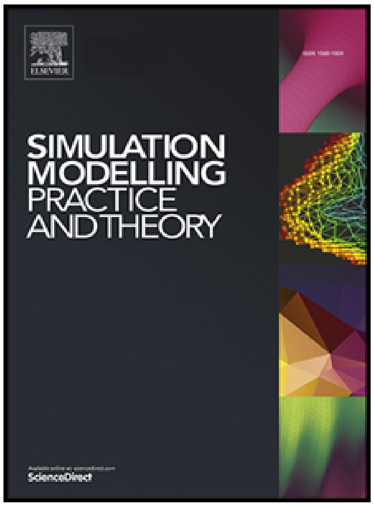
Journal Pre-proof



Intelligent recommender system based on unsupervised machine learning and demographic attributes

AFOUDI Yassine, LAZAAR Mohamed

PII: S1569-190X(20)30137-4
DOI: https://doi.org/10.1016/j.simpat.2020.102198
Reference: SIMPAT 102198

To appear in: *Simulation Modelling Practice and Theory*

Received date: 12 May 2020
Revised date: 17 August 2020
Accepted date: 18 September 2020

Please cite this article as: AFOUDI Yassine, LAZAAR Mohamed, Intelligent recommender system based on unsupervised machine learning and demographic attributes, *Simulation Modelling Practice and Theory* (2020), doi: https://doi.org/10.1016/j.simpat.2020.102198

Intelligent recommender system based on unsupervised machine learning and demographic attributes

AFOUDI Yassine, LAZAAR Mohamed

ENSIAS, Mohammed V University in Rabat, Morocco

Abstract

Recommendation systems aim to predict users interests and recommend items most likely to interest them. In this paper, we propose a new intelligent recommender system that combines collaborative filtering (CF) with the popular unsupervised machine learning algorithm K-means clustering. Also, we use certain user demographic attributes such as the gender and age to create segmented user profiles, when items (movies) are clustered by genre attributes using K-means and users are classified based on the preference of items and the genres they prefer to watch. To recommend items to an active user, Collaborative Filtering approach then is applied to the cluster where the user belongs.
Following the experimentation for well known movies, we show that the proposed system satisfies the predictability of the CF algorithm in GroupLens. In addition, our proposed system improves the performance and time response speed of the traditional collaborative Filtering technique and the Content-Based technique too.

Keywords: Collaborative filtering; K-means clustering; Movies recommendation; Recommender system

1. Introduction

Today, the internet allows people to access abundant online resources. Amazon, for example, has a huge collection of products. Although the amount of information available is increasing day by day, this leaves people or specially costumers in new trouble, they find it too difficult to choose what they really want to see or buy. This is where the recommendation system comes in.
Recommendation systems help users find and choose items (books, movies, restaurants, music... etc.) that may interest them, from the large number of

Email addresses: `yassine.afoudi@um5.ac.ma` (AFOUDI Yassine), `m.lazaar@um5s.net.ma` (LAZAAR Mohamed)

choices available on the web or other electronic information sources. Thats why we found many companies like Amazon, Netflix and Spotify use recommendation systems which, of course, allow them to generate significant engagement and revenue. The reason these companies and others are seeing increased revenue is that they are bringing real value to their customers, by suggesting items and making their lives easier, also by knowing what a user wants the threat of losing a customer to a competitor decreases.

Many approaches are used in recommender systems and we can make a distinction between them by analyzing the type of data used to generate recommendations. In general, there are three well-known techniques used in this field, Collaborative Filtering (CF), Content-Based Filtering (CBF) and Hybrid models. Collaborative filtering models are based on the idea that a person likes articles similar to other articles he likes in the past, and the articles that are liked by other people with similar taste, here we use data that the user provides, either explicitly (rating) or implicitly (clicking on a link). As the user provides more inputs or take actions on the suggestions, the system becomes more and more powerful. Content-Based models are based on the idea that each product has certain keywords or description, understands what the user likes based on items he likes before, look up those keywords or features in the database and recommend different products with the same attributes. For Hybrid recommender systems combine two or more recommendation methods to gain better performance with fewer drawbacks. Most common is collaborative filtering combined with some other techniques. In this article, we propose a hybrid recommendation system, which combines collaborative filtering with the K-means unsupervised machine learning algorithm to improve the performance of the traditional CF technique. We first cluster the items according to some of their features by k-means approach and after that we analyze the demographic characteristics of users to segment the users, then classify them according to the upper class of the items that the user likes. In order to recommend items for a user, we determine a cluster in advance where the features are similar to the user, then a Collaborative Filtering applies to the users who belong to the cluster in order to predict users preference for items.

The rest of the document is organized as follows. The following section briefly describes the recommendation systems as well as Collaborative Filtering the best well-known approach and some related work in the context of movie recommendation. Section 2 presents the proposed approach. Sections 3 evaluate our system with other techniques in the area of recommendation and discuss the results. Finally, Section4 summarizes the conclusion.

2. BACKGROUND

2.1. Recommendation Tasks

Recommendation systems have many algorithms that aim to provide the most relevant and accurate product to the user by filtering useful information from a huge database pool. In general, recommendation engine based on three types of approaches collaborative filtering, content based and hybrid recommender system, but in this article we are interested in Collaborative Filtering.

2.1.1. Collaborative filtering recommender system

Collaborative filtering is currently one of the most frequently used approaches and generally provides better results than content-based recommendations, that's why this method it can be found in the recommendation systems of Spotify, Netflix and YouTube.There is a lot of research on collaborative filtering, these types of systems use user interactions to filter items of interest and the most popular techniques are categorized into two basic approaches: model-based and memory-based.

2.1.2. Memory-Based Collaborative Filtering

Memory-based collaborative filtering approaches can be divided into two main section's user-item filtering and item-item filtering. User-item filtering takes a particular user we want to give him recommendation items, searches for users similar to this user in another way users with the same taste based on the similarity of the ratings and recommends items that those similar users liked. On the other hand, using item-item filtering, we take an item or product and we search for the users who liked this item, then we find other items that these users or similar users also liked. It takes the item as input and generates other items as recommendations.
The basic idea of the two sections is to create a rating matrix, in a simple way, we build a matrix whose rows represents the users and the columns represent the items and rating data as values, then we use a similarity approach to find similar users or items, the common distance metric used in the majority of work is cosine similarity. We can predict user-xs rating for the movie-y by taking a weighted sum of the movie-y ratings from all other users where weighting is a similar number between each user and user-x, thats why memory-based approach is easy to use but its performance decreases when we have sparse data and it takes a lot of time.

2.1.3. Model-Based Collaborative Filtering

In this approach, CF models are developed using machine learning algorithms and data mining techniques to predict user rating for items that do not have a rating. The goal is to train models to be able to make predictions in other words, we extract some information from the dataset, and use that as a "model" to make recommendations without needing to use the complete dataset every time. This approach potentially offers the benefits of both speed and scalability. In this paper, we will use a Model-Based technique based on matrix factorization named Singular Value decomposition (SVD).

2.2. Related work

Many researches have been done in the field of CF recommendation system and movies personalization recommendation. Several personalized movie systems have been built in recent years to help users deal with the large number of existing movies.
Reddy S. et al. [17] propose a method that uses content-based filtering using genre correlation. In Rohan Nayak et al.[14] hybrid system the users will be asked to provide feedback on some movies and movie genre, and based on the feedback provided, the user will be segregated, and a set of recommendations will be provided. Katarya Rahul [11] proposes a hybrid recommender system which utilized k-means clustering algorithm with bio-inspired artificial bee colony (ABC) optimization technique and applied to the Movielens dataset. Ponnam, L. T. et al.[15] propose an item based collaborative filtering technique that examine the user item rating matrix and identify the relationships among various items to compute the recommendations for the user. In Bagher Rahimpour Cami et al.[3] paper, they propose a content-based movie recommender system that captures the temporal user preferences in user modeling and predicts the preferred movies and with the evolution of deep learning techniques. We found also Rex Ying et al.[22] create an efficient GCN (Graph Convolutional Network) algorithm for data, which combines efficient random walks and graph convolutions to generate embeddings incorporations. BogdanWalek et al. [21] propose a monolithic hybrid recommender system which combines a collaborative filtering system based on the SVD algorithm with content-based system, and a fuzzy expert system. The expert system evaluates the importance of the movies based on several parameters such as the average movie rating, number of ratings, etc. Hong-Quan Do et al. [4] propose a hybrid recommender system focus on the weighted hybridization and rather than using fixed weighted for the combination, they aim to offer a simple method to dynamic weight the combination of CF and CBF. Arisara Pornwattanavichai et al. [16] propose a new method of recommending Tweets based on hybrid Recommender system with LDA the unsupervised topic modeling and generalized matrix factorization the supervised learning-based neural network.
As mentioned above, collaborative filtering is the well-known technique used to give powerful recommendations using the ratings data. Based on this and our selected dataset, we will continue the research to improve the achievement of results by this technique by presenting k-means clustering algorithm in the movie recommendation system, which is based on collaborative filtering recommendation method and demographic attributes segmentation.

3. PROPOSED WORK ARCHITECTURE

This paper proposes K-means clustering algorithm based on collaborative filtering approach in the movies recommendation system, Figure 1 represent the proposed architecture, it mainly consists of the following four functional modules: movies feature extrac-

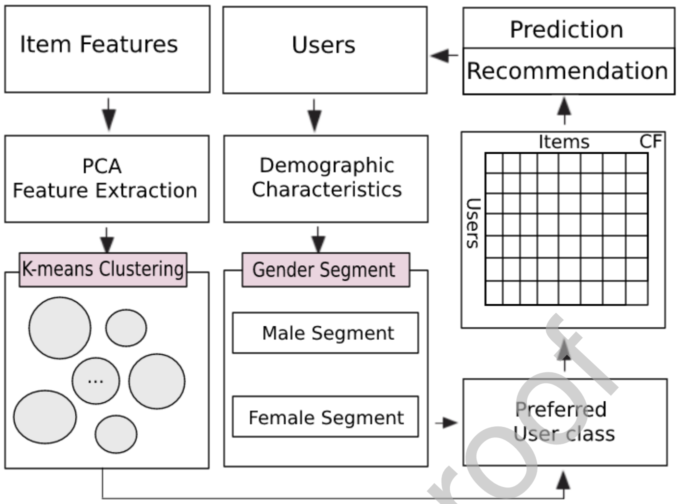


Figure 1: Adopted architecture

tion module, k-means clustering module, user profile module and finally recommendation module.

3.1. Movies feature extraction module

We test our system with a movielens dataset, each movie represented with features such as the title, the movie identifier, and 19 genres attribute (if a movie has a specific genre we found 1 as a value otherwise 0), for this we plan to use a feature selection or extraction technique to reduce the number of attributes, in general during our previous work[2], we found that there was an impact on the use of feature selection techniques on the results of the recommendations. Principal Component Analysis (PCA) is our choice to reduce the number of features, this technique is a dimensionality reduction technique used when we have a slow algorithm because the input dimension is too high and we want to speed it up.

PCA is a dimensionality reduction that identifies important relationships in our data, transforms the existing data based on these relationships, and then quantifies the importance of these relationships so we can keep the most important relationships and delete the others.

Based on the definition above, we use PCA to reduce the number of features in our movies database from 19 genres attributes to 10 attributes and by reading the explained variance, we found that the 10 components contain 86.92% of the item's information and that's good for us, but there is a question of the quality of recommendation using all features and using only 10 arises. To answer this important question, we test the two datasets with a combined CF approach with simple k-means, and we evaluate the result using the RMSE technique (we will explain this technique in the evaluation section), Figure 2 shows us that using 10 features obtained by PCA gives us much better performance instead of

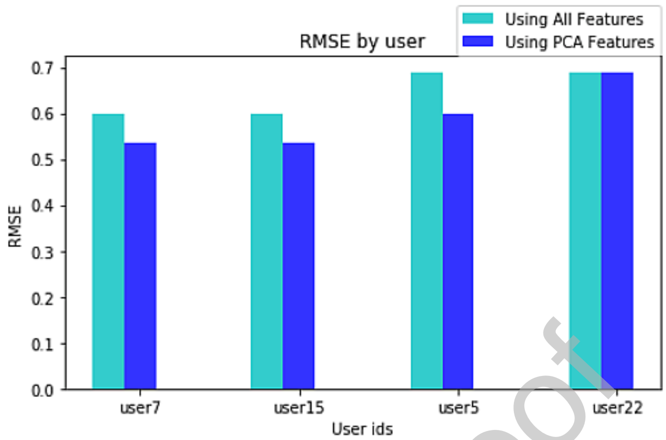


Figure 2: The Collaborative RMSE results using all features and PCA features on 4 random users

using 19 genre features because the Mean error has been reduced. To make the decision on which datasets we will use, we repeat our test several times, for each one, we found that the dataset contains PCA features gives us much better performance, that's why we will continue our work only with our 10 movie PCA features to minimize the memory occupied by our system.

3.2. K-means clustering module

This module is used to cluster movies into a specific number of classes, this part describes the standard k-means algorithm. K-means clustering is one of the simplest and popular unsupervised machine learning algorithms, as a rule, unsupervised algorithms make inferences from datasets using only input vectors without referring to known or labeled results.
Simply, K-means looks for a fixed number (k) of clusters in a dataset, in other words, the K-means algorithm identifies k number of centroids and calculate the distance between each object and each cluster center, then assign it to the nearest cluster, update the averages of all clusters, repeat this process until the criterion function converged.
The K-Means algorithm needs a way to compare the degree of similarity between the different features. Thus, two data which are similar, will have a reduced dissimilarity distance, while two different objects will have a greater separation distance, in this work, we use the k-means clustering algorithm based on the Euclidean similarity approach.
The Euclidean distance is the geometric distance, for example,if we have a matrix X with i quantitative variables. The Euclidean distance d between two features x1 and x2 is calculated as follows:

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

After all the information given above, we have arrived at the stage of finding the value of k. The most common method for choosing the number of clusters is to launch K-Means with different values of K and calculate the variance of the different clusters. The variance is the sum of the distances between each centroid of a cluster and the different items included in the same cluster. Thus, we seek to find a certain number of clusters K so that the selected clusters minimize the distance between their centroids and the items in the same cluster. Generally, by plotting the different numbers of clusters K as a function of the variance, the point of the elbow is that of the number of clusters whose variance no longer decreases significantly (Elbow approach).
In this work, we give our 10 components from the previous module to the K-means approach and we obtain as output our dataset of movies classified into a specific number of clusters.

3.3. User profile module

The main mission of this module is to build for each user a profile based on a set of parameters for recommendation purposes, in our work we take the rating data from each user, and we only use movies with a rating greater than or equal to 2.5. After getting the highest rated movies in another way for an active user, we assign for each movie in the list its cluster class, after that we calculate the sum of each movie class in the list and we use the maximum class result as a representative class of the most liked user movie group.
For example, we have a user X who likes movie classes [1,1,1,2,3,3,4,1,1], here, we can assign that our user liked movies classified in class 1 more than the other classes.

3.4. Recommender system module

This paper proposes a system based on Collaborative Filtering technique and specially the Model-Based approach, one of the most successful Model-Based CF techniques is Matrix Factorization. Singular value decomposition (SVD)[19] is the well-known matrix factorization method and it is our selected method, the general equation of SVD approach can be expressed as bellow.

$$X = U \times S \times V^T \tag{2}$$

The SVD technique was introduced into the recommendation system domain by Simon Funk during the Netflix Prize challenge [23].
Based on Linear Algebra, any real matrix R can be decomposed into three matrices U, W, and V, using this rule SVD result gives three matrices as output U, S and V^T , matrix U represents user vectors and Matrix V^T represents movie vectors in our case, when elements on the diagonal in S are known as singular values of X.
In this step and after classifying our movie datasets into clusters, we take a user and build a preferred movie class (user profile), then to predict the user ratings of all the movies using this method, we simply applied the SVD approach with only movies in the preferred user class and we build a matrix that contains rows representing the users with the same gender as the active user and columns representing the selected movies and for the values we use all available ratings, after that we take the dot product of U, S and V^T as indicated in Eq.3. Next, we select the highest rated movies and we give the recommendation after ignoring the previously viewed movies.

$$\hat{X} \approx U \cdot S \cdot V^T \tag{3}$$

4. RESULT AND DISCUSSION

4.1. The dataset

We chose the movielens 100k dataset [9] to evaluate our experiments because it is widely used and publicly available. MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. We use this dataset for a study where the goal is to generate recommendation of movies to users. This data set contains 100,000 ratings (1 to 5 scale) from 943 users on 1682 movies and each user have rated at least 20 movies, and there is some demographic information for the users like age, gender, occupation and zip code.

4.2. Implementation

4.2.1. Experimental steps

First step is to import the dataset in our project, and then we split the whole users interaction dataset into 75% as the training set and 25% as the test set.
The second step is to use PCA feature extraction to minimize the genre attributes in the movie dataset from 19 genres to 10 components, after that we use K-means clustering to group the whole dataset into a specific number of clusters, by plotting the different 10 numbers of clusters as a function of the variance, based on the approach of Elbow explained above and by observing the Figure 3 we choose the k=6 the number of our k-means clustering.
After the movies are grouped into 6 clusters,, we assign for each movie its cluster class number, then we create a user profile for all users according to the preferred viewed class, then we take an active user and we search for all movies in his favorite class and all users with his gender, and finally we apply the SVD collaborative filtering approach to make recommendations and give the result to the user after deleting all the movies previously viewed.
We will test and evaluate our recommendations by searching whether the recommended movies to a user according to the training set, are in the item list the user have seen and rated in the test set.

4.2.2. Experimental platform

All our experiments were implemented using Python and compiled using Jupyter notebook. We ran all our experiments on a MacBook Pro with Intel core i7 processor having a speed of 2.7 GHz and 8GB of RAM.

4.2.3. Experiment results

After building our system, we tested it for all users in the dataset and evaluate the results using time recommendation speed and other offline techniques with the state of the art approaches in recommendation filed. The result of the experiment is presented in Table 1 for three random users.

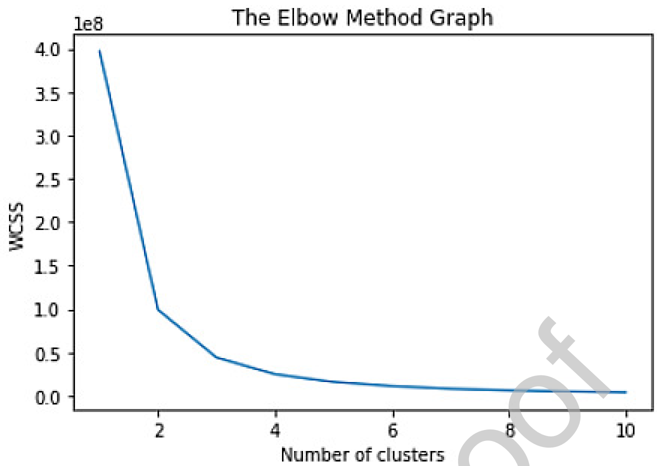


Figure 3: The variance of 10 numbers of clusters on our dataset (Elbow approach)

4.2.4. Evaluation

In the area of recommendation systems, the user wishes to receive the N best recommended items. In this way, the user will view some recommended movies, classified from best to worst. In fact, in some cases, the user does not care much about the exact order of the list, a few good recommendations are enough.
As a result, many evaluation techniques are involved in this area, classified as online and offline techniques. For those online, we need real users to give their opinion and feedback on the recommendation given, in this work we will use the offline techniques based on the movies cited for each user in the test set.
In this article, from many offline approaches,we choose to use the RMSE and Precision-Recall at k techniques. Root Mean Squared Error (RMSE) is a popular technique used to evaluate a recommender system accuracy based on ratings data. In other way after using the train set to build our model we predict movies rating for a given user in the test set and here we compare the predicted ratings with the true values, then we just compute the square root average of the errors from the whole test set to find the RMSE value as shown in formula below, when P is the predicted rating and R the true rating.

$$RMSE = \sqrt{\frac{\sum_{ratings}(P-R)^2}{\#ratings}} \quad (4)$$

The Precision-Recall at k method also is an offline way to evaluate a recommender system where k is a definable integer defined by the user to match the goal of the Top-N recommendations. For this method, we test every model by giving a list of movie recommendations for all users and we evaluate results by calculate and compare the mean of F-Measure result. Recall and Precision at k are then determined as Equations 5 and 6.

$$Recall@k = R_i/Nr \quad (5)$$

Table 1: The Top 3 movies recommended with our system on some random users.

User ids	User Preferred movie class	User Gender	Top three recommended movies
5	class 3	Female	- The Sting - Young Frankenstein - A Fish Called Wanda
17	class 2	Male	Fargo Twelve Monkeys Lone Star
9	class 2	Male	- Twelve Monkeys - Fargo - The Silence of the Lambs

$$Precision@k = R_i/Tr \tag{6}$$

Where Ri is the number of recommended movies at k that are relevant, Tr is the total of relevant items and Nr is the number of recommended items at @k. After calculating The Recall and Precision at k we should normalize the result, then we introduce F-Measure, which is

$$F - measure@k = (2 \times P@k \times R@k)/(P@k + R@k) \tag{7}$$

After building our system, we want to know and evaluate which model gives us powerful recommendation and best accuracy of our Movielens 100k database. As explained above we use precision-recall method, Figure 4 shows us a plot of the mean F-measure at 5 and 10 the number of recommended movies for all users, the Figure 5 shows us the F-measure at k=5 scatter plot of 10 first users and Figure 6 shows us the Precision curve at k=10 for first 20 users, when the Figure 7 shows us the time in seconds taken by all the models to recommend 20 movies to the same three users.

Finally, it should be mentioned that recommendation systems take a bit space and run-time to give suggestions due to the large number of features and parameters. By observing and reading the results, we can assume that the use of K-means clustering the popular unsupervised machine learning technique with Collaborative Filtering associated with demographics attributes gives us much better performance and accuracy, also a fast time response on movies recommendation than the models based on the traditional Collaborative Filtering or Content-Based approaches but our approach still has a limited point is the attributes associated with the items. Obviously, if the item does not have descriptive attributes, it cannot be used in the K-means clustering step to find out which cluster contains this item.

To show how well our approach is performing, we are testing our model using the IMDB dataset which contains 671 users with at least 20 interactions and 42,262 unique movies and 10,0004 user / item interactions, we split the rating data by 75% for the train set and 25% as a test set, reading the result given by the precision recall approach as shown in

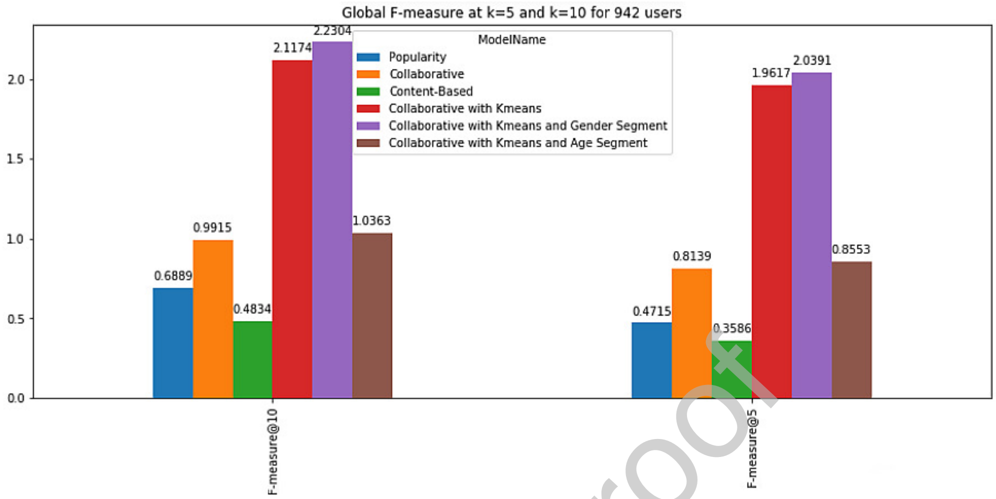


Figure 4: The mean F-measure plot of six models at K=5 and K=10 using Movielens dataset

figure 8, we have found that our approach always gives good and better results than the state of art approaches existing in the field of recommendation. To give more details on the generosity of our approach, we can say that, if we have an item dataset with many genre attributes and a user dataset with demographic attributes, our approach can work well and it will be a good choice to offer customers what they want.
The main Managerial Insights of our approach aim to give a powerful recommendation in a short time. It also helps to eliminate the limit of the cold start problem and give suggestions to a new user even if we don't have much information about user transaction

5. CONCLUSION AND FUTURE WORK

This article proposes an algorithm which uses the unsupervised machine learning algorithm K-means clustering with the collaborative filtering the well-known approach in the field of recommendation and we take into account certain demographic attributes.
The experimental results show the RMSE was reduced when we use PCA feature extraction instead of using all 19 movie features, after that we take those features for building our system based on clustering movies into 6 clusters, then we take the most viewed movie class for an active user and all users with the same gender and we apply the SVD collaborative filtering to give powerful recommendations. By using Precision-Recall the offline approach for evaluating a recommender system we found that the proposed algorithm can efficiently improve the accuracy and the performance of movies recommendation. So the proposed method is feasible.

In another study, we will continue to explore and improve other techniques used in recommendation systems and try to use advanced machine learning algorithms and deep learning models.

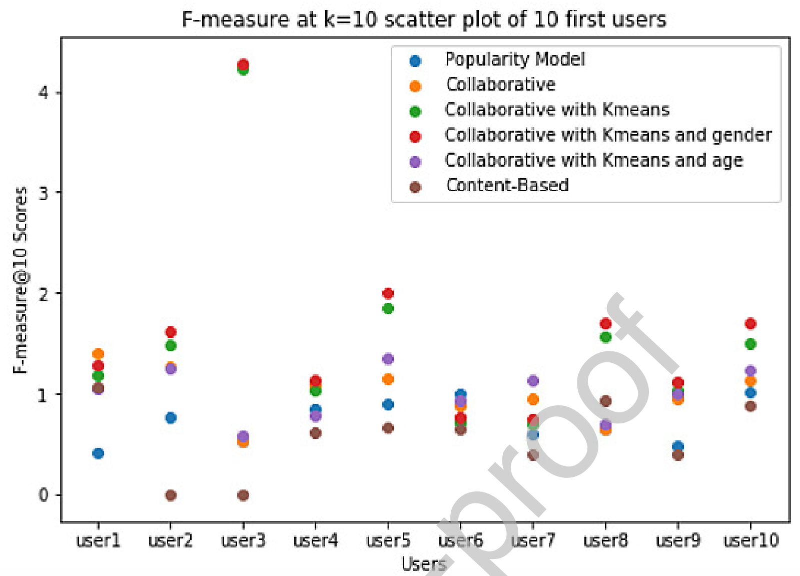


Figure 5: The F-measure scatter plot of 10 users

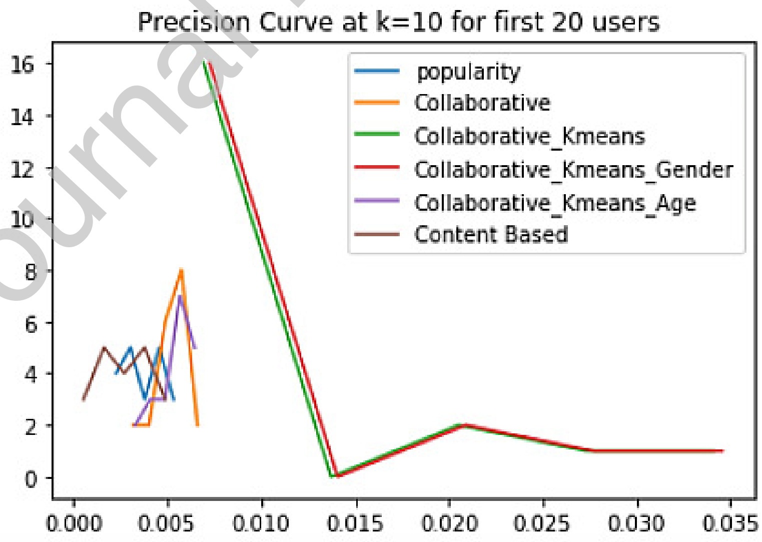


Figure 6: The Precision curve at k=10 for first 20 users

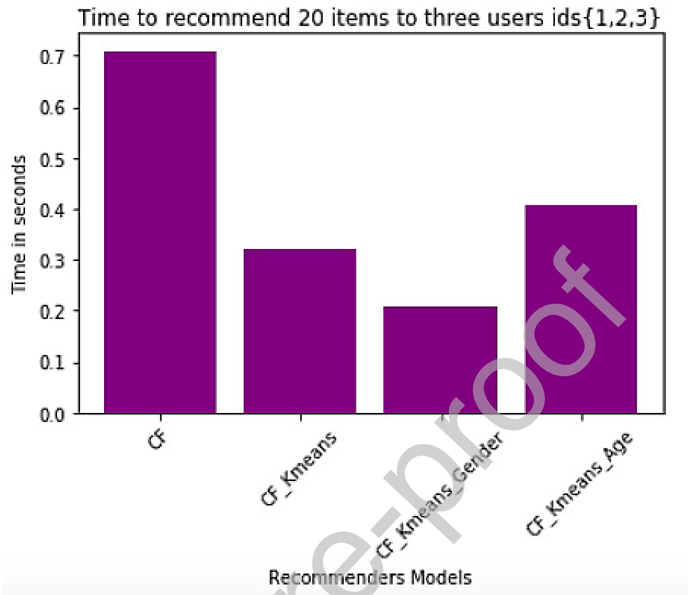


Figure 7: The time in seconds taken by all the models to recommend movies to three users

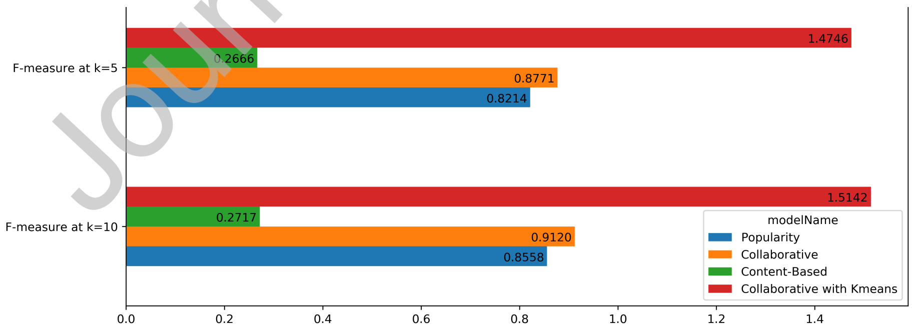


Figure 8: Global F-measure plot at K=5 and K=10 for 671 users using IMDB dataset

ACKNOWLEDGMENT

The authors would like to thank the Smart System Lab our research laboratory, and Al Borchers for cleaning up this data.

References

[1] Afoudi, Y., Lazaar, M., Al Achhab, M., 2019. Collaborative Filtering Recommender System. In: Advanced Intelligent Systems for Sustainable Development (AI2SD2018). Advances in Intelligent Systems and Computing. Springer International Publishing, Cham, pp. 332–345.

[2] Afoudi, Y., Lazaar, M., Al Achhab, M., 2019. Impact of Feature selection on content-based recommendation system. In: 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS). pp. 1–6.

[3] Cami, B. R., Hassanpour, H., Mashayekhi, H., 2017. A content-based movie recommender system based on temporal user preferences. In: 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS). IEEE, Shahrood, pp. 121–125.

[4] Do, H.-Q., Le, T.-H., Yoon, B., ???? Dynamic weighted hybrid recommender systems. In: 2020 22nd International Conference on Advanced Communication Technology (ICACT). pp. 644–650.

[5] Du, Z., Zhang, T., Chen, Y., Ai, L., Wang, X., 2011. A content and user-oblivious video-recommendation algorithm. Simulation Modelling Practice and Theory 19 (9), 1895–1912.

[6] Duwairi, R., Abu-Rahmeh, M., 2015. A novel approach for initializing the spherical K-means clustering algorithm. Simulation Modelling Practice and Theory 54, 49–63.

[7] Duwairi, R., Ammari, H., 2016. An enhanced CBAR algorithm for improving recommendation systems accuracy. Simulation Modelling Practice and Theory 60, 54–68.

[8] Graef, G., Schaefer, C., 2002. Application of ART2 Networks and Self-Organizing Maps to Collaborative Filtering. In: Hypermedia: Openness, Structural Awareness, and Adaptivity. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 296–309.

[9] Harper, F. M., Konstan, J. A., 2016. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems 5 (4), 1–19.

[10] Jakomin, M., Curk, T., Bosni, Z., 2018. Generating inter-dependent data streams for recommender systems. Simulation Modelling Practice and Theory 88, 1–16.

[11] Katarya, R., 2018. Movie recommender system with metaheuristic artificial bee. Neural Computing and Applications 30 (6), 1983–1990.

[12] Lee, M., Choi, P., Woo, Y., 2002. A Hybrid Recommender System Combining Collaborative Filtering with Neural Network. In: Adaptive Hypermedia and Adaptive Web-Based Systems. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 531–534.

[13] Ma, Z., Yang, Y., Wang, F., Li, C., Li, L., 2014. The SOM Based Improved K-Means Clustering Collaborative Filtering Algorithm in TV Recommendation System. In: 2014 Second International Conference on Advanced Cloud and Big Data. pp. 288–295.

[14] Nayak, R., Mirajkar, A., Rokade, J., Wadhwa, G., 2018. Hybrid Recommendation System For Movies 05 (03), 4.

[15] Ponnam, L. T., Deepak Punyasamudram, S., Nallagulla, S. N., Yellamati, S., 2016. Movie recommender system using item based collaborative filtering technique. In: 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS). IEEE, Pudukkottai, India, pp. 1–5.

[16] Pornwattanavichai, A., sakolnagara, P. B. n., Jirachanchaisiri, P., Kitsupapaisan, J., Maneeroj, S., ???? Enhanced tweet hybrid recommender system using unsupervised topic modeling and matrix factorization-based neural network. In: Supervised and Unsupervised Learning for Data Science. Springer International Publishing, pp. 121–143.

[17] Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., Venkatesh, B., 2019. Content-Based Movie Recommendation System Using Genre Correlation. In: Smart Intelligent Computing and Applications. Smart Innovation, Systems and Technologies. Springer, Singapore, pp. 391–397.

[18] Vembu, S., Baumann, S., 2005. A Self-Organizing Map Based Knowledge Discovery for Music Recommendation Systems. In: Computer Music Modeling and Retrieval. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 119–129.

[19] Vozalis, M. G., Margaritis, K. G., 2007. Using svd and demographic data for the enhancement of generalized collaborative filtering. Information Sciences 177 (15), 3017–3037.

[20] Wadia, K., Gupta, P., 2011. Movie recommendation system based on self-organizing maps. The University of Texas at Austin, Austin, Texas.

[21] Walek, B., Fojtik, V., 2020. A hybrid recommender system for recommending relevant movies using an expert system 158, 113452.

[22] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., Leskovec, J., 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '18. Association for Computing Machinery, London, United Kingdom, pp. 974–983.

[23] Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R., 2008. Large-Scale Parallel Collaborative Filtering for the Netflix Prize. In: Algorithmic Aspects in Information and Management. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 337–348.