International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Comparative Analysis of Various Algorithms of Recommender Systems for Serendipity using Novelty Scores

Tandel Saurabh[1], Rana Keyur*[2]

Submitted:10/03/2024 **Revised**: 25/04/2024 **Accepted**: 02/05/2024

Abstract: The thrust on serendipity is assisting the traditional recommender systems to narrow down on the abundance of recommendations with special weightage and emphasis on waiting-to-be-recommended 'long tail' items. Further, it also paves the way for moving from the overlooked 'accuracy' aspect of recommender systems to the highly fruitful and rightful aspect of 'user satisfaction'. As the serendipitous recommender systems inculcate the refreshing 'novelty' component, the inherent traditional recommender systems' issues of 'long tail problem', 'popularity bias', 'cold start problem', 'over specialization issue', 'matthew effect', etc. are overcome. Hence, in this paper, we investigate and analyze the effectiveness of three different serendipitous recommender system algorithms, TANGENT, KFN and an already published NOVEL SERENDIPITOUS ALGORITHM on a prominent 'novelty score' metric. The detailed and rigorous analysis suggest that all the three algorithms are able to surpass the 50 % novelty score benchmark, with the overall novelty scores of 55.57 % for the TANGENT algorithm, 79.39 % for the KFN algorithm and 83.03 % for the NOVEL SERENDIPITOUS ALGORITHM. The results vindicate the overall supremacy and efficacy of NOVEL SERENDIPITOUS ALGORITHM over the other two serendipitous algorithms.

Keywords: *Serendipity, Long Tail Items, Recommender System, TANGENT, KFN, Novelty Score, Relevance Score, Popularity Bias*

1. Introduction

Predominantly, the term recommender system (RS) refers to a software tool suggesting the items which can be interesting or appealing to the users of the RS [1]. In other words, the preliminary utility or offering of a RS is to present the user with an interesting and relevant set of filtered recommendations out of the entire catalog of products, which should be aimed at improving users' satisfaction [1-3] and larger the catalog of items, the task of finding a subset out of a plethora of offerings becomes increasingly more complex and critical [3-4].

Some authors and researchers of serendipity advocate that the Internet is limiting our horizons [1-2],[5], i.e. the personalized filters, such as Google search or Facebook delivery of news from our friends, form individual universes of information for each of us, in which we are served only with the information we already know and the information that confirms our beliefs.

To provide genuine recommendations to a user so that the suggested items or products are offering the utmost satisfaction should be given the priority while designing any recommender system. There are plenty of recommender systems available in the literature till date. But the items offered as Recommendations by the majority of the Recommender Systems do have the tendency to recommend Popular or easily identifiable or Routine items, termed as 'Popularity Bias' [6-8].

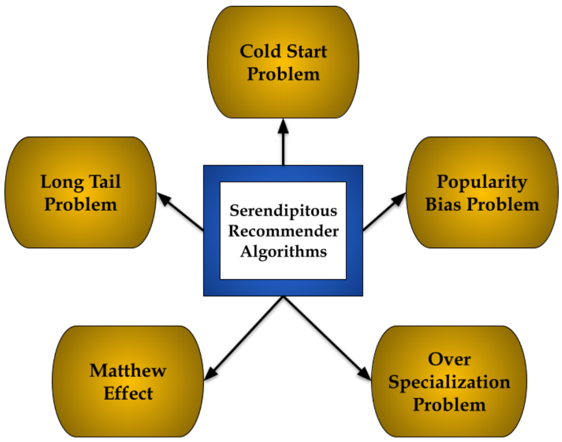

Fig. 1. Problems addressed by serendipitous recommender algorithms

Because these suggestions by the majority of the Recommender Systems lack the components of Novelty and Serendipity, such Recommender Systems end up facing the issues of 'Popularity Bias', ignorance of the 'Long Tail' [9-10] out of the less popular items and 'Matthew Effect' [11], etc. Because of such shortfalls of the Traditional Recommender Systems, the Products which are popular in the catalog have the tendency to gain even more popularity and contribute to the ever expanding lengthy list of the 'Long Tail' of Non-Popular Items, waiting to be recommended forever, leading towards the Starvation.

[1] *C K Pithawala College of Engineering and Technology, Surat, Gujarat, India & Research Scholar, Gujarat Technological University, Ahmedabad, Gujarat, India*
[2] *Sarvajanik College of Engineering and Technology, Surat, Gujarat, India*
* *Corresponding Author Email: saurabh.tandel@ckpcet.ac.in*

Moreover, there is always a possibility of introduction of New or Niche Items, termed as 'cold start items' [12-15], in the already existing catalog of Products. Now, to make such New or Niche Items forming the 'Long Tail' grab the attention of users of the system, there is always a need to improvise the system so that every product in the entire catalog gets the equitable attention and identification. Also, sometimes, the user gets bored with recommendations which are similar to their profiles, which leads to the over-specialization problem [16]. The recommender systems therefore, need to address all these issues with an introduction of serendipity as a component as can be seen in the Fig. 1 above.

2. Literature Survey

In literature, there are plenty of recommendation systems available, advocating the vital and much demanded component of serendipity. If we broadly classify these serendipitous recommender algorithms, there are two main and important categories we can derive: (1) Modified algorithms [18-20] which are updates of the already existing traditional algorithms targeting accuracy, i.e., the algorithms have good prediction capability, but are unable to suggest novel or diverse items and (2) Algorithms which have been designed not to modify or follow any previous accuracy-oriented algorithm, but they do have very different and diverse ideas of novel and unexpected recommendations [2-3],[17],[21-24].

Said et al [18] proposed the k-furthest neighbor (kFN) recommendation algorithm, similar to kNN [25]. As kNN is biased towards popular items, which results in poor personalization, kFN is designed to overcome this problem. Rather than recommending items that the neighborhood users like, kFN forms neighborhoods of users dissimilar to a target user. By selecting items dissimilar users dislike, kFN is supposed to overcome the bias of items liked by the majority of the users.

Nakatsuji et al [19] came up with a distinct idea of calculating ratings based on relatedness. Here, Relatedness is decided by utilizing random walk with restarts (RWR) on a user similarity graph. In RWR, a random particle travels from node to node with a probability equivalent to an edge weight. During each step, the particle has a probability to return to its starting node. The particle visits nodes a different number of times depending on the starting node. After a sufficient number of random walks, the ratio of the number of transitions and number of visits of a certain node stabilizes. The obtained probabilities indicate relatedness between the starting node and other nodes. In a user similarity graph, nodes correspond to users, while edges correspond to similarities. User similarities are based on a taxonomy of items. The similarities are calculated considering items and classes of items that a user has rated. By picking users who are dissimilar but related to a target user, the algorithm seems to suggest more items that are dissimilar to the target user profile.

Kawamae [20] proposed a recommendation algorithm based on estimated search time. Here, the more difficult it is to find an item, the more probable that the item is serendipitous to a user. The algorithm consists of three steps. First, for each target user the algorithm detects similar users (innovators) who have common tastes and who discover recently released items better than the target user. Second, the algorithm measures how likely it is that a target user will consume a particular item from a profile of an innovator. Third, the algorithm combines these two probability parameters into a ranking score and forms a sorted suggestion list.

Akiyama et al [21] undertook content-based filtering to inculcate serendipity in a recommender system. Initially, the algorithm puts together the items from the user's profile into clusters based on item attributes and after that, the items which are unrated by the user are assigned scores based on how distant and unexpected they are from the clusters found in the initial step. Lastly, ranks are assigned to the unrated items as per their scores and accordingly, are recommended to the target user.

Onuma et al [22] came up with the TANGENT algorithm to broaden the horizon of user tastes. The algorithm performs on a bipartite graph, where users and items correspond to nodes, while ratings correspond to edges. TANGENT detects groups of like minded users and suggests items relevant to users from different groups. For example, if a target user is a comedy fan, the algorithm will suggest a movie relevant not only to comedy fans, but also to users from other groups, such as action fans or romance fans.

Tandel et al [17] proposed a novel algorithm wherein, the users surpassing the definite number of ratings/interactions with a system are only considered, as they are called 'regular users'. After that, those items which have been rated less than a definite threshold are considered further, to generate a list of 'long tail' items. Next, the list of the nearest neighbors is found and the relevance score formula encompassing 'Bhattacharyya Coefficient' [26-28] is applied to emphasize on the 'relevance' aspect of the items to be recommended. Then, the Novelty Score metric is applied to find the novelty scores and finally, the top - n items are recommended to the target user, handling the problems ignored by the traditional accuracy-oriented algorithms.

3. Methodologies

This section is aimed at providing the details pertaining to the dataset and the novelty score metric for evaluating the overall effectiveness of various serendipitous algorithms under consideration. It also throws light on three different serendipitous algorithms, which we have used for our comparison purpose.

3.1. Dataset

For analysis purposes, we have used the MovieLens (ml-latest-small) dataset. Following are the statistics for the same [29].

Name: M ovielens-Latest-Small (ml-latest-small)

Number of Users: 610

Number of Ratings: 100836

Number of Movies: 9742

Rating Scale: 1 to 5

3.2. Algorithms

There are various categories of serendipitous recommender algorithms available in the literature as explained in Section 2 of Literature Survey, such as serendipity-oriented modification (Modification), novel algorithms (New), etc [1]. Here, we have considered three different algorithms from both the categories so that we can have an equitable comparison over an entire horizon of a variety of algorithms.

3.2.1. K Furthest Neighbor Algorithm

An example in this 'Modification' category is the k-furthest neighbor approach [18], which is a modification of the famous k-nearest neighbor algorithm (user-based Collaborative Filtering). Here, rather than suggesting items liked by users having similarity to a target user, the k-furthest neighbor algorithm recommends items disliked by users dissimilar to the target user. We have used this algorithm to compare with other benchmark algorithms of different categories. Following is the algorithmic flow (Fig. 2) to explain the working of this algorithm.

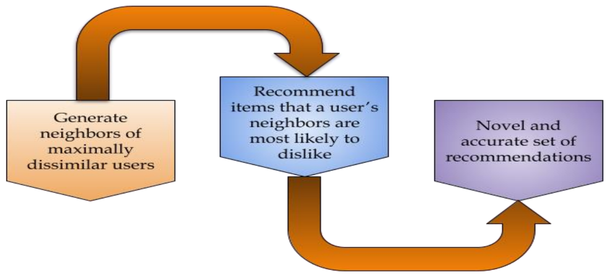

Fig. 2. Working of K-Furthest Neighbor Algorithm

3.2.2. Novel Serendipitous Recommender System (NSRS)

In this 'New' category as we have explained in 3.2, we have considered an algorithm, 'Novel Serendipitous Recommender System (NSRS)' [17], designed from scratch to cater to the non-popular items or products, which are striving to be recommended. This algorithm especially paves the way for the 'long tail' non-popular items, considering the relevance scores to yield the items of high relevance with reference to the target user. This algorithm not only recommends novel items, but also puts an equitable weightage on the relevance measure parameter to keep the non-relevant items out of the recommendation zone, while considering only 'long tail' items out of the huge catalog of the entire product portfolio, as it can be explained in the system flow in the following Fig. 3.

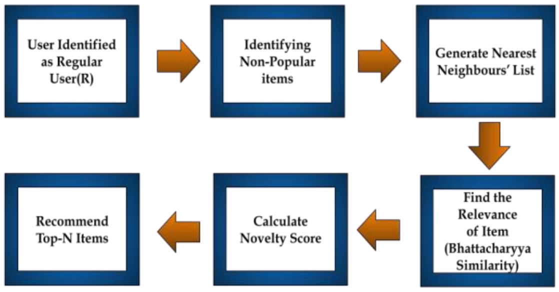

Fig. 3. Working of Novel Serendipitous Recommender System

3.2.3. Tangent Algorithm

Another example in the 'New' category is the TANGENT algorithm [22], which detects groups of like-minded users and suggests items simultaneously liked by users from the group of the target user and other groups. Recommended items are related to previous choices of the user and likely to be surprising, as these items are chosen by users from a group different than the one of the target user. Following diagram (Fig. 4) explains how the Tangent algorithm suggests the recommendations in a different way as compared to the traditional recommender algorithms.

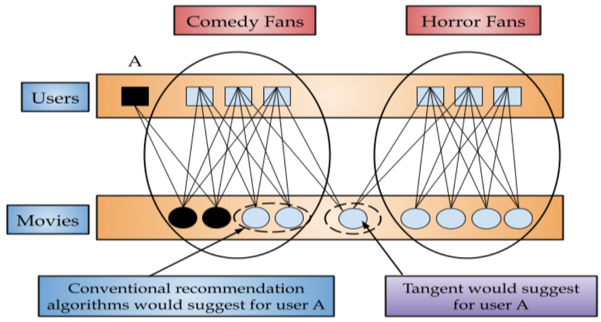

Fig. 4. Working of Tangent Algorithm

3.3. Overall System Flow

The following figure (Fig. 5) depicts the overall flow of the proposed system for the comparison of various serendipitous algorithms. It commences with the selection of an appropriate dataset, i.e., Movielens-Latest-Small (ml-latest-small). In the next step, it undertakes various serendipitous algorithms from different categories of algorithms as explained in section 3.2. After that, the system performs the computation of novelty scores of all the items under consideration and then recommends the top-10 novelty scores along with the respective novelty values. In the final step, we have undertaken the explanation of results obtained through the proposed algorithm in terms of analysis.

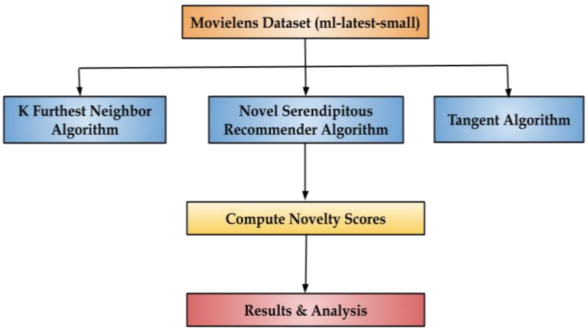

Fig. 5. Working of the overall proposed system

3.4. Novelty Score Metric

The Novelty Score Metric [1] for finding the novelty scores of the movies generated from three different algorithms to cope up with the 'Popularity Bias' and overcoming the issue of 'Long Tail' is as follows:

$$Nov_d(i,u) = \frac{1}{|I_u|} \sum_{j \in I_u} dist(i,j)$$

(1)

Here,

$Nov_d(i,u)$ = Novelty Score of movie 'i' for user 'u'

I_u = List of all movies user 'u' has rated

dist(i,j) = Distance between movie 'i' and movie 'j' = 1 - sim(i,j)

sim(i, j) is any kind of similarity(Cosine/Jaccard/Pearson) between movies i and j & sim(i, j) ∈ [0, 1]

This novelty score metric equation will give us a measure to evaluate how novel the movies are for the target user, presenting the testimony to judge the effectiveness of the algorithms under consideration.

4. Results and Analysis

4.1. Novelty Score Analysis

4.1.1. Distribution of Novelty Scores

The Distribution of Novelty Scores has been explained using two different ways: (1) using the tables of statistics to exhibit the average novelty scores for each of the three algorithms, along with the range of the values of novelty scores and (2) using the histogram chart.

4.1.1.1. K Furthest Neighbor Algorithm

The average novelty score for the kfn algorithm is 0.7939, i.e. 79.39 % and the novelty score values for the algorithm range from 41.20 % to 96 % as can be seen in Table 1 and Fig. 6 below.

Table 1. Novelty Score Statistics for the K Furthest Neighbor Algorithm

No of Users	Average Novelty Score	Highest Novelty Score	Lowest Novelty Score
610	0.7939	0.960	0.412

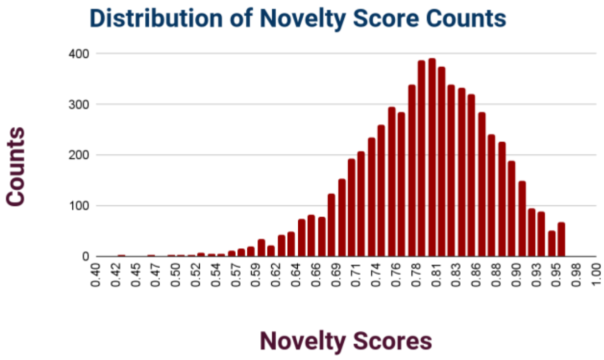

Fig. 6. Novelty Score distribution for the K Furthest Neighbor Algorithm

4.1.1.2. Novel Serendipitous Recommender Algorithm (NSRS)

The average novelty score for the novel serendipitous recommender algorithm is 0.8303, i.e. 83.03 % and the novelty score values for this algorithm range from 47.50 % to 100 % as can be observed in Table 2 and Fig. 7 below.

Table 2. Novelty Score Statistics for the Novel Serendipitous Recommender Algorithm (NSRS)

No of Users	Average Novelty Score	Highest Novelty Score	Lowest Novelty Score
243	0.8303	1.000	0.475

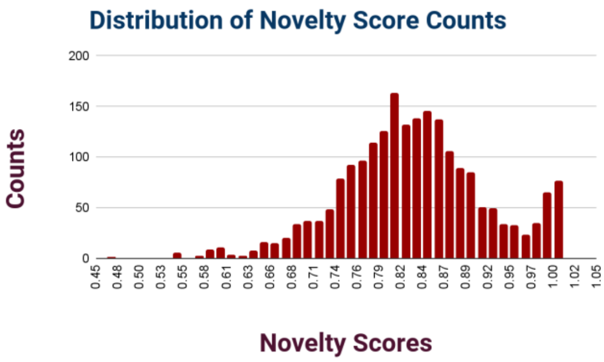

Fig. 7. Novelty Score distribution for the Novel Serendipitous Recommender Algorithm (NSRS)

4.1.1.3. Tangent Algorithm

The average novelty score for the novel serendipitous recommender algorithm is 0.5557, i.e. 55.57 % and the novelty score values for this algorithm range from 24.20 % to 87.70 % as can be seen in Table 3 and Fig. 8 below.

Table 3. Novelty Score Statistics for the Tangent Algorithm

No of Users	Average Novelty Score	Highest Novelty Score	Lowest Novelty Score
610	0.5557	0.877	0.242

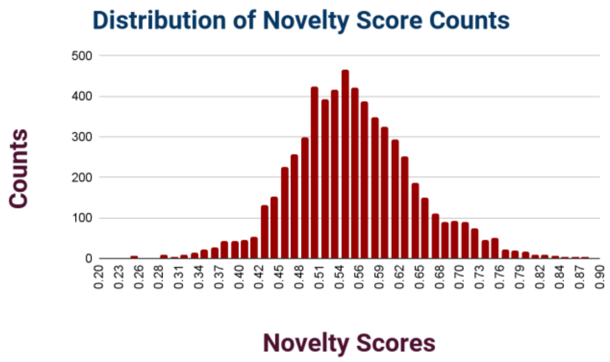

Fig. 8. Novelty Score distribution for the Tangent Algorithm

These results of novelty score values prove that all the algorithms are able to recommend items which are novel, as the average novelty score of 55.57 % which is the lowest, is achieved by the tangent algorithm; highly novel items are recommended by the kfn algorithm with an average novelty score value of 79.39 % and the novel serendipitous recommender algorithm proves to be the best algorithm to suggest the average novelty score value of 83.03 %.

So, this is a testimony of the fact that the novel serendipitous recommender algorithm outperforms the remaining two algorithms to recommend the items with the highest novelty values from the big catalog of items.

4.1.2. Novelty Scores Vs. User IDs

The next three graphs in Fig. 9 (Kfn), Fig. 10 (NSRS) and Fig. 11 (Tangent) have been depicted to show the average novelty score values for all the three different algorithms.

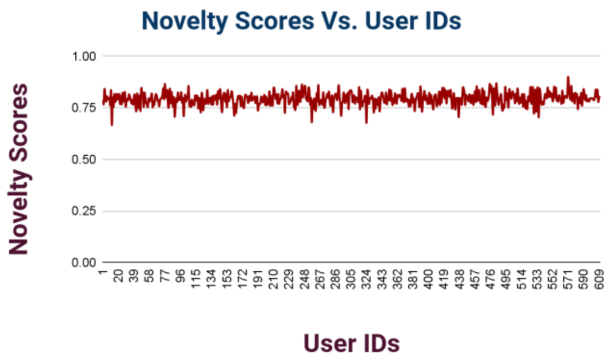

Fig. 9. Novelty Scores for the K Furthest Neighbor Algorithm

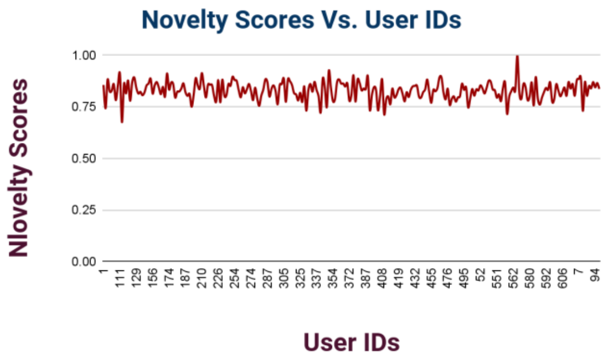

Fig. 10. Novelty Scores for the Novel Serendipitous Recommender Algorithm

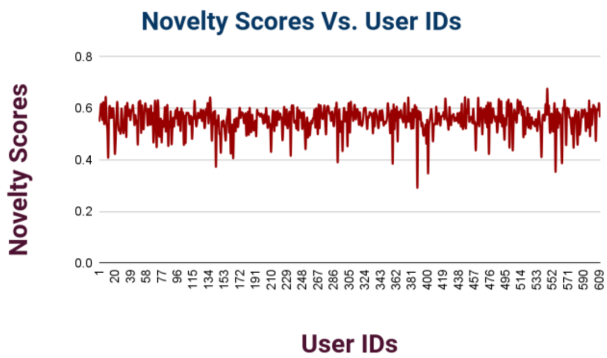

Fig. 11. Novelty Scores for the Tangent Algorithm

Here, the above three graphs show how different User IDs are having their novelty score values revolving around the average novelty score values for all the three algorithms.

4.1.3. Novelty Score Plots

The next three graphs, i.e., Fig. 12 (Kfn), Fig. 13 (NSRS) & Fig. 14 (Tangent), are the scatter charts exhibiting how the density of novelty score values have been scattered and around which value of novelty score, there is a high density of plotted points, for different users in the system.

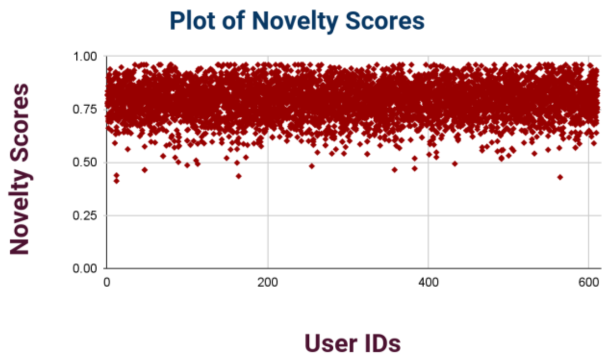

Fig. 12. Novelty Score points for the K Furthest Neighbor Algorithm

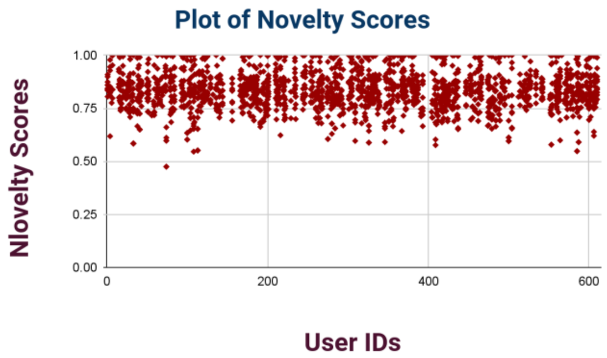

Fig. 13. Novelty Score points for the Novel Serendipitous Recommender Algorithm

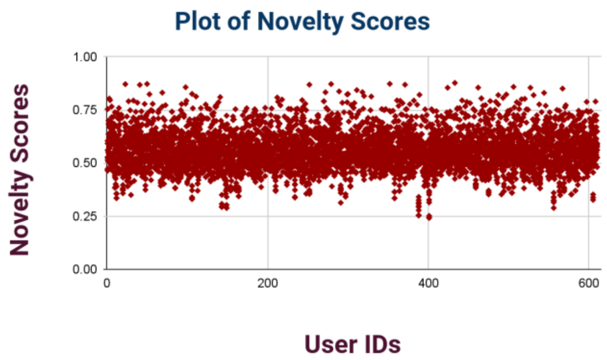

Fig. 14. Novelty Score points for the Tangent Algorithm

4.2. Comparison of average novelty scores

Here, in the following graph, i.e., Fig. 15, we can confirm the supremacy of the novel serendipitous recommender algorithm over the other two algorithms while bringing the large corpus of non-popular items into the recommendation territory.

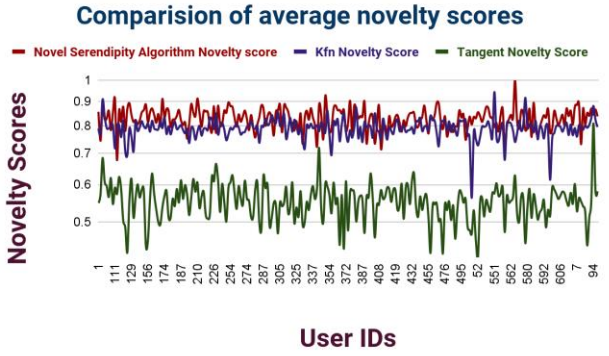

Fig. 15. Overall comparison of all the three algorithms for the average novelty scores

5. Conclusions

Primarily, all three serendipitous algorithms are extremely good at tackling the problem areas of 'long tail', 'popularity bias', 'over specialization', 'cold start', 'matthew effect', etc. as they put an emphasis on considerably long tail of waiting-to-recommend item corpus. Bringing the large pool of 'long tail' items is an increasingly important and crucial task as it puts a plethora of such items into reckoning and that can prove to be the starting point for many more such striving items to get an all important recommendation and recognition.

In this paper, we have comparatively evaluated three different serendipitous algorithms on a vital and important metric of novelty score. The experimental results and analysis prove that all the three serendipitous algorithms are capable of recommending items of vast horizons.

The K furthest neighbor algorithm, with an overall average novelty score of 79.39 % is quite good to recommend a diverse and different set of items. The Tangent algorithm, with an overall average novelty score of 55.57 % is also achieving satisfactory results to suggest and recommend novel items. But, the Novel serendipitous recommender algorithm, with an overall average novelty score of 83.03 % is exceedingly efficient to achieve the highest novelty score and brings together the large plethora of 'long tail' items into reckoning, which proves the supremacy of the Novel serendipitous recommender algorithm.

References

[1] D. Kotkov, S. Wang, and J. Veijalainen, "A survey of serendipity in recommender systems," *Knowledge-based Systems,* vol. 111, pp. 180–192, Nov. 2016

[2] M. Ge, C. Delgado-Battenfeld, and D. Jannach, 2010, September. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems* (pp. 257-260).

[3] R. J. Ziarani and R. Ravanmehr, "Serendipity in Recommender Systems: A Systematic Literature

review," *Journal of Computer Science and Technology/Journal of Computer Science and Technology*, vol. 36, no. 2, pp. 375–396, Mar. 2021.

[4] D. Kotkov, J. Veijalainen, and S. Wang, "Challenges of serendipity in recommender Systems," *International Conference on Web Information Systems and Technologies*.

[5] M. De Gemmis, P. Lops, G. Semeraro, and C. Musto, "An investigation on the serendipity problem in recommender systems," *Information Processing & Management*, vol. 51, no. 5, pp. 695–717, Sep. 2015.

[6] H. Abdollahpouri, R. Burke, and B. Mobasher, "Managing Popularity Bias in Recommender Systems with Personalized Re-Ranking.," *arXiv (Cornell University)*, pp. 413–418, Jan. 2019, [Online].

[7] H. Abdollahpouri, M. Mansoury, R. Burke, and B. Mobasher, "The Unfairness of Popularity Bias in Recommendation.," *arXivLabs*, Jan. 2019, [Online].

[8] H. Abdollahpouri, M. Mansoury, R. Burke, and B. Mobasher, "User-centered evaluation of popularity bias in recommender systems," *29th ACM Conference on User Modeling, Adaptation and Personalization*.

[9] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen, "Challenging the long tail recommendation," *Proceedings of the VLDB Endowment*, vol. 5, no. 9, pp. 896–907, May 2012.

[10] Y.-J. Park, and A. Tuzhilin, (2008) "The Long Tail of recommender systems and how to leverage it", *Proceedings of the 2008 ACM conference on Recommender systems* [Preprint].

[11] H. Wang, Z. Wang and W. Zhang, (2018) "Quantitative analysis of Matthew effect and sparsity problem of recommender systems," *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)* [Preprint].

[12] J. Gope, and S.K. Jain, (2017) 'A survey on solving cold start problem in Recommender Systems', *2017 International Conference on Computing, Communication and Automation (ICCCA)* [Preprint].

[13] X.N. Lam, T Vu, T.D. Le, and A.D. Duonget, (2008) 'Addressing cold-start problem in recommendation systems', *Proceedings of the 2nd international conference on Ubiquitous information management and communication* [Preprint].

[14] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, (2014) 'Facing the cold start problem in Recommender Systems', *Expert Systems with Applications*, 41(4), pp. 2065–2073.

[15] Z.-K. Zhang, C. Liu, Y.C. Zhang, and T. Zhou, (2010) 'Solving the cold-start problem in recommender systems with social tags', *EPL (Europhysics Letters)*, 92(2), p. 28002.

[16] O. Stitini, S. Kaloun, and O. Bencharef, (2022) 'An improved recommender system solution to mitigate the over-specialization problem using genetic algorithms', *Electronics*, 11(2), p. 242.

[17] S. Tandel, and K. Rana, (2023) 'Novel Serendipitous Recommender System using Relevance Scores for Long Tail Items', *International Journal ofINTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING*, 12(1), pp. 676–682.

[18] A. Said, B. Fields, B.J. Jain, and S.A. Albayrak, (2013) 'User-centric evaluation of a K-furthest neighbor collaborative filtering recommender algorithm', *Proceedings of the 2013 conference on Computer supported cooperative work* [Preprint].

[19] M. Nakatsuji, Y. Fujiwara, A. Tanaka,T. Uchiyama, K. Fujimura, T. and Ishida, (2010) 'Classical music for rock fans? Novel recommendations for expanding user interests', *Proceedings of the 19th ACM international conference on Information and knowledge management* [Preprint].

[20] N. Kawamae, (2010) 'Serendipitous recommendations via Innovators', *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* [Preprint].

[21] T. Akiyama, K. Obara, and M. Tanizaki, 2010, September. Proposal and Evaluation of Serendipitous Recommendation Method Using General Unexpectedness. In *PRSAT@ RecSys* (pp. 3-10).

[22] K. Onuma, H. Tong, and C. Faloutsos, 2009, June. Tangent: a novel,'surprise me', recommendation algorithm. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 657-666).

[23] D. Kotkov, J.A. Konstan, Q. Zhao, and J. Veijalainen, 2018, April. Investigating serendipity in recommender systems based on real user feedback. In *Proceedings of the 33rd annual acm symposium on applied computing* (pp. 1341-1350).

[24] K. Kaminskas, and D. Bridge, 2016. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *7*(1), pp.1-42.

[25] A. Moldagulova, and R.B. Sulaiman, 2017, May. Using KNN algorithm for classification of textual documents. In *2017 8th international conference on information technology (ICIT)* (pp. 665-671). IEEE.

[26] K.G. Derpanis, 2008. The bhattacharyya measure. *Mendeley Computer*, *1*(4), pp.1990-1992.

[27] X. Guorong, C. Peiqi, and W. Minhui, 1996, August. Bhattacharyya distance feature selection. In *Proceedings of 13th International Conference on Pattern Recognition* (Vol. 2, pp. 195-199). IEEE.

[28] H. Cao, J. Deng, J., H. Guo, B. He, and Y. Wang, 2016, September. An improved recommendation algorithm based on Bhattacharyya Coefficient. In *2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA)* (pp. 241-244). IEEE.

[29] "MovieLens 100k Dataset | GroupLens," Available: https://grouplens.org/datasets/movielens/. [Accessed: Jan. 03, 2024]