Please cite this paper as:

Leal, F., García-Méndez, S., Malheiro, B., Burguillo, J.C. (2022). Explanation Plug-In for Stream-Based Collaborative Filtering. In: Rocha, A., Adeli, H., Dzemyda, G., Moreira, F. (eds) Information Systems and Technologies. WorldCIST 2022. Lecture Notes in Networks and Systems, vol 468. Springer, Cham. https://doi.org/10.1007/978-3-031-04826-5_5

Accepted version

General rights

Explanation Plug-in for Stream-based Collaborative Filtering

Fátima Leal[1], Silvia García-Méndez[2], Benedita Malheiro[3,4], and Juan C. Burguillo[2]

[1] REMIT, Universidade Portucalense, Porto, Portugal
[2] Information Technologies Group, atlanTTic, University of Vigo, Spain
[3] INESC TEC, Porto, Portugal
[4] School of Engineering, Polytechnic Institute of Porto, Porto, Portugal
fatimal@upt.pt, sgarcia@gti.uvigo.es, mbm@isep.ipp.pt,
J.C.Burguillo@uvigo.es

Abstract. Collaborative filtering is a widely used recommendation technique, which often relies on rating information shared by users, *i.e.*, crowdsourced data. These filters rely on predictive algorithms, such as, memory or model based predictors, to build direct or latent user and item profiles from crowdsourced data. To predict unknown ratings, memory-based approaches rely on the similarity between users or items, whereas model-based mechanisms explore user and item latent profiles. However, many of these filters are opaque by design, leaving users with unexplained recommendations. To overcome this drawback, this paper introduces Explug, a local model-agnostic plug-in that works alongside stream-based collaborative filters to reorder and explain recommendations. The explanations are based on incremental user Trust & Reputation profiling and co-rater relationships. Experiments performed with crowdsourced data from TripAdvisor show that Explug explains and improves the quality of stream-based collaborative filter recommendations.

Keywords: Data Streams, Explanations, Recommendations, Trust & Reputation

1 Introduction

Many data-intensive Artificial Intelligence (AI) systems are currently interweaved in crucial decision processes. Those processes interpret, predict, and discover hidden knowledge with the help of Machine Learning (ML) algorithms trained using large data sets on complex computational platforms. However, users are often expected to "blindly" trust this cognitive assistance, raising ethical concerns. On this respect, recent studies have reinforced the importance of designing transparent and explainable AI systems [2]. Moreover, the lack of interpretability or explainability eventually decreases the trust on AI systems. Opaque models, opposed to self-explainable interpretable models, mitigate these issues with the help of complementary techniques such as model explanation, outcome explanation, or model inspection [9]. Since recommendation systems interact directly with users, it is relevant to show why an item has been recommended by employing transparent and explainable mechanisms. Explainable recommendation methods have the ability to justify the output without requiring knowledge from ML algorithms [1].

Most collaborative recommendation systems use memory- or model-based filters. Memory-based algorithms predict user preferences based on similarities, making it possible to know which users originated the recommendations. k-Nearest Neighbours (k-NN) [10] is the widest used algorithm to create memory-based recommendations. Conversely, model-based algorithms build user and item latent profiles, hindering the traceability of the generated recommendations as is the case of matrix factorisation approaches. Finally, complex deep learning models have also been used to generate recommendations. In this regard, Dacrema *et al.* (2019) [7] provide a comparative analysis between standard collaborative filtering techniques and deep learning models to conclude that almost half of the tested deep learners were outperformed by k-NN. Moreover, rather than exploring offline scenarios, where static models are built and then deployed, the present contribution addresses the online or stream-based challenge, where models are incrementally updated in real time. Consequently, this work focuses on stream-based collaborative filters. To work alongside stream-based collaborative filters, we propose a local model-agnostic explanation plug-in. Explug reorders and explains recommendations based on incremental Trust & Reputation (T&R) profiling. In fact, the T&R profiles quantify the relatedness between users taking into account the set of relevant co-rated items. Hence, this paper contributes with Explug, an explanation plug-in for stream-based collaborative recommendation filters. The results highlight that Explug not only explains but also improves the recommendation quality of stream-based collaborative filters.

The rest of this paper is organised as follows. Section 2 provides a literature review. Section 3 describes the proposed method. Section 4 reports the results of the experiments. Finally, Section 5 summarises and discusses the outcomes.

2 Related Work

Filtering and ordering elements from ongoing data streams is a relevant research problem [28,16,20,3], even more within the context of social media feeds and information aggregation platforms. Typically, in such platforms, the crowdsourced feedback takes the form of ratings and reviews [4,13]. Then, online recommendation systems analyse how users interact with the vast amount of available data to suggest unseen items they might like, *i.e.*, items of interest. In this vein, collaborative filtering has been extensively used in rating-based recommendation systems [21,6]. It attempts to identify user preferences by exploiting similarity patterns between the sets of users and items. In other words, if users a and b have rated the same item identically, there is a high probability that a will be interested in items that b liked and vice versa. The most popular collaborative techniques are:

Memory-based filters revolve around the k-NN algorithm, the most widely used memory-based collaborative filter [26,11]. Neighbour-based filters build direct interaction models and apply predefined similarity functions to predict user preferences. While they are by design affected by the dimensionality curse, there are stream-based scalable and adaptive collaborative filtering implementations [28].

Model-based filters explore latent rather than direct relationships [8,17,31,24]. They dominate collaborative filtering due to dimensionality reduction, effectiveness and simple render, being broadly employed by researchers and businesses. In particular, matrix factorisation stands out among latent factor techniques due to its higher-level performance and flexibility.

Regarding model explainability, there are two major approaches: (*i*) interpretable or transparent models like linear, tree-based or rule-based predictive algorithms, which are self-explainable by design; and (*ii*) opaque or black-box models like latent factors or deep learning models. Nonetheless, according to Rudin (2019) [23], it is always preferable to adopt a self-explainable method. Whenever this is not possible, model-agnostic explanation methods are the most promising solution, given their model, explanation, and representation flexibility [19]. In the case of recommendation engines, where the most popular and recent works implement opaque methods, there is a strong case in favour of the design of post hoc model-agnostic explanation methods.

Well-known post hoc model-agnostic offline approaches include the Local Interpretable Model-agnostic Explanations (LIME) [22] and SHapley Additive exPlanations (SHAP) [18] methods. LIME creates a surrogate model around the prediction to explain, whereas Shapley values decompose the final prediction into the contribution of each attribute, which are then used to explain the prediction. To this end, Zafar & Khan (2021) [29] provide a deterministic version of LIME based on Agglomerative Hierarchical Clustering, instead of random perturbations, along with k-NN. The explanations are generated using linear interpretable models such as Decision Trees. Alternatively, Tian & Liu (2020) present a Model-Agnostic Non-linear Explanation model for Deep Learning, combining nonlinear Gradient Boosting Decision Tree with neighbour-based linear regression [25]. Moreover, Zhang et al. (2021) exploit Self-Regulated learning and Graph Neural Networks to explain black-box relational models, obtaining competing performance compared to other solutions [30].

Contrary to existing solutions, our contribution is a *post hoc local model-agnostic explanation plug-in for stream-based collaborative filters*, supported by inter-user T&R profiling. Explug explains and improves individual recommendations, overcoming the accuracy-explainability trade-off, all in real time.

3 Proposed Method

Explug, illustrated in Figure 1, was designed to work alongside stream-based collaborative filters to explain and reorder their results. It includes: (*i*) an incremental T&R profiling module based on incoming events; and (*ii*) a T&R cascade filter to reorder the outcomes and generate the explanations.

The plug-in builds the T&R profiles incrementally, receives and reorders the top N predictions generated by the collaborative recommendation filter. The cascade filter orders the predictions by value, pairwise trust between the active user and the relevant co-raters and, finally, by the reputation of relevant co-raters. The explanations are based on the incremental Trust & Reputation profiles of the relevant co-raters. Explug was evaluated by calculating the incremental Root Mean Square Error (RMSE) and/or Recall@N metrics.

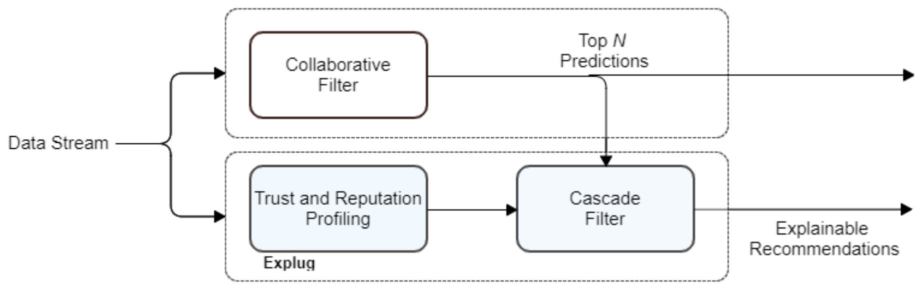

Fig. 1. Explainable plug-in for stream-based collaborative filtering.

3.1 Trust & Reputation Profiling

Trust & Reputation have been explored in several domains, including recommendation. Trust & Reputation profiling, proposed by Leal *et al.* (2019) [14], was designed and tested just for model-based collaborative filters. It analyses the behaviour of the current user and its co-raters. The profiling is based on the inter-user or pairwise trust, a direct one-to-one relationship, and reputation, a many-to-one relationship derived from the pairwise trust. Explug extends this approach, adapting the solution to other stream-based collaborative filters.

User Trust according to the active user ($T_{a,k}$) is the trust he/she has in the relevant co-rater k. It is based on the items co-rated by a and k with a rating within $\pm 10\,\%$. Equation 1 depicts $T_{a,k}$ where $I_{a,k}$ is the number of items similarly co-rated by both users, and I_a is the total items rated by a.

$$T_{a,k} = \frac{I_{a,k}}{I_a} \tag{1}$$

User Reputation (R_k) is the system-wide average trust of k. It considers a trustworthiness threshold corresponding to the average system reputation μ_R. Equation 2 depicts R_k, where $T_{c,k}$ is the trust that a co-rater c has in k,

U_k is the number of co-raters of k, and U'_k is the number of relevant co-raters of k, *i.e.*, those with $T_{c,k} \geq \mu_R$ (the average system reputation).

$$R_k = \frac{\sum_{c=1}^{U_k} T_{c,k}}{U'_k}, T_{c,k} \geq \mu_R \tag{2}$$

Item Trust according to the active user ($T_{a,i}$) is the average trustworthiness his/her relevant co-raters have on the item. Equation 3 represents $T_{a,i}$ where a is the active user, i represents the item, c a co-rater, $T_{a,c}$ the trustworthiness that a has in c, $U_{a,i}$ the number of co-raters of a who rated i, and $U'_{a,i}$ is the number of relevant co-raters of a who rated i (those with $T_{a,c} \geq \mu_R$).

$$T_{a,i} = \frac{\sum_{c=1}^{U_{a,i}} T_{a,c}}{U'_{a,i}}, T_{a,c} \geq \mu_R \tag{3}$$

Item Reputation according to the active user ($R_{a,i}$) is the average item reputation based on the relevant item co-raters. Equation 4 depicts $R_{a,i}$ where i represents the item, $U_{a,i}$ the number of co-raters of a who rated i, R_c the reputation of co-rater c, and $U'_{a,i}$ is the number of relevant co-raters of a who rated i (those with $R_c \geq \mu_R$).

$$R_{a,i} = \frac{\sum_{c=1}^{U_{a,i}} R_c}{U'_{a,i}}, R_c \geq \mu_R \tag{4}$$

Statistical Item Trust according to the active user ($T_{S a,i}$) combines the current trust the user has on the item with its statistics. It was proposed by Leal *et al.* (2021) [15] as item trust with look-back refinement. Equation 5 depicts $T_{S a,i}$, where a represents the active user, i the item, α the linearisation parameter, and $T_{a,i}$, $\mu_{T_{a,i}}$ and $\sigma_{T_{a,i}}$ the current, average and standard deviation of the incremental trust (Equation 3), respectively.

$$T_{S a,i} = \alpha T_{a,i} + (1-\alpha)|\mu_{T_{a,i}} - \sigma_{T_{a,i}}| \tag{5}$$

Statistical Item Reputation according to the active user ($R_{S a,i}$) combines the current reputation of the item from the active user perspective with its statistics. It was proposed by Leal *et al.* (2021) [15] as item reputation with look-back refinement. Equation 6 depicts $R_{S a,i}$ where α is a linearisation parameter, $R_{a,i}$, $\mu_{R_{a,i}}$ and $\sigma_{R_{a,i}}$ represent the current, average and standard deviation of the incremental reputation (Equation 4), respectively.

$$R_{S a,i} = \alpha R_{a,i} + (1-\alpha)|\mu_{R_{a,i}} - \sigma_{R_{a,i}}| \tag{6}$$

3.2 Cascade Filter

The cascade filter module reorders the top N item recommendations provided by the collaborative filter (ordered by rating) by the item Trust & Reputation (T&R) or statistical item Trust & Reputation profiles (T_s&R_s). This module calculates, according to the active user, the default item Trust & Reputation (Equation 3 and Equation 4) as well as the statistical item Trust & Reputation (Equation 5 and Equation 6). Finally, depending on the selected option, it reorders the top N recommendations by the item trust and, any resulting draws, by reputation.

3.3 Explanations

Explug generates explanations for the top N predictions of stream-based collaborative filters. With each incoming event, after updating the Trust & Reputation profiles of users and items, the plug-in explores the item Trust & Reputation profiles to produce standard explanations of the type Why?, Who?, and Which? for the reordered top N recommendations. Table 1, retrieved from Leal *et al.* (2021) [15], illustrates these explanations.

Table 1. Explug Explanations [15].

1.	Hotel Porto: 4 stars | 4.8 rating (Why?)
Why?	Porto is our top recommendation for you: it was rated 4.8 by a group of seven like-minded users (Who?) with a joint reputation $\mu_R = 13\,\%$ and in whom you trust $\mu_T = 13\,\%$
Who?	User 11 (u_{11}) rated Porto 4.8 and has a system-wide reputation of $R_{11} = 19\,\%$; Your trust in u_{11} is $T_{u,11} = 20\,\%$; In the past you chose 12 items (Which?) based on u_{11}. User 82 (u_{82}) gave Porto a 5.0 and has a reputation of $R_{82} = 16\,\%$; Your trust in u_{82} is $T_{u,82} = 18\,\%$; In the past you have chosen 12 items (Which?) based on u_{82}.
Which?	You chose in the past Hotel Vigo based on u_{11} (4.5 rating); You chose in the past Hotel Dublin based on a 4.6 rating of u_{11}; etc. You chose in the past Hotel Vigo based on u_{82} (4.7 rating); You chose in the past Hotel Dublin based on a 4.8 rating of u_{82};

3.4 Evaluation

Explug was evaluated using incremental updating and Personalised Weighted Rating Average (PWRA) profiling as proposed by Leal *et al.* (2017) [12]. The experiments were performed with the following stream-based collaborative filters: (*i*) k-NN, a memory-based collaborative filter; (*ii*) Singular Value Decomposition with Stochastic Gradient Descent (SVD-SGD), a model-based collaborative filter; and (*iii*) Random Forest (RF) regression, a tree-based collaborative filter.

After receiving the top N recommendations from the collaborative filter, Explug applies the cascade filter. In order to evaluate the experiments, we employ:

– **Incremental Root Mean Square Error** (RMSE) which incrementally measures the error between the predicted rating and real rating.
– **Incremental Recall@N**, proposed by Cremonesi *et al.* (2010) [5], which computes the classification accuracy. This metric randomly selects a sample of 1000 items never rated by the active user together with the new incoming rating done by the current user. The predictions of the selected sample are

sorted, and finally the top N are suggested to the current user. If the incoming rated item belongs to the sorted top N list, then it counts as a hit. The same process is executed for the total number of events, building the Recall incrementally. The value of N in these incremental metrics was set to 10.

4 Experiments and Results

The evaluation protocol assesses Explug working alongside collaborative filters. The implementation relies on the Python library scikit-multiflow[5], an ML package for data stream mining. The collaborative filters include an opaque (SVD-SGD) and two interpretable models (k-NN and RF regression). To quantify the impact of the proposed plug-in in terms of the incremental RMSE and Recall@10, the default filter implementation serves as the baseline method. All experiments start with empty models, *i.e.*, zero events and default pairwise trust between users of 0 %. The models are then incrementally updated with each incoming stream event. For the statistical profiles, we have used an $\alpha = 0.5$. The predictions are sorted in descending order of the predicted rating value, default or statistical item trust, and default or statistical item reputation. The experiments were conducted on a server with Ubuntu 20.04.2 LTS 64 bits with a Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30 GHz 125 GiB RAM, 40 CPU and 790 GiB of hard-disk space.

4.1 Data set

The selected data set was TripAdvisor collected by Wang *et al.* (2010) [27]. The anonymous users were removed and the data set was filtered to ensure that at least each user has rated 10 hotels and each hotel has been rated 10 times. The resulting data set contains 9114 hotels, 7453 users and 127 517 reviews. The average and standard deviation of the number of hotels rated per user are 17 ± 209 and the reviews per hotel are 14 ± 27.

4.2 Results

The results of the experiments are represented in Table 2. They compare the average incremental accuracy (RMSE) and classification accuracy (Recall@10) of the baseline methods with and without Explug. For each filter, we performed three different experiments: #1 baseline; #2 baseline plus Explug, using default item Trust & Reputation cascade filtering (T&R); and #3 baseline plus Explug, using statistical item Trust & Reputation cascade filtering (T_s&R_s). Collaborative filtering results (Table 2) show a similar impact of Explug in both opaque and transparent collaborative filters. The best results occur for all collaborative filters with the statistical Trust & Reputation profiles. The best overall result was obtained with SVD-SGD for both Recall@10 and RMSE metrics. Analysing the algorithms individually, the prediction accuracy (RMSE) remains unchanged

[5] Available at https://scikit-multiflow.github.io, January 2022.

since Explug acts *a posteriori*, *i.e.*, does not alter prediction values. The Recall@10 with $\text{T}_s\&\text{R}_s$ increases 55 % for k-NN, 105 % for RF regression, and 96 % for SVD-SGD when compared with the baseline approach, *i.e.*, without Explug. These results show the positive impact of Explug with all collaborative filters, regardless of being interpretable or opaque.

Table 2. Results with different stream-based collaborative filtering algorithms.

Algorithm	Experiment	Cascade Filter	RMSE	Recall@10
k-NN	#1	–	0.224	0.307
	#2	T&R	0.224	0.426
	#3	$\text{T}_s\&\text{R}_s$	0.224	**0.477**
RF	#1	–	0.218	0.311
	#2	T&R	0.218	0.502
	#3	$\text{T}_s\&\text{R}_s$	0.218	**0.637**
SVD-SGD	#1	–	0.165	0.334
	#2	T&R	0.165	0.456
	#3	$\text{T}_s\&\text{R}_s$	0.165	**0.655**

5 Conclusions

Collaborative filters are deeply explored to generate recommendations from user feedback. However, these recommendations are mostly opaque, leaving users with no clue about why those items were recommended. To address this problem, this paper proposes Explug – a post hoc model-agnostic explanation plug-in – to provide transparency and improve the accuracy of stream-based collaborative filters. Thus, independently of the memory-, model- and tree-based collaborative filter, Explug is able to explain and reorder recommendations.

This novel stream-based explanation plug-in explores Trust & Reputation (T&R) profiling to enhance user experience and retention. T&R profiles quantify the relatedness among users, considering the number of items rated between users with similar ratings, *i.e.*, co-rated items. Explug uses these co-rater T&R profiles to derive the T&R profiles of unknown items for the active user. Then, reorders the top N item predictions by trust, and, finally, reputation, employing the derived T&R item profiles. Therefore, recommended items can be explained in terms of the T&R of item co-raters.

As future work, we intend to mitigate the impact of malicious users in stream-based collaborative recommendation by using rating and textual information to build the Trust & Reputation profiles. This multi-source profiling approach facilitates the identification of malicious users and the automatic generation of explanations based on Natural Language Processing.

Acknowledgements

This work was partially supported by: (*i*) Xunta de Galicia grant ED481B-2021-118, Spain; and (*ii*) Portuguese National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UIDB/50014/2020.

References

1. Abdollahi, B., Nasraoui, O.: Transparency in Fair Machine Learning: the Case of Explainable Recommender Systems. Springer International Publishing, Cham (2018)
2. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence. IEEE Access 6, 52138–52160 (2018)
3. Al-Ghossein, M., Abdessalem, T., Barré, A.: A Survey on Stream-Based Recommender Systems. ACM Computing Surveys 54(5), 1–36 (2021)
4. Bilal, M., Marjani, M., Hashem, I.A.T., Gani, A., Liaqat, M., Ko, K.: Profiling and Predicting the Cumulative Helpfulness (Quality) of Crowd-Sourced Reviews. Information 10(10), 295 (2019)
5. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems. pp. 39–46. ACM (2010)
6. Cunha, T., Soares, C., de Carvalho, A.C.: Metalearning and Recommender Systems: A literature review and empirical study on the algorithm selection problem for Collaborative Filtering. Information Sciences 423, 128–144 (2018)
7. Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems. pp. 101–109. ACM (2019)
8. Dakhel, A.M., Malazi, H.T., Mahdavi, M.: A social recommender system using item asymmetric correlation. Applied Intelligence 48(3), 527–540 (2018)
9. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. 51(5) (2018)
10. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN Model-Based Approach in Classification. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 986–996. Springer (2003)
11. Gupta, M., Thakkar, A., Aashish, Gupta, V., Rathore, D.P.S.: Movie Recommender System Using Collaborative Filtering. In: Proceedings of the International Conference on Electronics and Sustainable Communication Systems. pp. 415–420. IEEE (2020)
12. Leal, F., González-Vélez, H., Malheiro, B., Burguillo, J.C.: Profiling and rating prediction from multi-criteria crowd-sourced hotel ratings. In: Proceedings of the European Conference on Modelling and Simulation. pp. 576–582. ECMS (2017)
13. Leal, F., Malheiro, B., Burguillo, J.C.: Analysis and prediction of hotel ratings from crowdsourced data. WIREs Data Mining and Knowledge Discovery 9(2), 1–9 (2019)
14. Leal, F., Malheiro, B., Burguillo, J.C.: Incremental Hotel Recommendation with Inter-guest Trust and Similarity Post-filtering. In: Proceedings of the World Conference on Information Systems and Technologies. pp. 262–272. Springer (2019)

15. Leal, F., Veloso, B., Malheiro, B., Burguillo, J.C., Chis, A.E., González-Vélez, H.: Stream-based explainable recommendations via blockchain profiling. Integrated Computer-Aided Engineering 29(1), 105–121 (2022)
16. Leal, F., Veloso, B., Malheiro, B., González-Vélez, H.: Trust and Reputation Smart Contracts for Explainable Recommendations. In: Proceedings of the World Conference on Information Systems and Technologies. pp. 124–133. Springer (2020)
17. Liu, C.L., Chen, Y.C.: Background music recommendation based on latent factors and moods. Knowledge-Based Systems 159, 158–170 (2018)
18. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 4768–4777. Curran Associates Inc. (2017)
19. Molnar, C.: Interpretable Machine Learning. Lulu.com (2020)
20. Naiseh, M., Jiang, N., Ma, J., Ali, R.: Personalising Explainable Recommendations: Literature and Conceptualisation. In: Proceedings of the World Conference on Information Systems and Technologies. pp. 518–533. Springer (2020)
21. Najafabadi, M.K., Mahrin, M.N.: A systematic literature review on the state of research and practice of collaborative filtering technique and implicit feedback. Artificial Intelligence Review 45(2), 167–201 (2016)
22. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1135–1144. Association for Computing Machinery (2016)
23. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1(5), 206–215 (2019)
24. Tahmasbi, H., Jalali, M., Shakeri, H.: TSCMF: Temporal and social collective matrix factorization model for recommender systems. Journal of Intelligent Information Systems 56(1), 169–187 (2021)
25. Tian, Y., Liu, G.: MANE: Model-Agnostic Non-linear Explanations for Deep Learning Model. In: 2020 IEEE World Congress on Services. pp. 33–36. IEEE (2020)
26. Valcarce, D., Landin, A., Parapar, J., Barreiro, Á.: Collaborative filtering embeddings for memory-based recommender systems. Engineering Applications of Artificial Intelligence 85, 347–356 (2019)
27. Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: A rating regression approach. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 783–792. ACM (2010)
28. Yagci, A.M., Aytekin, T., Gurgen, F.S.: Scalable and adaptive collaborative filtering by mining frequent item co-occurrences in a user feedback stream. Engineering Applications of Artificial Intelligence 58, 171–184 (2017)
29. Zafar, M.R., Khan, N.: Deterministic Local Interpretable Model-Agnostic Explanations for Stable Explainability. Machine Learning and Knowledge Extraction 3(3), 525–541 (2021)
30. Zhang, Y., Defazio, D., Ramesh, A.: RelEx: A Model-Agnostic Relational Model Explainer. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society. pp. 1042–1049. ACM (2021)
31. Zheng, X., Luo, Y., Sun, L., Zhang, J., Chen, F.: A tourism destination recommender system using users' sentiment and temporal dynamics. Journal of Intelligent Information Systems 51(3), 557–578 (2018)