Instituto Infnet

MIT - Data Science, Data Analytics & Machine Learning

How to recommend movies based on ratings?

Final Assignment

Abordagem híbrida para sistema de recomendação

Renan Rocha e Thiago de Carvalho

problem_statement

Is it possible that a movie recommendation system that suggests items prone to be rated (favorably and off-bubble) in order to organically reduce the sparsity of the data?

`print(sincere)`

Develop a collaborative movie recommendation system, based on ratings that suggest to users 4 movie segments:

Types of movies seen
frequently

TRUE LOVE
(zona de conforto)

AFFAIR
(when love falters)

ONE NIGHT STAND
why not?)

DRUNK FLERT
(I don't recognize myself!)

rarely

print(scope, len(movielens_dataset_ml-25m))

SCOPE

Develop a collaborative and ratings-based filtered movie recommendation system.

- hybrid approach, applying at least 3 algorithms
- to address data sparsity issues

OUT OF SCOPE

- Other classic problems (Scalability, Cold Start)
- Shared accounts: Multiple users/screen usage
- Heavy users/bots

25MM ratings

162K users

62K movies

95% sparsity

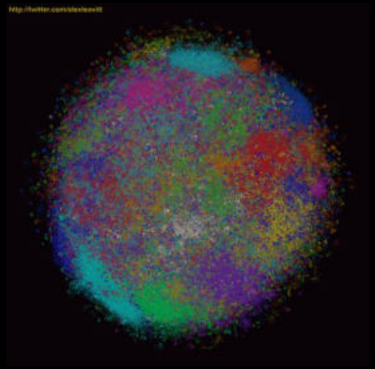
concepts.restore(memory_cards)

Sparsity index

Index ranging from 1 to 0, the higher the index, the more sparse (with few ratings / total possible ratings) the item or user is.

$$sparsity = 1 - \frac{count\_nonzero(A)}{total\_elements\_of\_A}$$

Fuzzy clustering



A form of clustering where each data point can belong to more than one group.

NPS – Net promoter score

Metric that shows the likelihood that users will recommend a company, product or service to a friend or colleague - developed by Fred Reichheld.

SVD

Matrix factoring algorithm, popularized by Simon Funk during the Netflix Awards, which is equivalent to Probabilistic Matrix Factoring when baselines are not used.

For Sincere, the hyperparameter of latent factors was set to 75 neighbors. This means, extracting features and correlation from the matrix of 75 closest user items.

Benchmarks and references

- Najafabadi et Al. - An Effective Collaborative User Model Using Hybrid Clustering Recommendation Methods
- Mohammed Fadhel Aljunid, Manjaiah DH - An Efficient Deep Learning Approach for Collaborative Filtering Recommender System
- Nicholas Becker - https://beckernick.github.io/

pip update recommender_systems (a)

Sequential application of two different clustering methods:

1st (hard clustering) items are grouped by k means to reduce disparity

2nd (soft clustering) user grouping using fuzzy c means.

[!] The output of the k means is used as input of the fuzzy, quantity of movies in each cluster seen per user attribute differentiates them regarding usage.

Ingénierie des Systèmes d'Information
Vol. 26, No. 2, April, 2021, pp. 151-158
Journal homepage: http://iieta.org/journals/isi

An Effective Collaborative User Model Using Hybrid Clustering Recommendation Methods

Maryam Khanian Najafabadi[1*], Azlinah Mohamed[2], Madhavan A/L Balan Nair[1], Sayed Mojtaba Tabibian[1]

[1] Department of Internet Engineering & Computer Science, Lee Kong Chian Faculty of Engineering & Science, Universiti Tunku Abdul Rahman, Kajang 43000, Malaysia
[2] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam 40450, Malaysia

Corresponding Author Email: maryamkn@utar.edu.my

https://doi.org/10.18280/isi.260202

Received: 27 January 2021
Accepted: 29 March 2021

Keywords:
recommendation system, fuzzy clustering, collaborative filtering, sparsity

ABSTRACT

Collaborative Filtering (CF) has been known as the most successful recommendation technique in which recommendations are made based on the past rating records from like-minded users. Significant growth of users and items have negatively affected the efficiency of CF and pose key issues related to computational aspects and the quality of recommendation such as high dimensionality and data sparsity. In this study, a hybrid method was proposed and was capable to solve the mentioned problems using a neighborhood selection process for each user through two clustering algorithms which were item-based k-means clustering and user-based Fuzzy Clustering. Item-based k-means clustering was applied because of its advantages in computational time and hence it is able to address the high dimensionality issues. To create user groups and find the correlation between users, we employed the user-based Fuzzy Clustering and it has not yet been used in user-based CF clustering. This clustering can calculate the degree of membership among users into set of clustered items. Furthermore, a new similarity metric was designed to compute the similarity value among users with affecting the output of user-based Fuzzy Clustering. This metric is an alternative to the basic similarity metrics in CF and it has been proven to provide high-quality recommendations and a noticeable improvement on the accuracy of recommendations to the users. The proposed method has been evaluated using two benchmark datasets, MovieLens and LastFM in order to make a comparison with the existing recommendation methods.

1. INTRODUCTION

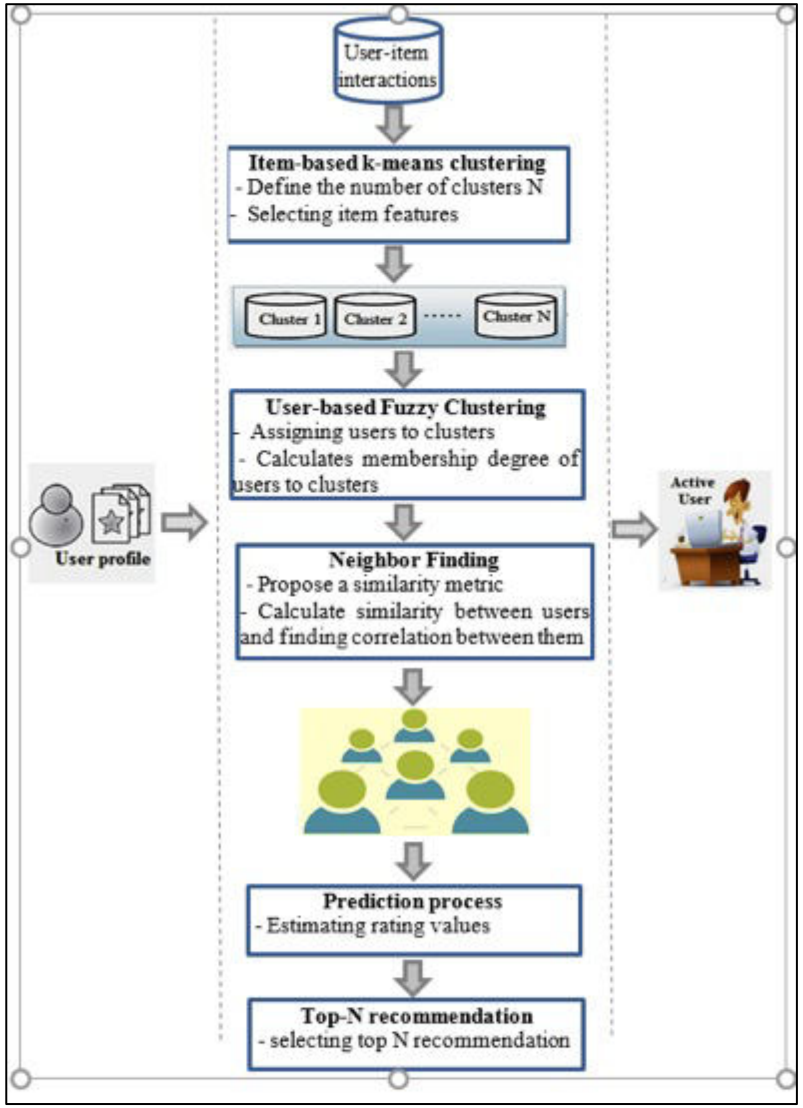
A recommender system provides a personalized set of recommendations by incorporating users' needs into a user model and applying suitable recommendation algorithms in mapping the user model into targeted item recommendations [1-3]. Due to the advancement in Internet technology, the development of recommender systems in e-commerce sites for product purchase advice is becoming more significant. This is due to its ability to save users' time and effort in searching for items [4-6].

Recent works have showed that to provide high-quality recommendations, the similarity metrics design have to be innovative and artificial learning machine and artificial intelligence ought to be employed [7, 8]. The major challenge is to accurately discover users' interests through creating a proper user model. In doing this, it is significant to identify the computation times which is necessary for defining the relations among users or items that can be regarded as performance issue of the recommender systems due to the large numbers of items or users. Moreover, there are drawbacks of CF recommendation systems that need to be addressed in increasing the quality of recommendation and accuracy of the predicted rated. These drawbacks are high dimensionality, data sparsity, and cold-start [9-12]. Most of the proposed recommender systems in solving drawbacks of CF failed to take action based on both sides of similarity (similarity among users and items) and it was discovered that the amount of time spent in calculating similarity among users or items to produce recommendations was extended. With the goal of reducing the execution of time with the number of bit processing, this study proposes a hybrid recommender system with a new similarity measurement method that combines the calculation of similarity between items and users in predicting the score of active users on unseen items.

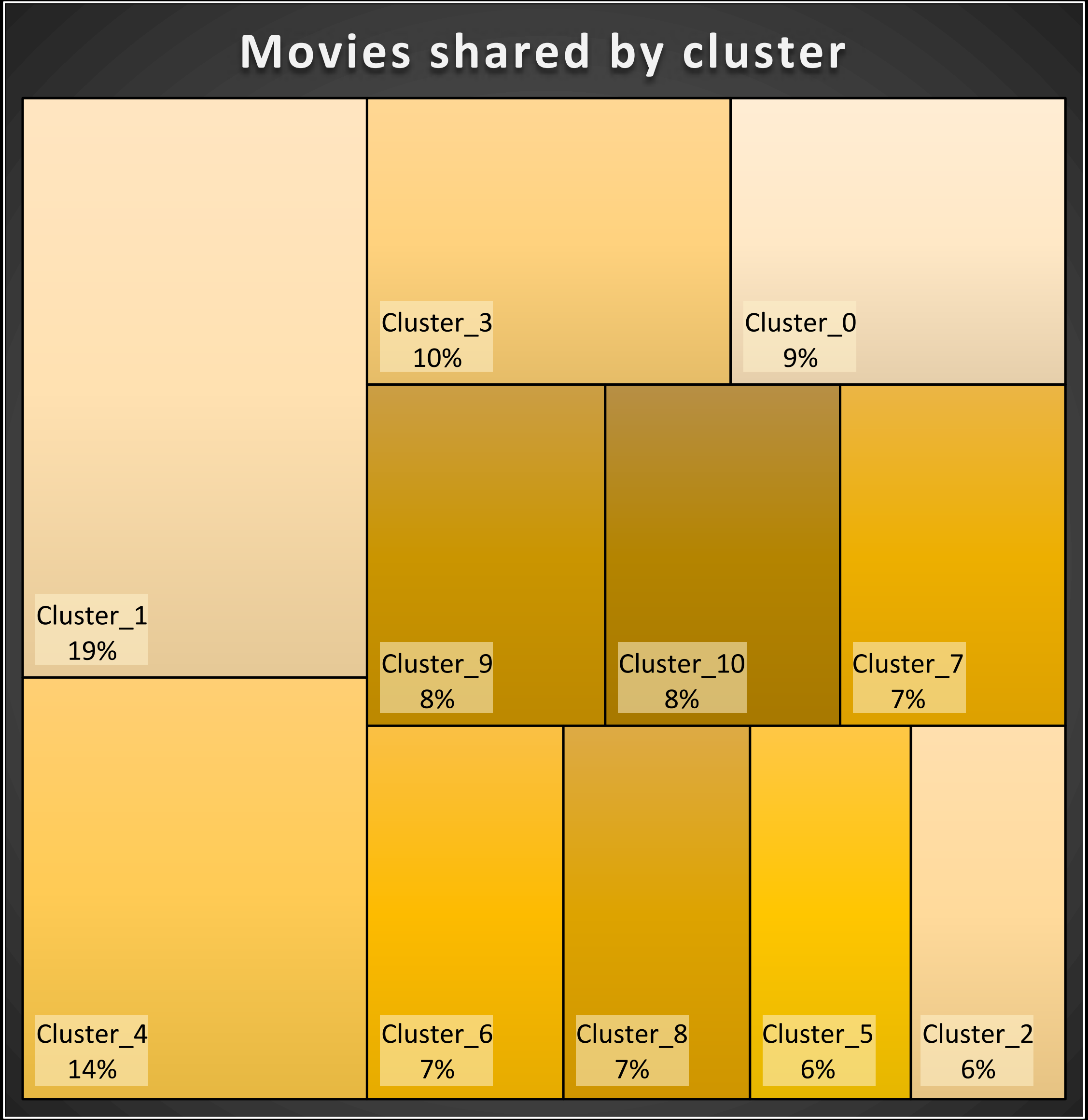
The motivation and contribution of this study will be presented in sub-section 1.1. This paper is organized into the following sections: Section 2 briefly provides reviews on previous works on recommender systems and the clustering techniques. Section 3 presents the research methodology used in this study. The proposed recommendation method and experiment methodology will be described in the following subsections (3.1 and 3.2). Section 4 describes results of the experiment conducted. Section 5 outlines the conclusions and future direction of this work.

1.1 Motivation

One of the most successful clustering techniques to overcome the issues of CF is fuzzy c-means. In fact, there are research methodologies developed to increase the quality of recommendations that apply fuzzy C-means clustering in CF. However, these research methodologies have not yet been applied in user's modeling for making recommendations and none of those concentrate on execution time that is required to calculate the similarity of active users among the existing users



print(movie_clustering_results)



	Movie_cluster_0	Movie_cluster_1	Movie_cluster_2	Movie_cluster_3
movie age	-	-	old	youth
popularity ML	very rated	few ratings	very rated	few ratings
popularity IMDB	-	awarded	-	higher rates
NPS ML	-	higher satisfaction	-	high satisfaction
Genres	Crime	Biography	Animation	Biography
	Film-Noir	Sport	Family	Documentary
	Mystery	War	Fantasy	Music
	Thriller	Drama	Musical	News
	-	-	Short	Sport

	Movie_cluster_4	Movie_cluster_5	Movie_cluster_6	Movie_cluster_7
movie age	old	youth	-	-
popularity ML	-	-	-	-
popularity IMDB	-	-	-	lower rating
NPS ML	-	-	low satisfaction	low satisfaction
Genres	Comedy	Mystery	Comedy	Adult
	Musical	Thriller	-	Horror
	Reality-TV	-	-	Sci-Fi
	Talk-Show	-	-	-
	Western	-	-	-

	Movie_cluster_8	Movie_cluster_9	Movie_cluster_10
movie age	-	old	-
popularity ML	lots of ratings	-	-
popularity IMDB	-	awarded	-
NPS ML	lower satisfaction	-	low satisfaction
Genres	Action	Romance	Romance
	Adventure	-	Musical
	Sci-Fi	-	-
	-	-	-
	-	-	-

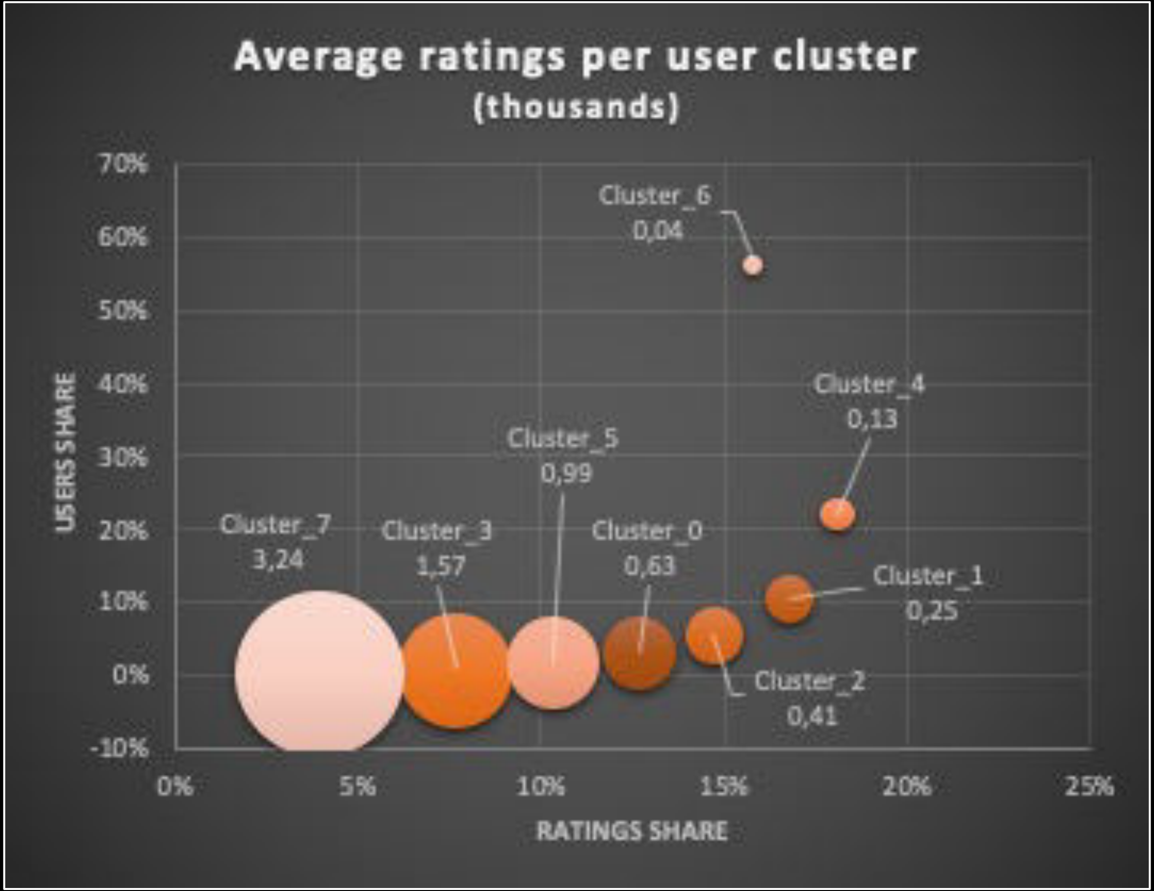
Clustering metrics used to define k=11: Elbow, Silhouette, David Boudin and Dendrogram

Fuzzy C-means

	movie_cluster_0.0	movie_cluster_1.0	movie_cluster_2.0	movie_cluster_3.0	movie_cluster_4.0	movie_cluster_5.0	movie_cluster_6.0	movie_cluster_7.0	movie_cluster_8.0	movie_cluster_9.0	movie_cluster_10.0
0	6.0	21.0	2.0	1.0	7.0	3.0	8.0	0.0	2.0	13.0	7.0
1	16.0	47.0	21.0	0.0	16.0	9.0	8.0	1.0	43.0	9.0	14.0
2	92.0	63.0	49.0	1.0	62.0	61.0	22.0	22.0	237.0	16.0	31.0
3	18.0	15.0	31.0	5.0	35.0	17.0	1.0	4.0	110.0	1.0	5.0
4	14.0	18.0	10.0	0.0	16.0	4.0	11.0	1.0	12.0	6.0	9.0
...	...	...	...	...	...	...	...	...	...	...	...
162536	6.0	11.0	8.0	0.0	17.0	4.0	10.0	2.0	9.0	10.0	24.0
162537	6.0	14.0	14.0	0.0	21.0	7.0	11.0	2.0	14.0	30.0	35.0
162538	3.0	10.0	2.0	0.0	3.0	4.0	4.0	0.0	12.0	7.0	2.0
162539	5.0	10.0	20.0	0.0	2.0	7.0	3.0	6.0	17.0	8.0	10.0
162540	15.0	26.0	17.0	1.0	31.0	6.0	14.0	4.0	40.0	14.0	14.0

Fuzzy Matrix		User Fuzzy C Means Clustering							
		0	1	2	3	4	5	6	7
User_cluster_Label	0	45%	9%	22%	2%	5%	12%	4%	0%
	1	2%	51%	14%	0%	22%	0%	9%	0%
	2	14%	22%	47%	0%	9%	2%	5%	0%
	3	9%	4%	6%	49%	3%	21%	3%	5%
	4	0%	13%	2%	0%	61%	0%	23%	0%
	5	20%	6%	9%	12%	4%	46%	3%	0%
	6	0%	1%	0%	0%	10%	0%	89%	0%
	7	6%	5%	5%	17%	4%	9%	4%	50%

print(user_clustering_results)

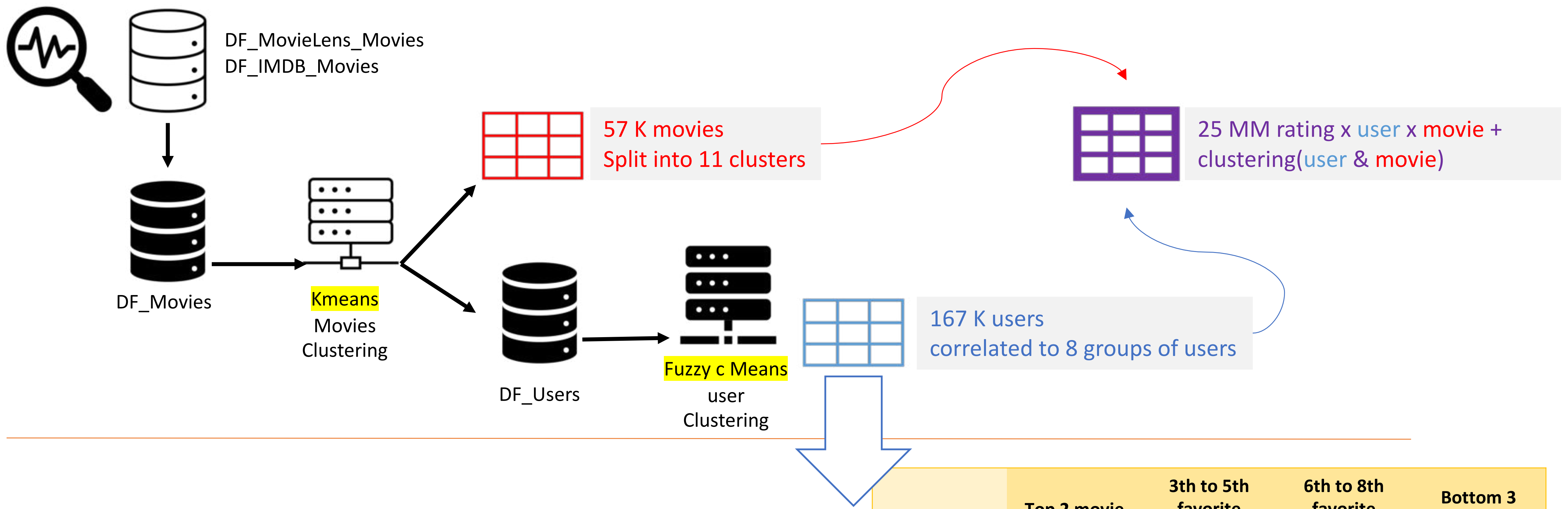


		User_cluster_0	User_cluster_1	User_cluster_2	User_cluster_3
	Favorite day	Monday Wednesday	Saturday Monday	Saturday Monday	Tuesday -
	Favorite time	afternoon	morning	morning	evening
	Voter profile	neutral mid voter	neutral mid voter	neutral mid voter	detractor heavy voter
Movie Clusters	True love movie clusters	8, 0	8, 4	0, 1	8, 1
	Affair movie clusters	2, 1, 9	1, 0, 10	8, 4, 10	0, 5, 7
	1 night stand movie clusters	4, 10, 5	2, 7, 6	5, 6, 2	2, 4, 9
	Drunk flert movie clusters	6, 7, 3	5, 9, 3	9, 7, 3	10, 6, 3

		User_cluster_4	User_cluster_5	User_cluster_6	User_cluster_7
	Favorite day	not Monday -	Wednesday -	Thursday Friday	Saturday Wednesday
	Favorite time	afternoon	morning	night	evening
	Voter profile	promoter lower voter	detractor mid voter	promoter lower voter	neutral heavy voter
Movie Clusters	True love movie clusters	0, 8	8, 1	8, 4	0, 1
	Affair movie clusters	1, 10, 2	10, 2, 4	0, 1, 2	8, 4, 10
	1 night stand movie clusters	4, 5, 9	9, 0, 5	10, 6, 5	5, 6, 9
	Drunk flert movie clusters	6, 7, 3	6, 7, 3	7, 3, 9	3, 2, 7

plt.sincere_setup.show()

Setup



Output

User Cluster	Top 2 movie clusters	3th to 5th favorite movie clusters	6th to 8th favorite movie clusters	Bottom 3 movie clusters
1	1, 2	4,5,6	7,8,9	3,10,11
2	2,5	1,3,9	4,8,10	6, 7,11
…	…	…	…	…
n	6,9	7,8,10	2,3,11	1,4,5

pip update recommender_systems (b)

A hybrid recommender system that is established based on matrix factoration (...) that works on implicit feedbacks from users and also auxiliary information from both users and items.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 829–836

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

An Efficient Deep Learning Approach for Collaborative Filtering Recommender System

Mohammed Fadhel Aljunid[1], Manjaiah DH[b]

[a]*Department of Computer Science, Mangalore University,India*
[b]*Department of Computer Science,Mangalore University,India*

Abstract

Owing to the enormous growth in information over the past few decades, the world has become a global village. The recommendation system remains the most widely used type of commercial websites. The personalized recommender system is of paramount importance in modeling user's preference on items based on their past interactions (e.g., ratings and clicks), known as collaborative filtering (CF) technique. Although CF is very important among the algorithms used in recommendation systems, it suffers some setbacks such as the sparsity of matrix ratings, scalability, and integrals nature of data. Several research studies have shown that the above-mentioned obstacle could be tackled with the help of matrix factorization (MF) techniques. In spite of the fact that the technique is likely to suffer from lack of some meaningful signals by using a low ranked approximation as well as lack of sparsity in times of denser singular vectors. Recently, deep learning techniques have proven to learn good representation in natural language processing, image classification, and so on. In this work, we propose a deep learning method of collaborative recommender systems (DLCRS). We have made a comparative study of the proposed method and existing methods. Experimental results demonstrate that our approach gives improved results compared to already existing methods. We empirically evaluate DLCRS on two famous datasets: 100K and 1M Movielens.

Keywords: Recommender System , Collaborative Filtering , Matrix Factorization , Deep Learning , Movielens Datasets;

1. Introduction

The advancement of artificial intelligence and machine learning technologies has brought intelligent products that are essential in providing access to various endeavors of peoples' day-to-day life. Effective and useful information

* Corresponding author. Tel.: +917304385644 .
E-mail address: ngm505@yahoo.com

1877-0509 © 2020 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the Third International Conference on Computing and Network Communications (CoCoNet'19).
10.1016/j.procs.2020.04.090

plt.sincere_schema.show()

32 FACTORED MATRICES VIA SVD

8 USER CLUSTERS
*
4 SEGMENTS

- TRUE LOVE
- AFFAIR
- 1 NIGHT STAND
- DRUNK FLERT

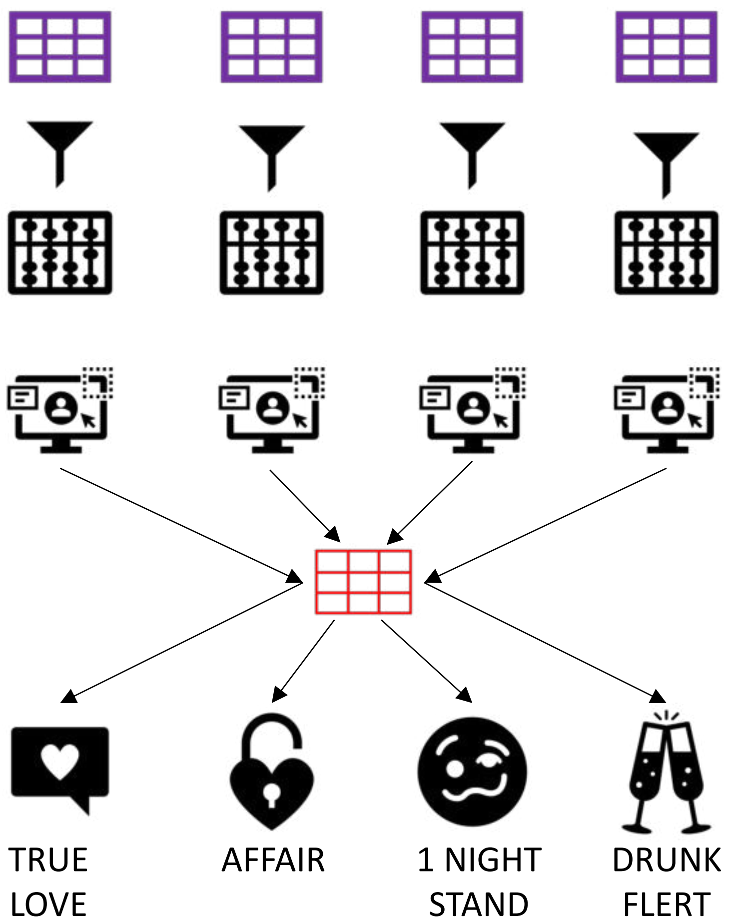
User Cluster	Top 2 movie clusters	3th to 5th favorite movie clusters	6th to 8th favorite movie clusters	Bottom 3 movie clusters
2	2,5	1,3,9	4,8,10	6,7,11

Filter 25 MM ratings database by user cluster and movie cluster

Factorization Matrix via (SVD)

Making Predictions from the Decomposed Matrices

Making Movie Recommendations



print(sincere)

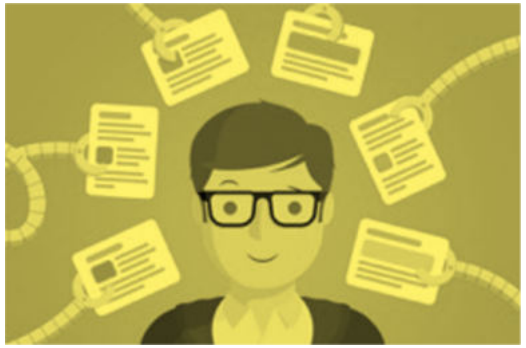
Sincere recommendations

recommending and checking results

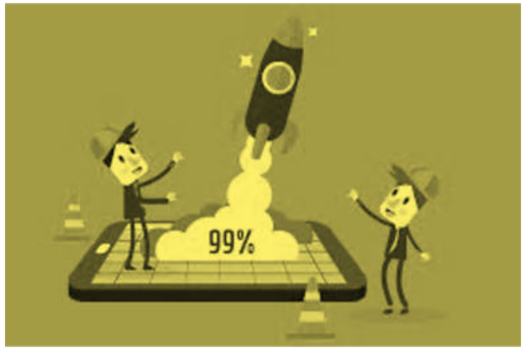
tests.describe( )

Sampling

Recommending



Checking results



Target

1 Sample of 80 users. 10 from each cluster with high fuzzy index.

Recommendation of 20 movies (5 for each affinity segment) using Sincere.

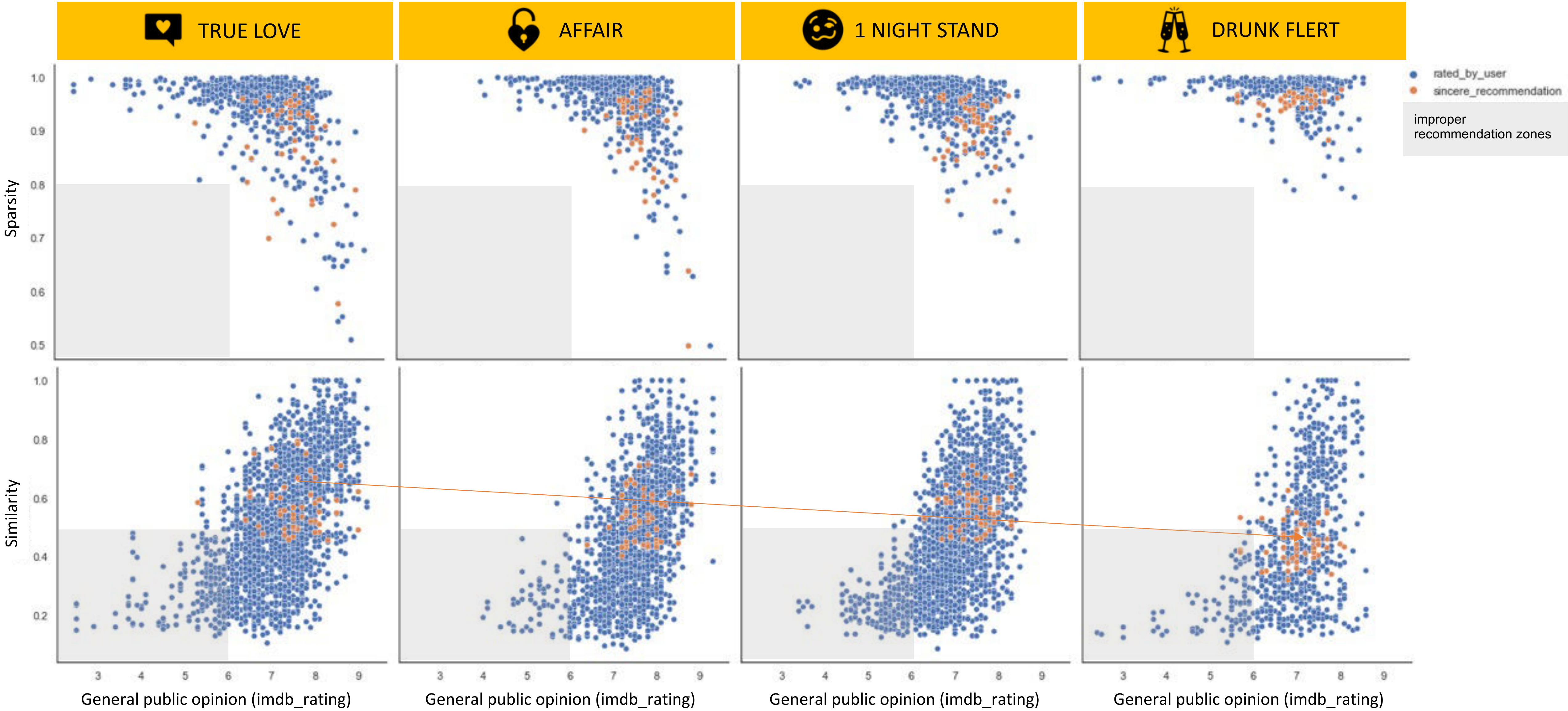
Analyze the similarity and sparsity of the recommendations

Did Sincere recommend movies with sparse ratings?

Were the recommended movies relevant to the general public?

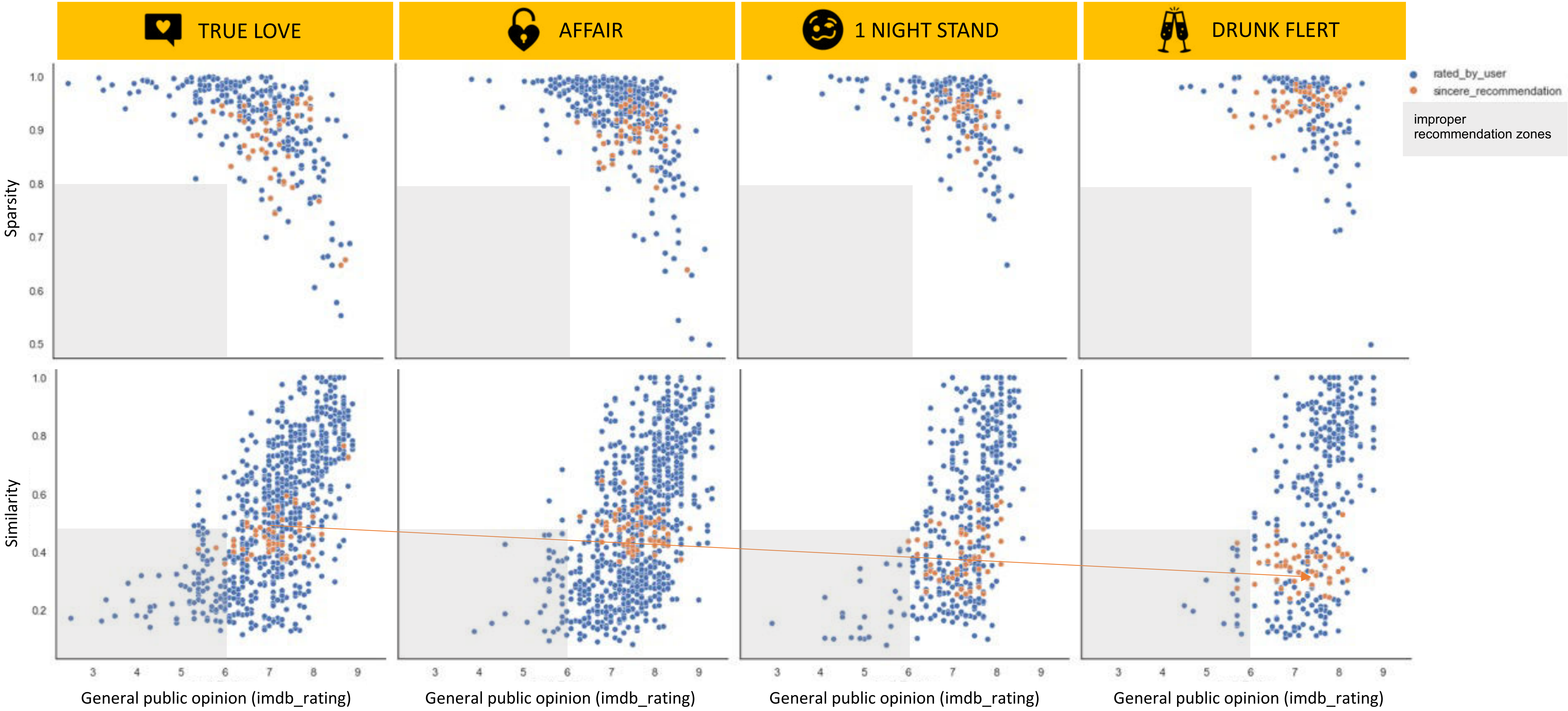
plt.sincere_results(Cluster_0)

For the densest user groups (0, 1, 2, 3, 5 and 7), Sincere recommended movies with high sparsity and with an overall (IMDB) rating higher than 6. The relative similarity to the user's common interest is decreasing according to the recommendation segment, as expected.



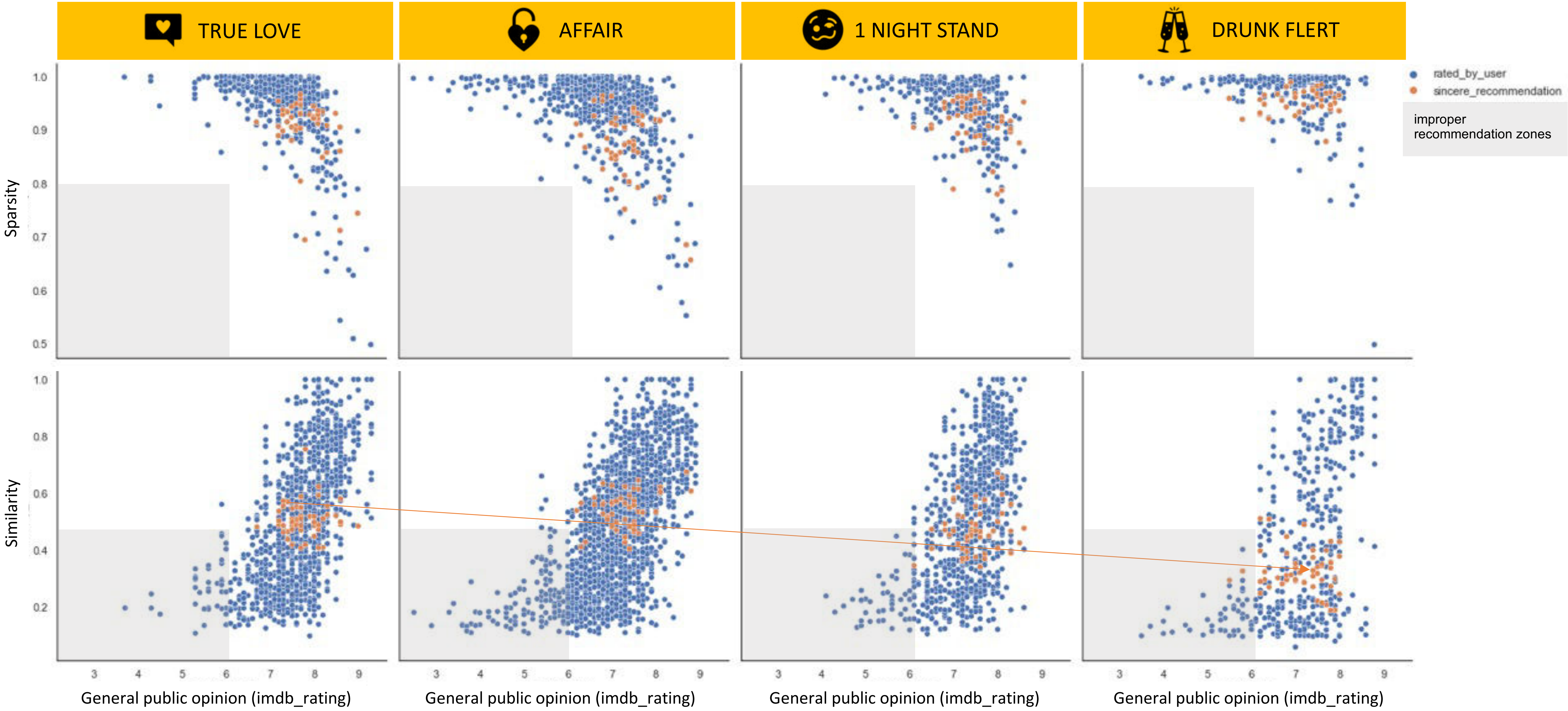
plt.sincere_results(Cluster_1)

For the densest user groups (0, 1, 2, 3, 5 and 7), Sincere recommended movies with high sparsity and with an overall (IMDB) rating higher than 6. The relative similarity to the user's common interest is decreasing according to the recommendation segment, as expected.



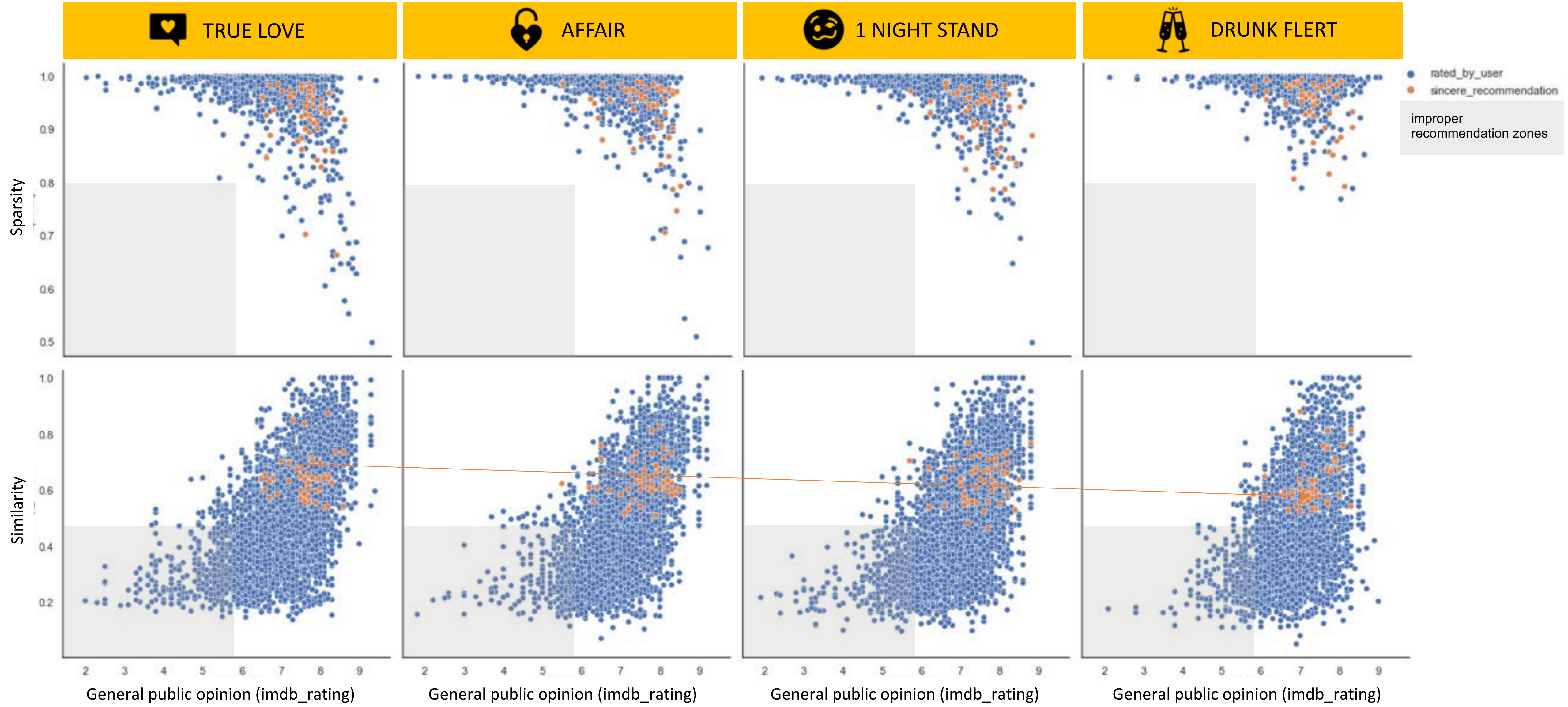
plt.sincere_results(Cluster_2)

For the densest user groups (0, 1, 2, 3, 5 and 7), Sincere recommended movies with high sparsity and with an overall (IMDB) rating higher than 6. The relative similarity to the user's common interest is decreasing according to the recommendation segment, as expected.



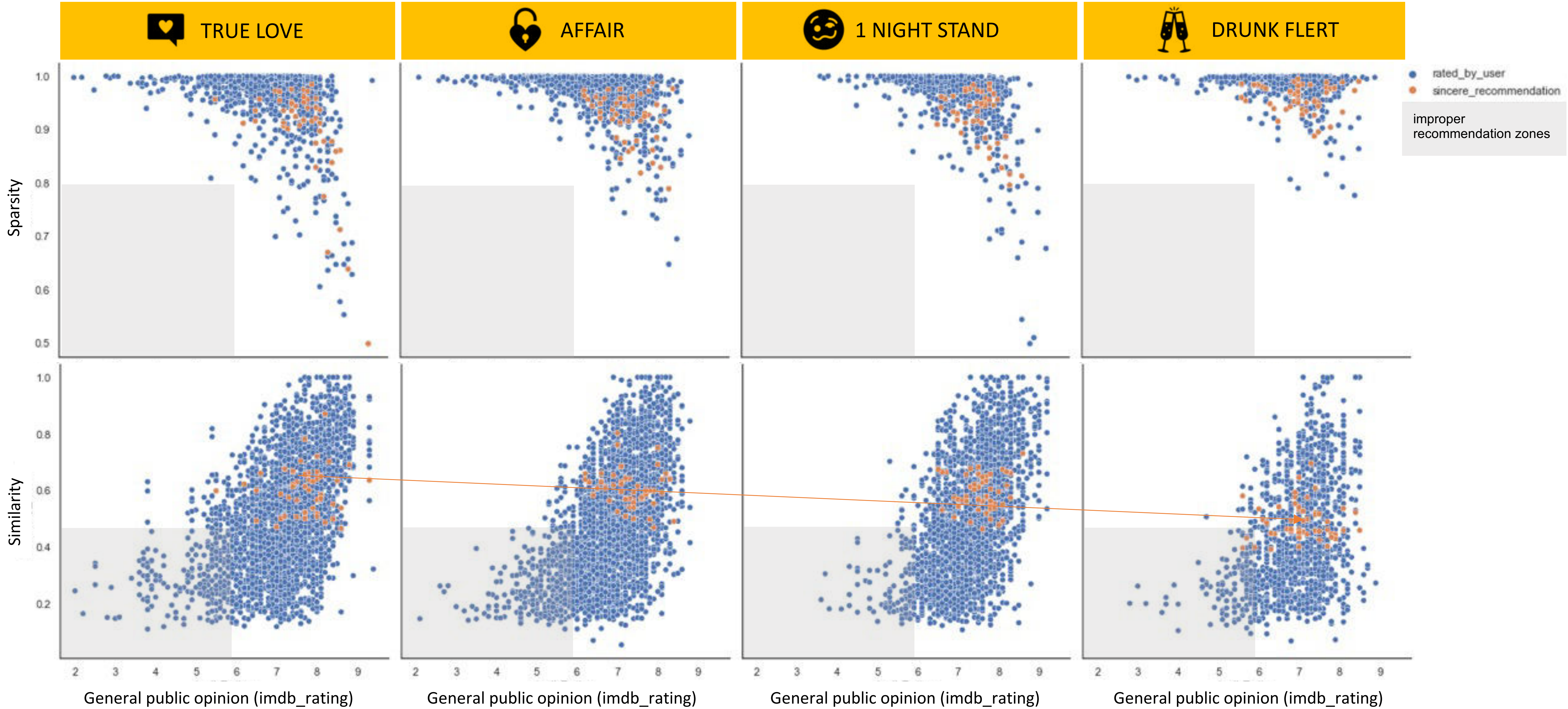
plt.sincere_results(Cluster_3)

For the densest user groups (0, 1, 2, 3, 5 and 7), Sincere recommended movies with high sparsity and with an overall (IMDB) rating higher than 6. The relative similarity to the user's common interest is decreasing according to the recommendation segment, as expected.



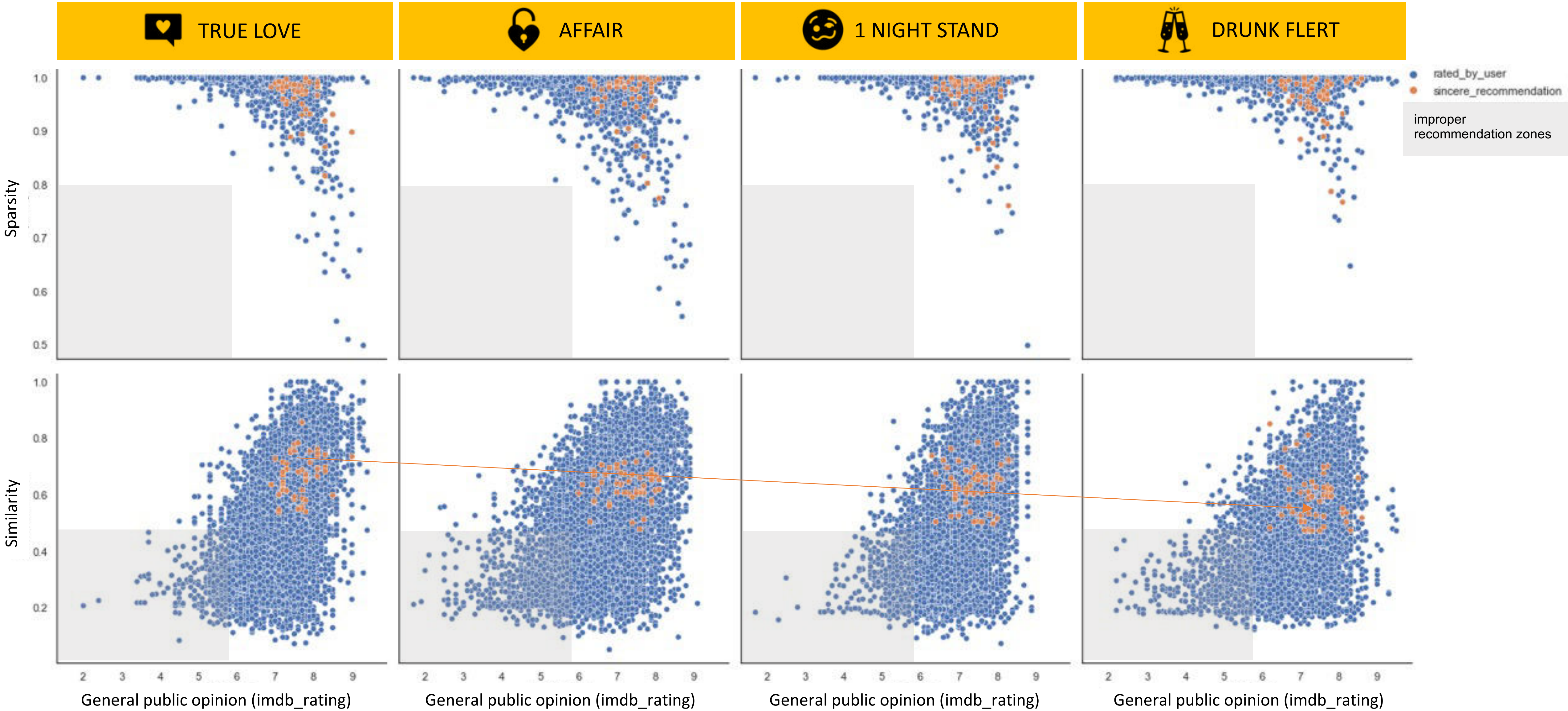
plt.sincere_results(Cluster_5)

For the densest user groups (0, 1, 2, 3, 5 and 7), Sincere recommended movies with high sparsity and with an overall (IMDB) rating higher than 6. The relative similarity to the user's common interest is decreasing according to the recommendation segment, as expected.



plt.sincere_results(Cluster_7)

For the densest user groups (0, 1, 2, 3, 5 and 7), Sincere recommended movies with high sparsity and with an overall (IMDB) rating higher than 6. The relative similarity to the user's common interest is decreasing according to the recommendation segment, as expected.



Example.({userId:130311,cluster:0})

	TRUE LOVE	AFFAIR	1 NIGHT STAND	DRUNK FLERT

Rated by user

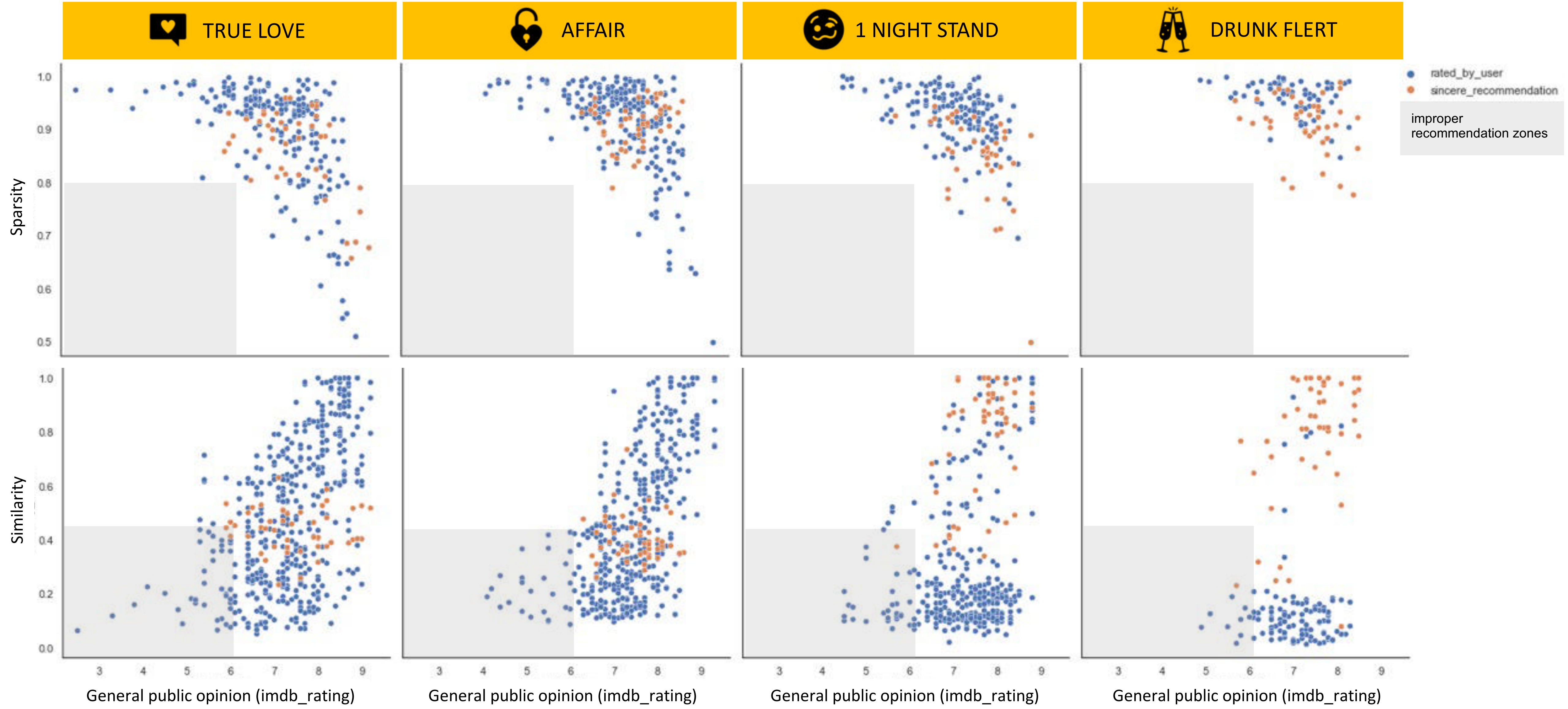
	TRUE LOVE	AFFAIR	1 NIGHT STAND	DRUNK FLERT
Imdb	9.0	9.3	8.6	8.4
Sparsity	0.99	0.99	0.99	0.99

Sincere Recommendation

	TRUE LOVE	AFFAIR	1 NIGHT STAND	DRUNK FLERT
Imdb	8.1	7.3	8.3	7.7
Sparsity	0.96	0.94	0.91	0.97
Similarity	0.61	0.43	0.56	0.55

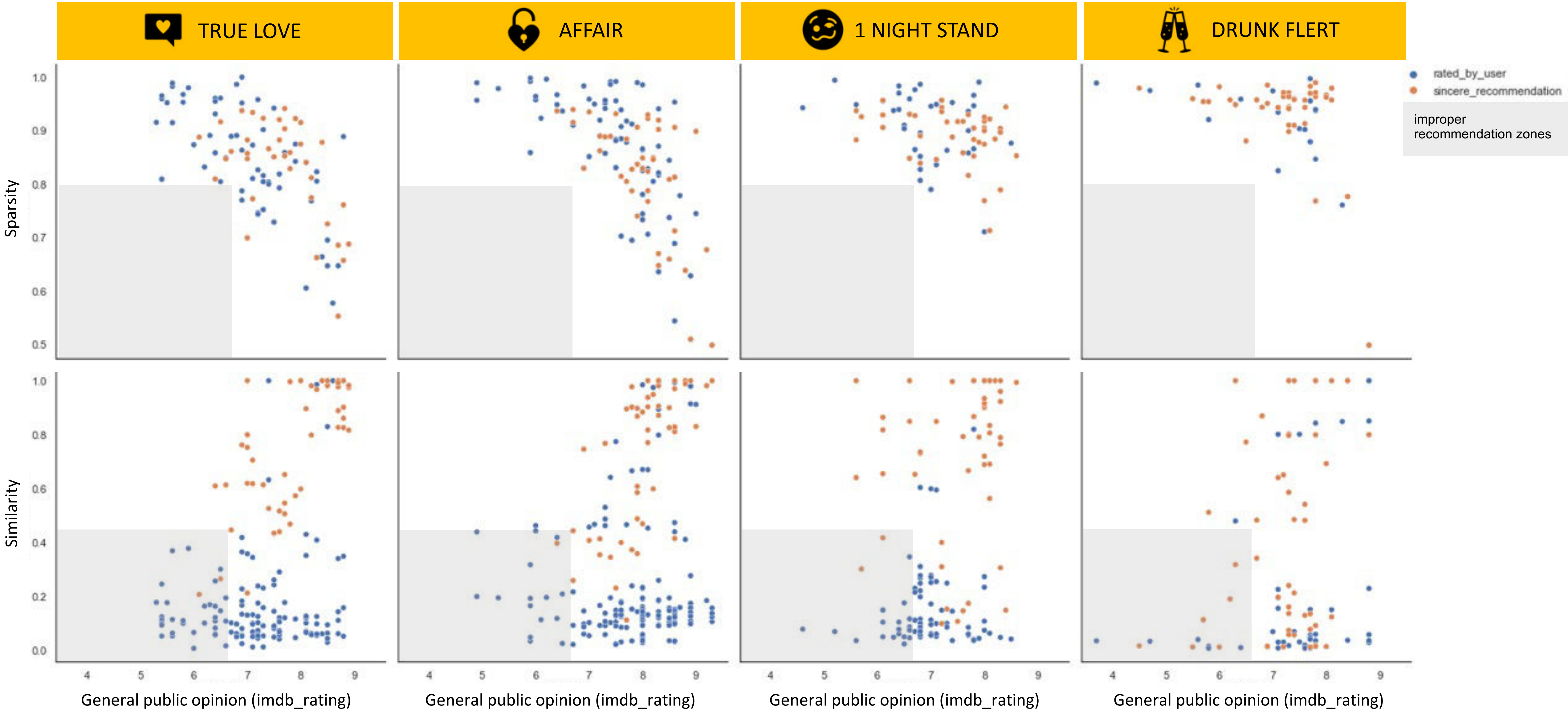
plt.sincere_results(Cluster_4)

For clusters with increased sparsity (user clusters 4 and 6), Sincere tried to properly recommend movies following our sparsity removal guidelines by choosing options with better overall (IMDB) ratings and relative similarity to the user's common interest.



plt.sincere_results(Cluster_6)

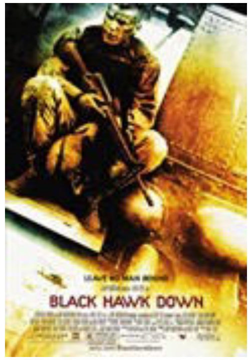
For clusters with increased sparsity (user clusters 4 and 6), Sincere tried to properly recommend movies following our sparsity removal guidelines by choosing options with better overall (IMDB) ratings and relative similarity to the user's common interest.

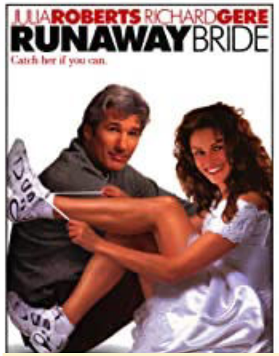


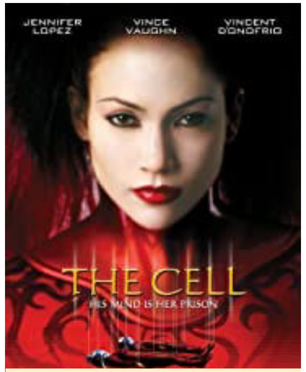
Example.({userId:135548,cluster:6})

	TRUE LOVE	AFFAIR	1 NIGHT STAND	DRUNK FLERT

Rated by user

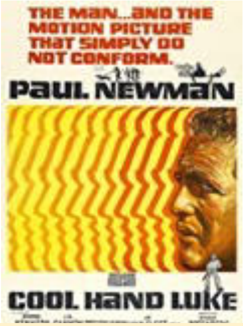


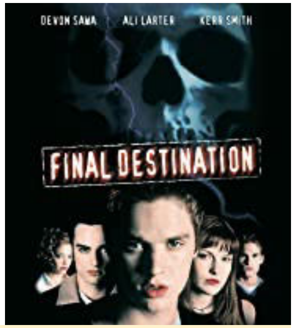




	TRUE LOVE	AFFAIR	1 NIGHT STAND	DRUNK FLERT
Imdb	8.7	8.8	7.8	8.7
Sparsity	0.96	0.96	0.95	0.95
Similarity	0.16	0.66	0.58	0.84

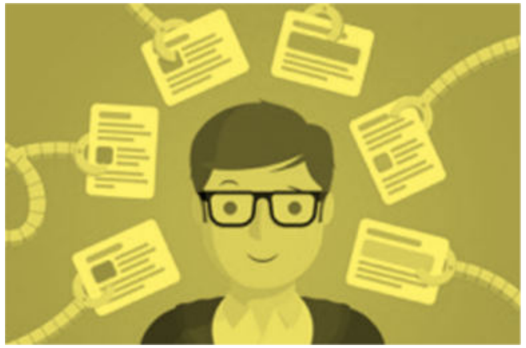
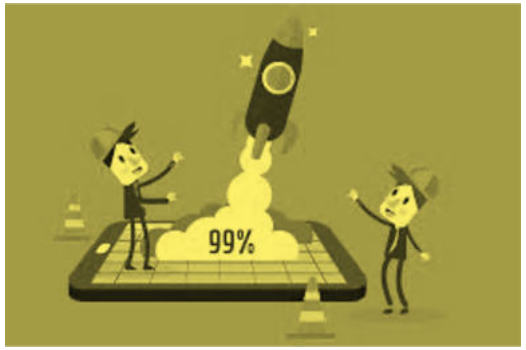
Sincere Recommendation





	TRUE LOVE	AFFAIR	1 NIGHT STAND	DRUNK FLERT
Imdb	8.1	8.0	8.3	7.7
Sparsity	0.96	0.97	0.91	0.97
Similarity	1	1	1	1

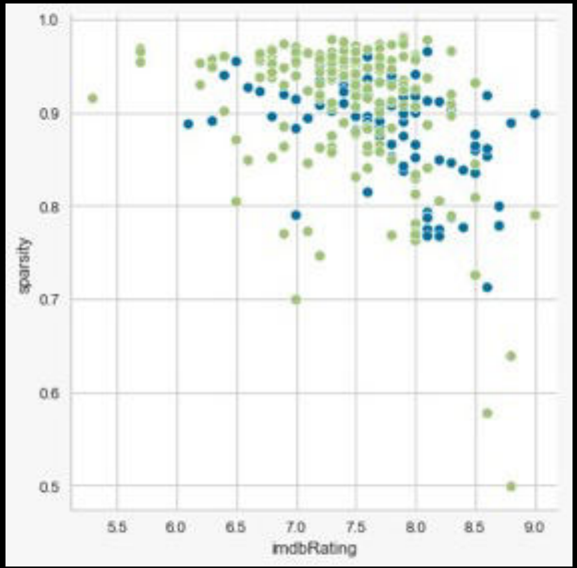
tests.describe( )

	Sampling	Recommending	Checking results	Target
				
1	Sample of 80 users. 10 from each cluster with high fuzzy index.	Recommendation of 20 movies (5 for each affinity segment) using Sincere.	Analyze the similarity and sparsity of the recommendations	Did Sincere recommend movies with sparse ratings?
2	Split the database to identify the 10% of users with the highest fuzzy index. (With the sample used in test 1 included).	Recommending 20 movies to the same 80 users with the traditional recommendation system.	Compare sincere results and the common recommendation system.	Were the recommended movies relevant to the general public?

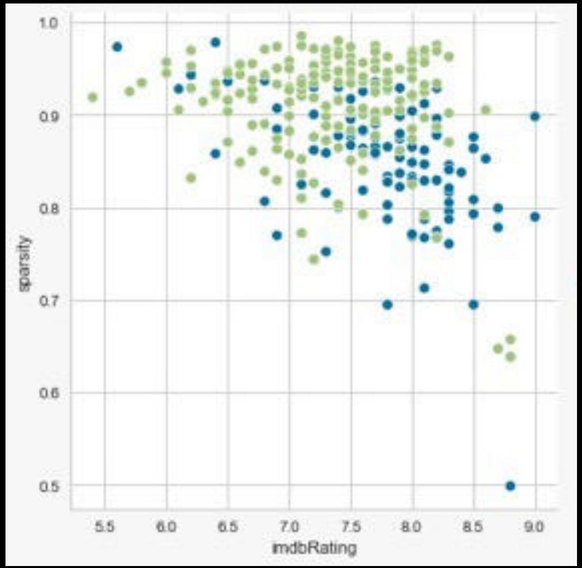
sparsity_vs_rating.compare()

ordinary recommendations
sincere recommendations

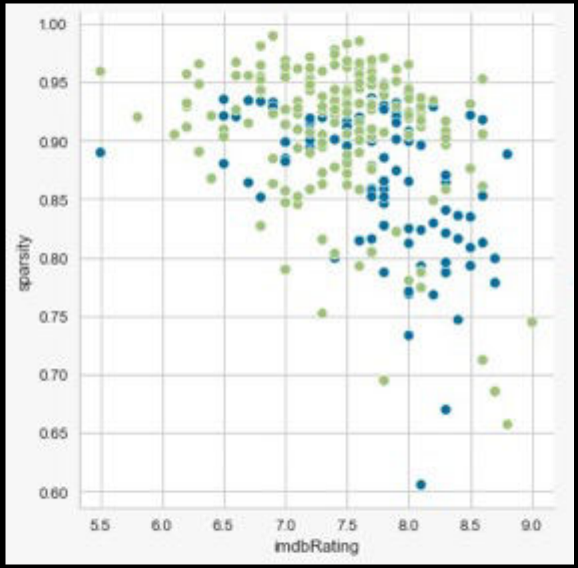
Cluster 0



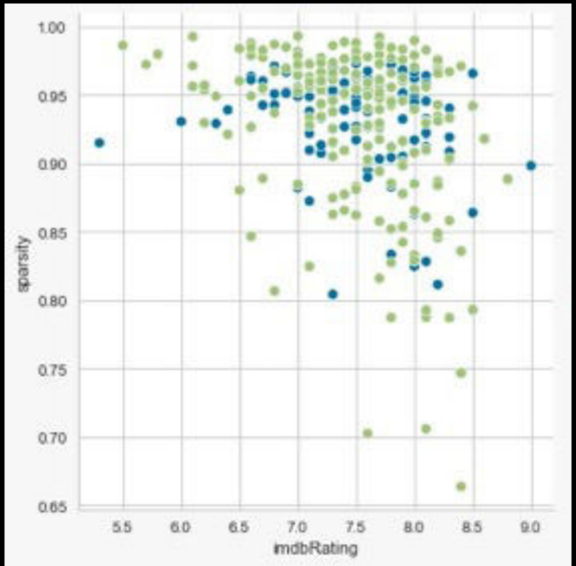
Cluster 1



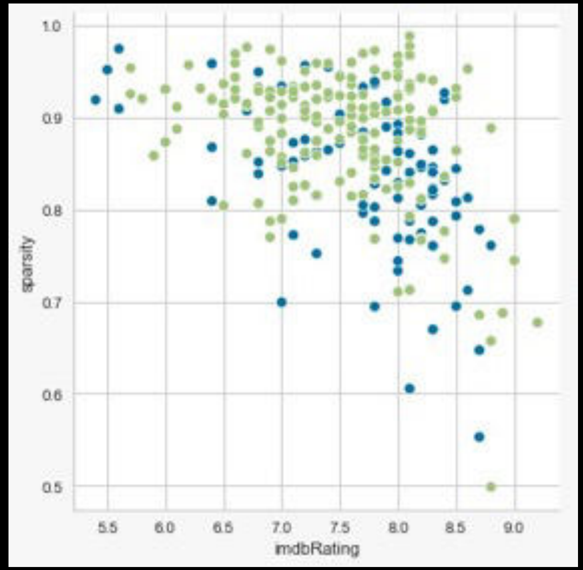
Cluster 2



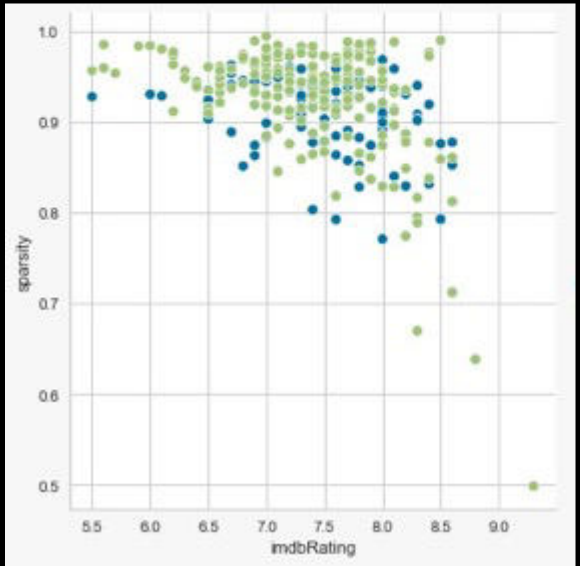
Cluster 3



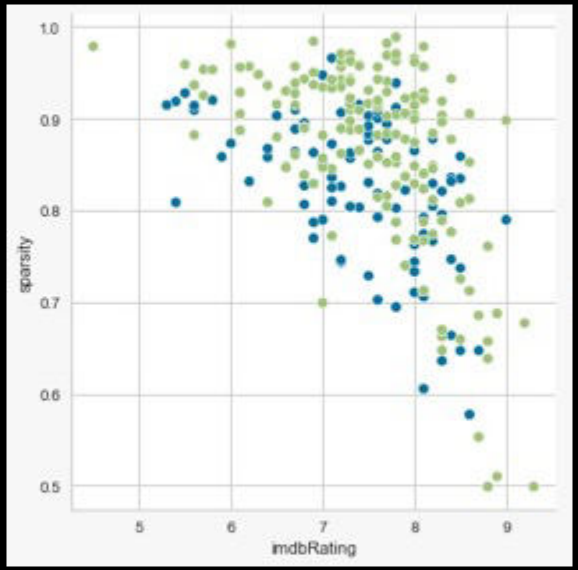
Cluster 4



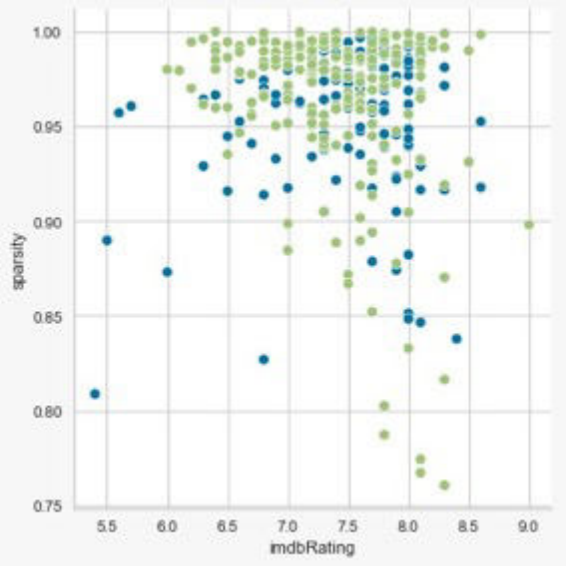
Cluster 5



Cluster 6



Cluster 7



print(sincere_key_findings)

`print(f'conclusion: {conclusion}')`

The SINCERE experiment, with its preliminary results suggests that:

- It is possible to implement recommendation mechanisms that reduce sparsity without affecting user satisfaction.
- There is no silver bullet to recommend correctly for all users, it is convenient to mix techniques.
- Recommender systems can also be a weapon to burst the bubble of cultural segregation in networks.
- Splitting the dataset by 4 segments of similar movies can also overcame some computational limitations.

Evoluções do trabalho

- Enrich the model with user descriptive data
- Review user clustering especially for clusters 6 and 4 in order to maximize the benefits of fuzzy.
- Test other factorization methods: PMF, SVD++, NMF
- Apply other traditional recommendation models and compare with Sincere results.
- Deploy the input based on online research in order to cluster new users

Business insights

- Voter profile analysis can generate more engagement and satisfaction when notifying users.
- Promotional campaign for voters can stimulate good results against sparsity [example: invitations to exclusive premiers].

Sincerely, Yours

Renan Rocha
renanbdr@hotmail.com
Github: renanbdr

Linkedin

Thiago de Carvalho
thiago_de_carvalho@outlook.com.br
Github: ThiagoCarvalho-81

Linkedin