 1. recommendations details for user 276

User	Movie ID	Title	Gender	Predicted rating
276	705	Sing in the Rain	Romance-Musical	4.04619
276	205	Patton	War-Drama	3.011581
276	410	Mission impossible	Action-Adventure-Mystery	2.896674
276	50	Star Wars	War-Western-Drama	2.872546
276	174	Raiders of the Lost Ark	Action-Adventure	2.869741

3.3. Experimentations and discussion

In this work, we aim to improve the results of the collaborative filtering system. One way to increase accuracy and recommendation precision is to use large datasets. For this purpose, we use clustering based on users' attributes to segment the dataset into clusters and focus on each user-specific cluster.

We used demographic filtering to cluster users based on age and gender. First, we cluster the users based on gender, with k=2 clusters (one cluster for male and other one for female). Then, we further cluster the data based on age. For instance, with k=4, the users are clustered first by gender (female and male), and then each gender group is further divided into age-based clusters (resulting in 2 male clusters and 2 female clusters). When using k=20, users are first divided into female and male clusters. Then, each gender group is further grouped by age, resulting in 10 male clusters and 10 female clusters. The dataset includes 943 users, of whom 273 are female and 670 are male. When k=2, the recommendations are based solely on gender.

Figures 4 to 8 illustrate the different values of MAE, MSE, and RMSE using the different values of k clusters. From the comparison of the different values of our results in Figures 9 and 10, we illustrate that the best values are achieved when k=20 in Figure 8, and the recommendations are based on age, gender, and collaborative filtering. The evaluation metrics for the male gender are RMSE: 0.662, MSE: 0.439, and MAE: 0.373. For the female gender, the corresponding metrics are RMSE: 0.623, RME: 0.388, and RMA: 0.263. When k=10 in Figure 7, the evaluation metrics for the male gender are RMSE: 0.704, MSE: 0.496, and MAE: 0.467. For the female gender, the metrics are RMSE: 0.646, MSE: 0.417, and MAE: 0.324. In the case of K=8, the RMSE, MSE, and MAE values are 0.726, 0.527, and 0.478 for the male gender. And 0.705, 0.498, and 0.433 for the female gender. When k=6 in Figure 5, the evaluation metrics for the male gender are RMSE: 0.711, MSE: 0.505, and MAE: 0.500. For the female gender, the metrics are RMSE: 0.718, MSE: 0.516, and MAE: 0.417. When k=4, the evaluation metrics for the male gender are RMSE: 0.736, MSE: 0.542, and MAE: 0.533. For the female gender, the metrics are RMSE: 0.715, MSE: 0.512, and MAE: 0.434. When k=2 (with recommendations based on gender only), the evaluation metrics for the female gender are RMSE: 0.749, MSE: 0.561, and MAE: 0.488. For the male gender, the metrics are RMSE: 0.753, MSE: 0.568, and MAE: 0.550. The best recommendation results occur when k=20 and the recommendation approach integrate collaborative filtering with gender and age information.

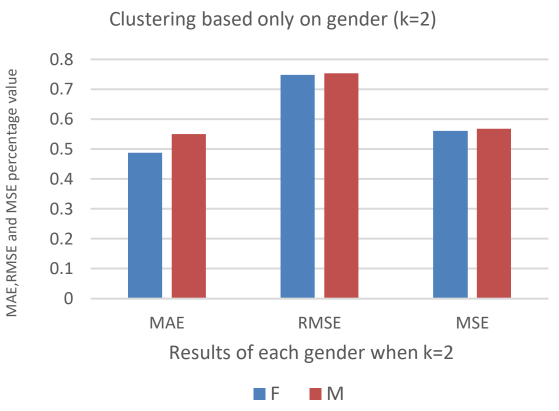


Figure 4. Recommendation based on age and gender k=2

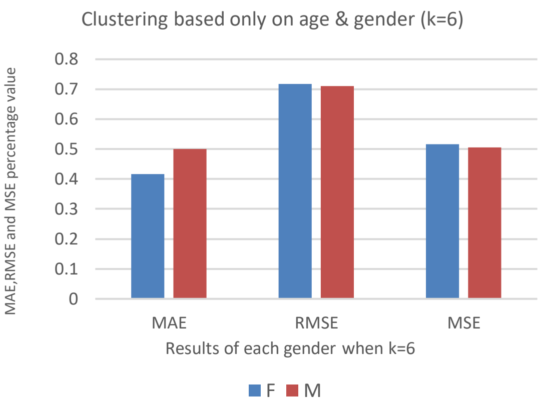


Figure 5. Recommendation based on age and gender k=6

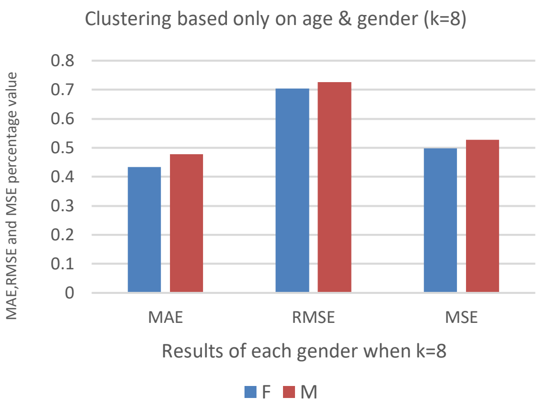


Figure 6. Recommendation based on age and gender k=8

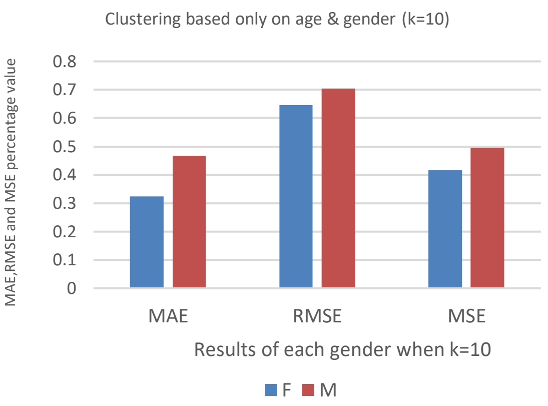


Figure 7. Recommendation based on age and gender k=10

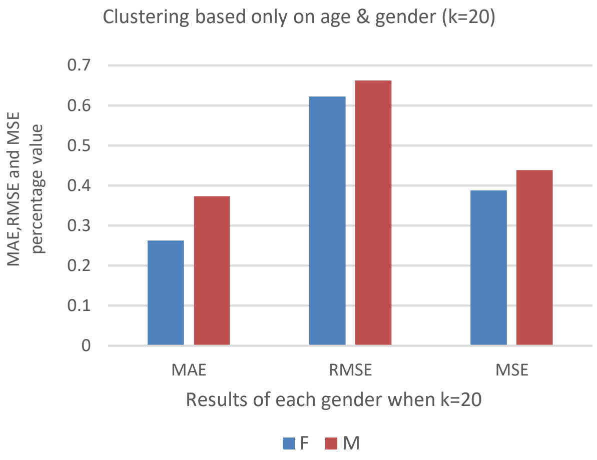


Figure 8. Recommendation based on age and gender k=20

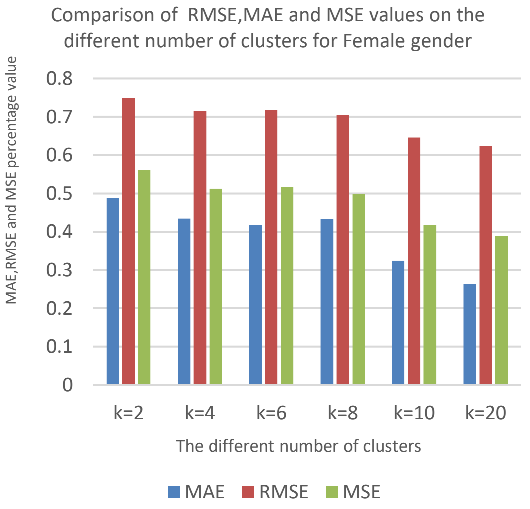


Figure 9. Results of the recommendation on the different number of clusters for female gender

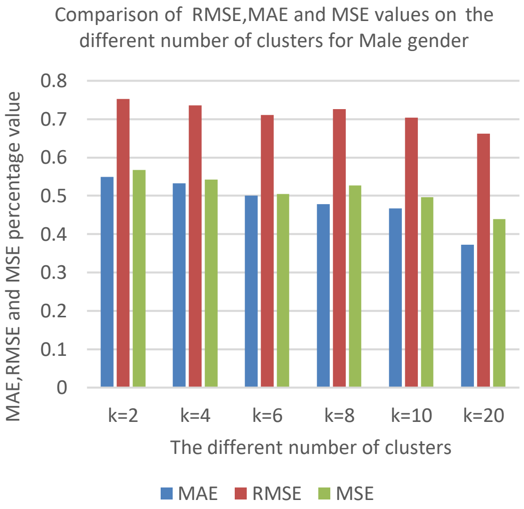


Figure 10. Results of the recommendation on the different number of clusters for male gender

Tables 2 and 3 illustrate the values of RMSE, MAE, MSE, and Precision at k for some chosen users from the dataset. We also evaluated the system using precision at k=10 in Figures 11 and 12. The best results were achieved when k=20, yielding a precision at k=10 value of 0.307 for the female gender in Figure 11. We observe that the value of precision at k increases with the number of clusters (k). The same pattern holds true for the male gender: in Figure 12 the best results are achieved when k=20, with a precision at k=10 value of 0.100.

Table 2. RMSE, MSE, and MAE for different chosen users from dataset

Users	RMSE	MSE	MAE
1	0.660	0.436	0.404
244	0.639	0.409	0.391
44	0.657	0.432	0.402
58	0.656	0.431	0.401
181	0.662	0.438	0.400
201	0.671	0.451	0.404
268	0.659	0.434	0.399
160	0.649	0.422	0.394

Table 3. Precision@k for different chosen users from the dataset

Users	Precision@k=10
1	0.081
244	0.095
44	0.088
58	0.089
181	0.094
201	0.091
268	0.100
160	0.090

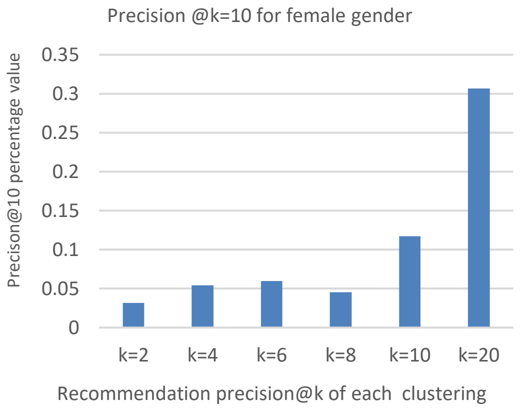


Figure 11. Precision@k When the gender is female

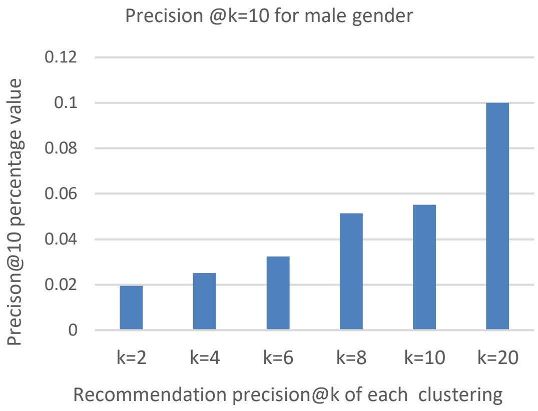


Figure 12. Precision@k when the gender is male

In Tables 4 and 5, and Figures 13 and 14, we compare our system, which integrates collaborative filtering with demographic information (gender, age, and zip code), against other systems. These include a system that uses collaborative filtering with demographic information based solely on gender, a system that uses collaborative filtering with other demographic approaches, and a system based only on collaborative filtering. We evaluate the results using RMSE, MSE, MAE, and precision. The best outcomes are achieved when using the user's age, gender, and zip code, yielding values of 0.587 for RMSE, 0.345 for MSE, and 0.247 for MAE.

Table 4. The comparison of results of different approaches

The different approaches	RMSE	MSE	MAE
Recommendation based on CF only	0.767	0.589	0.573
CF & user's Age	0.715	0.512	0.382
CF & user's Gender	0.749	0.561	0.488
CF & user's Gender & Age	0.587	0.345	0.247
CF & user's Gender & Zipcode	0.643	0.414	0.31
CF & user's Age &Sexe & Zip20K	0.659	0.434	0.319

Table 5. The comparison of Precision@k results of different approaches

The different approaches	Precision@k=10
Recommendation based on CF only	0.013
CF & user's Age	0.06
CF & user's Gender	0.031
CF & user's Gender & Age	0.41
CF&Age & zip20k	0.39
CF&Sexe Zip code	0.144
CF&AgeSexeZip20K	0.17

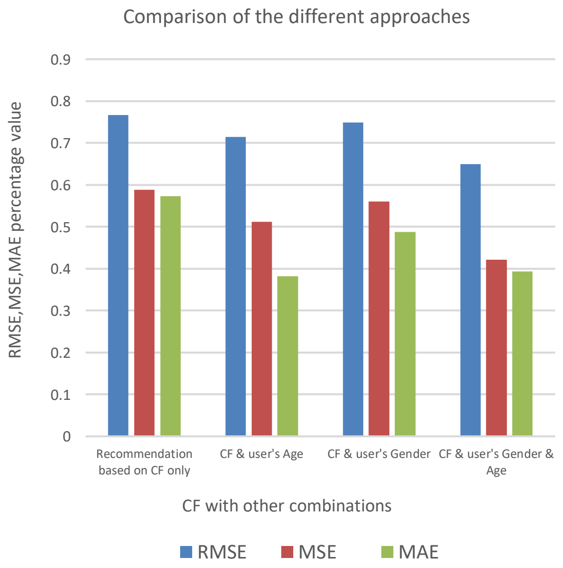


Figure 13. Comparison of MSE, RMSE and MAE of our system with others

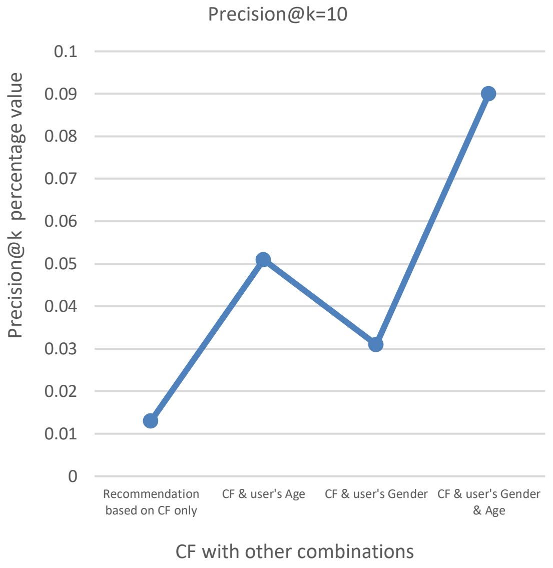


Figure 14. Comparison of the precision@k=10 of our system with other systems

4. CONCLUSION

In this article, we propose a recommender system based on the collaborative filtering and demographic approach using age, gender, and zip code information; our system consists of 2 main modules: the demographic filtering module and the collaborative filtering module. In the first module, we cluster users using the k-means algorithm based on their demographic information; we then construct the users-items matrix of users from the same cluster and then apply collaborative filtering to recommend items to the users using the factorization matrix. We used the MovieLens dataset 100K for experiments, and the results demonstrate that RMSE, MAE, MSE, and Precision@k values improve when the number of used clusters increases. The best results are given when k=20.The main goal of this work is to improve the results of the collaborative filtering recommender system, recommend diverse items to the users, and enhance the precision of recommendation systems based on collaborative filtering. Our experiments in comparison with other works, show that the proposed approach gives better results than collaborative filtering.

ACKNOWLEDGEMENTS

The work is done through the support of the CNRST in part merit scholarship of research.

REFERENCES

[1] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, pp. 393–408, 1999.

[2] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, Boston, MA: Springer US, 2011, pp. 1–35.

[3] D. Das, L. Sahoo, and S. Datta, "A survey on recommendation system," *International Journal of Computer Applications*, vol. 160, no. 7, pp. 6–10, Feb. 2017, doi: 10.5120/ijca2017913081.

[4] D. Mladenic, "Text-learning and related intelligent agents: a survey," *IEEE Intelligent Systems*, vol. 14, no. 4, pp. 44–54, Jul. 1999, doi: 10.1109/5254.784084.

[5] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, Jun. 2005, doi: 10.1109/tkde.2005.99.

[6] M. Aamir and M. Bhusry, "Recommendation system: state of the art approach," *International Journal of Computer Applications*, vol. 120, no. 12, pp. 25–32, Jun. 2015, doi: 10.5120/21281-4200.

[7] S. Nabil, J. Elbouhdidi, and M. Yassin Chkouri, "Recommendation system based on data analysis-application on tweets sentiment analysis," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, Oct. 2018, pp. 155–160, doi: 10.1109/CIST.2018.8596418.

[8] J. Kadurhalli Sangappa and S. Chikkanaravangala Paramashivaia, "Multi-domain aspect-oriented sentiment analysis for movie recommendations using feature extraction," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 2, pp. 1216–1223, Feb. 2024, doi: 10.11591/ijeecs.v33.i2.pp1216-1223.

[9] A. Ziani *et al.*, "Recommender system through sentiment analysis," in *2nd international conference on automatic control, telecommunications and signals*, 2017, pp. 1–7.

[10] C. N. Dang, M. N. Moreno-García, and F. De la Prieta, "An approach to integrating sentiment analysis into recommender systems," *Sensors*, vol. 21, no. 16, Aug. 2021, doi: 10.3390/s21165666.

[11] I. Karabila, N. Darraz, A. El-Ansari, N. Alami, and M. El Mallahi, "Enhancing collaborative filtering-based recommender system using sentiment analysis," *Future Internet*, vol. 15, no. 7, Jul. 2023, doi: 10.3390/fi15070235.

[12] Y. Wang, S. C.-F. Chan, and G. Ngai, "Applicability of demographic recommender system to tourist attractions: a case study on trip advisor," in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Dec. 2012, pp. 97–101, doi: 10.1109/WI-IAT.2012.133.

[13] M. Sridevi and R. R. Rao, "DECORS: a simple and efficient demographic collaborative recommender system for movie recommendation," *Advances in Computational Sciences and Technology*, vol. 10, no. 7, pp. 1969–1979, 2017.

[14] A. Yassine, L. Mohamed, and M. Al Achhab, "Intelligent recommender system based on unsupervised machine learning and demographic attributes," *Simulation Modelling Practice and Theory*, vol. 107, Feb. 2021, doi: 10.1016/j.simpat.2020.102198.

[15] M. F. Aljunid and D. H. Manjaiah, "Movie recommender system based on collaborative filtering using Apache Spark," *Advances in Intelligent Systems and Computing (AISC)*, vol. 839, 2019, pp. 283–295.

[16] J. F. G. da Silva, N. N. de Moura Junior, and L. P. Caloba, "Effects of data sparsity on recommender systems based on collaborative filtering," in *2018 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8489095.

[17] J. Chen, C. Zhao, Uliji, and L. Chen, "Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering," *Complex & Intelligent Systems*, vol. 6, no. 1, pp. 147–156, Apr. 2020, doi: 10.1007/s40747-019-00123-5.

[18] H. Zarzour, Z. Al-Sharif, M. Al-Ayyoub, and Y. Jararweh, "A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, Apr. 2018, pp. 102–106, doi: 10.1109/IACS.2018.8355449.

[19] N. Nassar, A. Jafar, and Y. Rahhal, "A novel deep multi-criteria collaborative filtering model for recommendation system," *Knowledge-Based Systems*, vol. 187, Jan. 2020, doi: 10.1016/j.knosys.2019.06.019.

[20] R. M. Sallam, M. Hussein, and H. M. Mousa, "Improving collaborative filtering using lexicon-based sentiment analysis," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 2, pp. 1744–1753, Apr. 2022, doi: 10.11591/ijece.v12i2.pp1744-1753.

[21] Apache Spark, "Extracting, transforming and selecting features," Apache Spark, https://spark.apache.org/docs/latest/ml-features.html (accessed Jun. 02, 2023).

[22] W. N. I. Al-Obaydy, H. A. Hashim, Y. A. Najm, and A. A. Jalal, "Document classification using term frequency-inverse

document frequency and K-means clustering," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 3, pp. 1517–1524, Sep. 2022, doi: 10.11591/ijeecs.v27.i3.pp1517-1524.

[23] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009, doi: 10.1109/MC.2009.263.

[24] B. Mitroi and F. Frasincar, "An elastic net regularized matrix factorization technique for recommender systems," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, Mar. 2020, pp. 2184–2192, doi: 10.1145/3341105.3373847.

[25] S. Gosh, N. Nahar, M. A. Wahab, M. Biswas, M. S. Hossain, and K. Andersson, "Recommendation system for E-commerce using alternating least squares (ALS) on Apache spark," Advances in Intelligent Systems and Computing (AISC), vol. 1324, 2021, pp. 880–893.

[26] Apache Flink, "Alternating least squares," *Apache Flink version 1.4*. https://nightlies.apache.org/flink/flink-docs-release-1.4/dev/libs/ml/als.html#:~:text=Examples,Description,and is called latent factors (accessed Jun. 03, 2023).

[27] "MovieLens 100K dataset," Kaggle, https://www.kaggle.com/datasets/prajitdatta/movielens-100k-dataset?resource=download (accessed Dec. 23, 2022).

BIOGRAPHIES OF AUTHORS

Sana Nabil is a Ph.D. student at the research Laboratory of the Information System and Software Engineering (SIGL), University Abdelmalek Essaadi, National School of Applied Sciences-Tetouan-Morocco. His research interests include big data, machine learning, sentiment analysis and recommender systems. He can be contacted at email: sana.nabil@etu.uae.ac.ma.

Mohamed Yassin Chkouri is a professor of computer science at the SIGL Laboratory, University Abdelmalek Essaadi, National School of Applied Sciences-Tetouan-Morocco, he is Head of Computer Engineering Department, SIGL Research Laboratory Director and Academic coordinator Erasmus + e-VAL project. He can be contacted at email: mychkouri@uae.ac.ma.

Jaber El Bouhdidi is an HDR professor of computer science at the SIGL Laboratory, University Abdelmalek Essaadi, National School of Applied Sciences-Tetouan-Morocco. His research interests include web semantic, multi-agents' systems, e-learning adaptive systems and big data. He has several papers in international conferences and journals. He can be contacted at email: jaber.elbouhdidi@uae.ac.ma.