Check for updates

KNN-Based Collaborative Filtering for Fine-Grained Intelligent Grad-School Recommendation System

Jinfeng Xu[1], Jiyi Liu[2], Zixiao Ma[3], Yuyang Wang[4], Wei Wang[5], and Edith Ngai[1(✉)]

[1] Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong 999077, China
chngai@eee.hku.hk

[2] School of Software Engineering, Sun Yat-Sen University, Zhuhai 528406, Guangdong, China
liujy557@mail2.sysu.edu.cn

[3] Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA
zixiao.ma@duke.edu

[4] Department of Computer Science, Cornell University, Ithaca, NY 14853, USA

[5] Guangdong-Hong Kong-Macao Joint Laboratory for Emotional Intelligence and Pervasive Computing, Artificial Intelligence Research Institute, Shenzhen MSU-BITUniversity, Shenzhen 518172, Guangdong, China
ehomewang@ieee.org

Abstract. The development of the Internet has led to information overload, and how to filter and sift information is a rigorous requirement in all fields. In response to this challenge, recommendation systems have emerged as a valuable tool, offering personalized content and services by efficiently searching and processing dynamically generated information. For students applying to grad schools, finding relevant information can be time-consuming and unreliable from official websites or forums. In light of these challenges, we present a novel solution in the form of an application recommendation platform. Our proposed platform leverages specific open-source datasets and real-time information from platform users using KNN (K-Nearest Neighbor) and CF (Collaborative Filtering) techniques to provide recommendations based on users' individual backgrounds, we aim to reduce the complexity inherent in information retrieval while simultaneously enhancing the relevance of the recommendations delivered to users. Specifically, we first collect user behavior data, then we will construct the data model and perform some preprocessing on it. Calculate the user similarity, and find out the K-nearest neighbors and rate based on K-nearest neighbors, finally, the recommendation engine is used to calculate the highest-rated items to be recommended to the users.

Keywords: Recommendation System · Collaborative Filtering · K-Nearest Neighbor

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2024
Published by Springer Nature Switzerland AG 2024. All Rights Reserved
V. C. M. Leung et al. (Eds.): Qshine 2023, LNICST 573, pp. 494–508, 2024.
https://doi.org/10.1007/978-3-031-65126-7_41

1 Introduction

The Internet's rapid growth has led to an overwhelming amount of information, causing information overload [11,27] for users. To address this, two main approaches are used: controlling information generation and filtering information access. However, controlling information generation has become ineffective due to the Internet's speed of development [8]. Thus, there is an urgent need for robust filtering mechanisms to prioritize relevant content and enable efficient communication to tackle information overload [14]. Dynamic faceted filters have shown promise in alleviating information overload in previous studies [22]. Additionally, recommendation systems that filter based on specific requirements have demonstrated their effectiveness in commercial applications [14,31].

Presently, college students encounter difficulties accessing and screening information about graduate programs. The available channels, such as forums and official school websites, are inefficient and require manual screening and comparison. Recommendation systems offer a promising solution to this challenge, providing efficient and accurate information filtering, thus reducing the time and effort required by students [28].

Our recommendation system employs KNN-based Collaborative Filtering (CF) to create personalized recommendation lists for users with similar backgrounds. CF is a highly effective and popular algorithm for recommendation systems, known for its robustness and efficiency [16]. By using KNN in Collaborative Filtering, we address potential personalization issues and generate more reliable recommendation lists by considering multiple similar cases together [5,23]. To handle large volumes of data, we deploy KNN-based CF on Hadoop, significantly improving the system's performance for handling substantial data.

This paper proposes an intelligent fine-grained recommendation system for grad-school application, the contributions are:

- Tanimoto Coefficient Similarity is used in user similarity calculation to focus the similarity on the correlation relationship between users and items, and reduce the focus on specific ratings.
- A recommendation system that focuses on both users' features and interest preferences is proposed, which is more suitable for graduate school recommendation scenarios than the past school recommendation system that only focuses on interest preferences.
- The experimental results verify that the recommendation system focuses on user features while also playing a sizable role in the recommendation of interest preferences.

Section 2 presents the related work on the recommendation systems, as well as the fundamental principles of KNN (K-Nearest Neighbor) and CF (Collaborative Filtering). In Sect. 3, we present our framework, which comprises essential stages such as Information Collection, Model Construction, Similarity Calculation, k-Neighbors Identification, Recommendation Engine Development, and Performance Enhancement using the Hadoop Cluster. The outcomes and findings of our approach are outlined in Sect. 4. Finally, in Sect. 5, we provide a concise summary and conclusion, highlighting the key contributions and implications of the research presented in this paper.

2 Related Works

This section is dedicated to a comprehensive exploration of existing recommendation systems, where we analyze various implementation methods and algorithms in the context of their applicability (Sect. 2.1). Subsequently, we delve into the details of KNN (K-Nearest Neighbor) and its relevance and suitability for recommendation systems (Sect. 2.2). Moving forward, we examine the distinctive characteristics of collaborative filtering and assess the feasibility of incorporating KNN-based collaborative filtering (Sect. 2.3). Finally, we provide a succinct overview of the current landscape of school recommendation systems (Sect. 2.4).

2.1 Technical Options for Recommendation Systems

The utilization of efficient and precise recommendation techniques holds paramount significance for a system aiming to deliver valuable and relevant recommendations to its individual users. Figure 1 shows the anatomy of different recommendation filtering techniques.

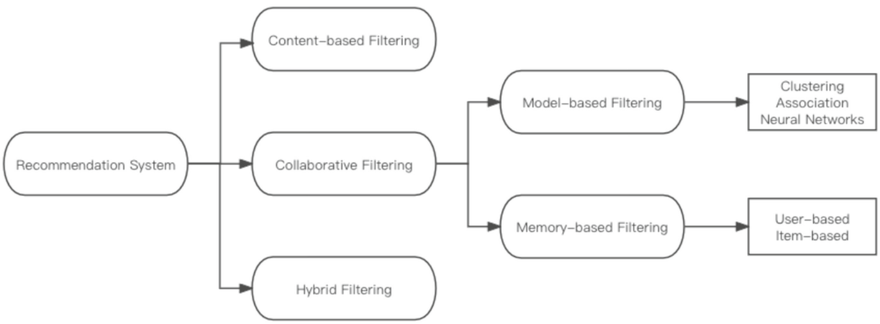

Fig. 1. Recommendation filtering techniques

In general, recommendation systems are categorized according to the methods they use. These methods can be categorized into four main groups [14,21]. The different types of recommendation approaches or methods are briefly discussed below:

- Collaborative Filtering (CF)
  CF refers to recommending items to target users by identifying users with similar interests. This method is designed to help users get appropriate recommendations through individuals or groups with the same preferences or behaviors.
- Content-Based (CB)
  The content-based method recommends products to users using their historical data. It analyzes the user's past searches and purchases to suggest related items. The method heavily relies on user ratings, making it especially valuable in business, information, and education domains [26,37].
- Knowledge-based
  Recommendation methods are employed to assist users in making informed decisions while purchasing complex items with various attributes. Users often seek items with specific features like car models, engine types, or house interior designs. In certain business contexts, finding ratings for recommendations is challenging due to expensive items and low purchase demand. These methods are particularly useful in cold start situations where traditional rating-based approaches may not be feasible [32].
- Hybrid
  These approaches bring together the advantages of different types of recommendation systems. The aim is to create recommendation systems using techniques that are more efficient and effective in terms of performance [3,9].

2.2 KNN Are Suitable for Recommendation System

The k-Nearest-Neighbours (kNN) is a non-parametric classification method, which is simple but effective in many cases [12].

KNN stands out as one of the most effective neighboring algorithms. Given its proven success, the KNN algorithm finds widespread application in numerous recommendation systems, particularly for computing user similarities [1,3]. However, the KNN algorithm also has many drawbacks. Therefore, many special KNNs for different circumstances have been proposed. Such as Adaptive KNN [30], Improved KNN [18], and A hybrid action-related KNN [25].

2.3 The Characteristics of Collaborative Filtering

Collaborative filtering (CF) is a highly successful method for building recommendation systems. It uses known user preferences to predict unknown preferences [29]. CF can be categorized into three types: Memory-based, Model-based, and Hybrid (see Fig. 2). Memory-based CF calculates user similarities from interaction data to make recommendations. Model-based CF uses machine learning algorithms to build predictive models. Hybrid CF combines both memory-based and model-based techniques for improved and accurate recommendations [29].

CF categories	Representative techniques	Main advantages	Main shortcomings
Memory-based CF	*Neighbor-based CF (item-based/user-based CF algorithms with Pearson/vector cosine correlation) *Item-based/user-based top-*N* recommendations	*easy implementation *new data can be added easily and incrementally *need not consider the content of the items being recommended *scale well with co-rated items	*are dependent on human ratings *performance decrease when data are sparse *cannot recommend for new users and items *have limited scalability for large datasets
Model-based CF	*Bayesian belief nets CF *clustering CF *MDP-based CF *latent semantic CF *sparse factor analysis *CF using dimensionality reduction techniques, for example, *SVD*, *PCA*	*better address the sparsity, scalability and other problems *improve prediction performance *give an intuitive rationale for recommendations	*expensive model-building *have trade-off between prediction performance and scalability *lose useful information for dimensionality reduction techniques
Hybrid recommenders	*content-based CF recommender, for example, *Fab* *content-boosted CF *hybrid CF combining memory-based and model-based *CF* algorithms, for example, Personality Diagnosis	*overcome limitations of CF and content-based or other recommenders *improve prediction performance *overcome CF problems such as sparsity and gray sheep	*have increased complexity and expense for implementation *need external information that usually not available

Fig. 2. Overview of collaborative filtering techniques [29].

The usual memory-based collaborative filtering techniques can be separated into item-based, and user-based methods.

- item-based
  The method calculates predictions based on product/item similarity rather than user similarity [4].
- user-based
  The method predicts user behavior by using a weighted sum based on average ratings of users who rated the item in the past and the average user rating.

KNN models are highly accurate and widely used in collaborative filtering (CF) recommendation systems. They have been popular since they were introduced and are known for providing reasonable explanations for their recommendations. Experiments have shown that CF with KNN-based methods significantly reduces error rates [30].

2.4 Overview of Existing School Recommendation Systems

Previous studies on school recommendation systems have utilized cosine similarity and TF-IDF vectorization for matching [28], as well as KNN and Support Vector Machine for filtering decisions [7]. However, these former model may bias recommendations towards popular schools, which might not be the best fit for all students. Moreover, updating the latter model can be computationally expensive.

In contrast, our recommendation system employs KNN-based collaborative filtering, prioritizing successful admission results of students with similar backgrounds. To handle large data volumes efficiently, we utilize Hadoop for improved performance.

3 Framework

In this section, we provide a comprehensive overview of our core recommendation system, presenting its structure, workflow, and underlying motivation. Additionally, we conduct a thorough performance analysis of the recommendation system, evaluating its Applicability, Stability, Efficiency, and Deployment cost (Sect. 3.1). Next, we introduce the overall architecture of our website, and some work other than the recommendation system (Sect. 3.2). Lastly, we present our implementation scheme for KNN-based Collaborative Filtering (CF) utilizing Hadoop to address the challenges posed by large data volumes and enhance system performance (Sect. 3.3).

3.1 Recommendation System Architecture

We choose user-based collaborative filtering [38] to construct a data model based on the user's important background information, use Tanimoto Coefficient Similarity [10,33] (as Eq. 1) to construct a similarity model by calculating the similarity based on the background information, and find out K-Nearest-Neighbors to construct a similarity model. Finally, all rated items in the neighborhood are prioritized by the recommendation engine [6,39] and recommended to the user. The selection of these technologies is guided by factors such as applicability, stability, efficiency, and deployment cost, ensuring the effectiveness and practicality of our recommendation system for real-world applications.

$$s(i,j) = \frac{n(c_i \cap c_j)}{n(c_i \cup c_j)} = \frac{n(c_i \cap c_j)}{n(c_i) + n(c_j) - n(c_i \cap c_j)} \tag{1}$$

Applicability. In our specific situation, user-based collaborative filtering is the suitable choice, matching user preferences with recommendations from similar situations. KNN-based similarity modeling improves accuracy, leading to precise recommendations. Our system prioritizes schools where successful application is more likely for the user, and we compute similarity using the Tanimoto coefficient [19,34].

Stability. Stability is a crucial factor to consider. Using KNN for similarity modeling allows effective filtering of abnormal information based on the most similar users. However, the presence of inauthentic data can impact the accuracy of KNN-based similarity calculations, affecting the overall precision and reliability of our recommendation system [2].

Efficiency. Efficient data processing is crucial in our recommendation system. While the utilization of Tanimoto Coefficient Similarity narrows down the results to binary form, thus enhancing efficiency to some extent, collaborative filtering efficiency still faces challenges, especially in scenarios involving large datasets. To address this, we adopt Hadoop for distributed processing, implementing a collaborative filtering recommendation algorithm on clustering. This approach significantly enhances the system's efficiency, allowing us to handle large volumes of data more effectively, improving overall performance and scalability [35,36].

Deployment Cost. The KNN algorithm offers two key advantages in our recommendation system. Firstly, it enables fast training, making processing of large datasets efficient. Secondly, its flexibility allows easy expansion of the training set. When users update their backgrounds and decisions, our recommendation model can be promptly retrained with the updated data. This dynamic training process ensures that our system remains up-to-date and responsive to users' changing preferences and needs. By leveraging the speed and adaptability of KNN, we create a recommendation platform that can accommodate a growing user base and continuously improve its performance over time.

The specific workflow of the recommendation system is shown in Fig. 3.

Step (1) user uploads personal background information to our website, Step (2) constructs a data model based on the background information, Step (3) calculates the similarity using Tanimoto Coefficient Similarity, finds K-Nearest-Neighbor users and constructs a similarity model through Step (4). Step (5) Calculate the score based on the weights of different items of all N neighboring users by recommendation engine. Steps (6)–(7) select M highest rated items to recommend to the user.

3.2 Website Architecture

Our website offers more than just a Recommendation System for master program applications. Users can take virtual campus tours using 3D Rendering and Google Earth API. They can compare multiple programs and universities side by side. Additionally, the platform provides resources like application tips, financial aid information, and career prospects. Users can communicate with university representatives and current students through our Instant Messaging (IM) service.

To enhance user experience, we focus on High Availability, Robustness, and user-friendly design. We employ advanced front-end techniques like Vue.js, Bootstrap, and Element UI. Our website is optimized for speed using caching, Content Delivery Network (CDN), and static file compression. We perform rigorous testing to catch and resolve bugs and regularly monitor the website's performance using Portainer and phpMyAdmin. We support Open Authorization (OAuth 2.0) [13] and OIDC [20] for easy login using third-party identity providers like Google, Github, and WeChat, streamlining the authentication process.

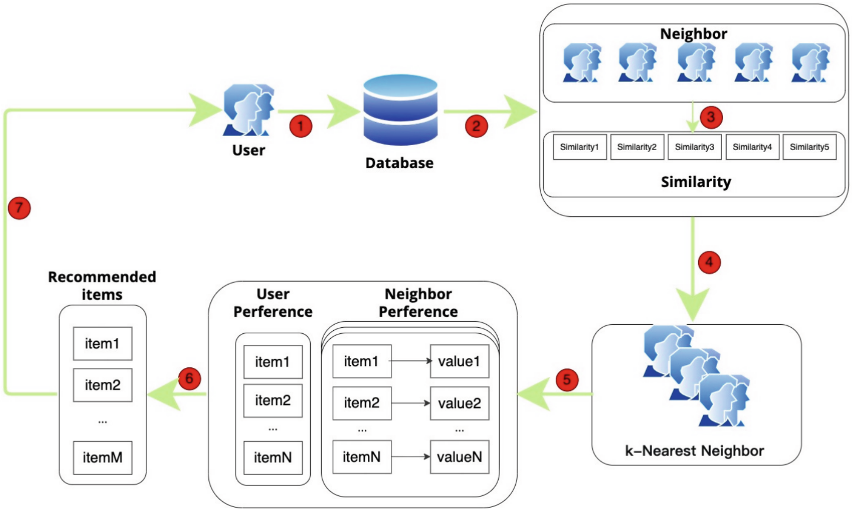

Fig. 3. Recommendation system workflow

3.3 How to Improve CF Performance with Hadoop

The Mahout library [24] provides implementations of paralleled versions of algorithms in the field of machine learning for the Hadoop platform. It focuses on classification, grouping, and collaborative filtering algorithms. The Mahout library was used for creating a recommendation system based on the Apache Hadoop technology. In the Mahout library, the two most important programs which realize the paralleled CF algorithm based on items are RecommenderJob (which calculates recommendations) and ItemSimilarityJob (which calculates the similarity matrix (Eq. 2)) [38]. A full implementation of the paralleled version of the collaborative filtering algorithm based on items, according to the MapReduce paradigm [15] (Fig. 4), is realized in the form of nine consecutive jobs [17].

$$\begin{bmatrix} a\ b \\ c\ d \end{bmatrix}_{ItemSim.} * \begin{bmatrix} x \\ y \end{bmatrix}_{UserPrefs.} = \begin{bmatrix} xa\ yb \\ xc\ yd \end{bmatrix}_{UserRecs.} \quad (2)$$

4 Result

Our platform offers a highly personalized and comprehensive experience, providing users with their top 10 best-matched colleges and programs. Unlike other sites that rely solely on exam scores, we consider multiple selection parameters for our college recommendations. Collaborative Filtering (CF) is utilized to curate recommendations based on success stories from users with similar backgrounds. Additionally, we enhance the user experience with 3D rendering and Google Earth APIs, allowing virtual campus tours. Our platform also offers paperwork

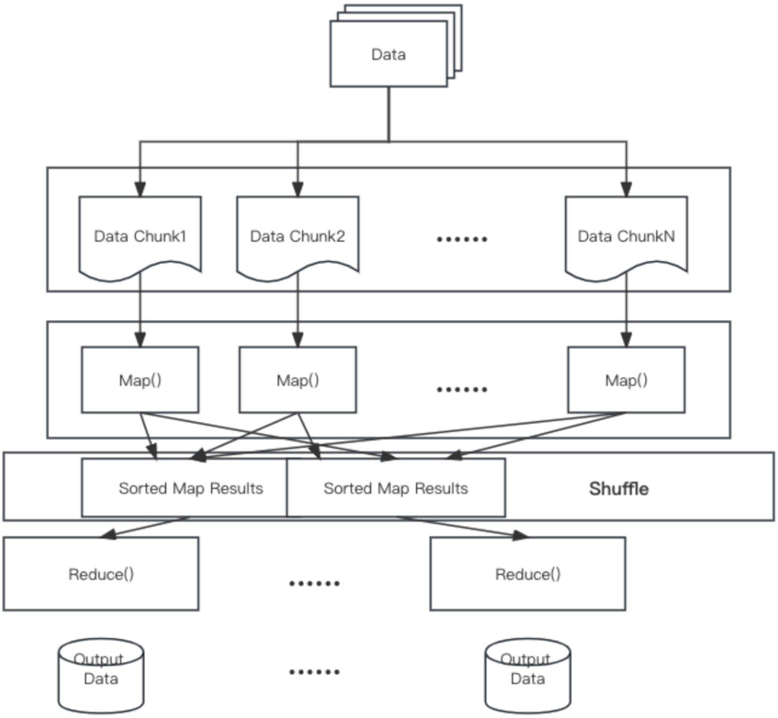

Fig. 4. The MapReduce model

revisions and a user forum for communication and sharing. We aim to revolutionize the college application process by providing comprehensive support and resources for informed decision-making and academic success.

Personal Information. Users can edit their profiles including avatar, username, personal information, password, etc. (1)–(3). They can also upload or update their backgrounds to guide the recommendation system in generating matching recommendations (4) (Fig. 5).

Recommendation System. Figure 6 illustrates the functionality of our recommendation system. Leveraging the user's basic information as input, our system employs KNN-based Collaborative Filtering to generate a personalized recommendation list.

To enhance the user experience and facilitate informed decision-making, our website provides detailed views of recommended programs, offering comprehensive information about the school and specific programs. We also integrate advanced virtual 3D campus technology for virtual campus exploration. Furthermore, our platform includes various functions and resources to enrich the user experience and assist applicants in their college selection process.

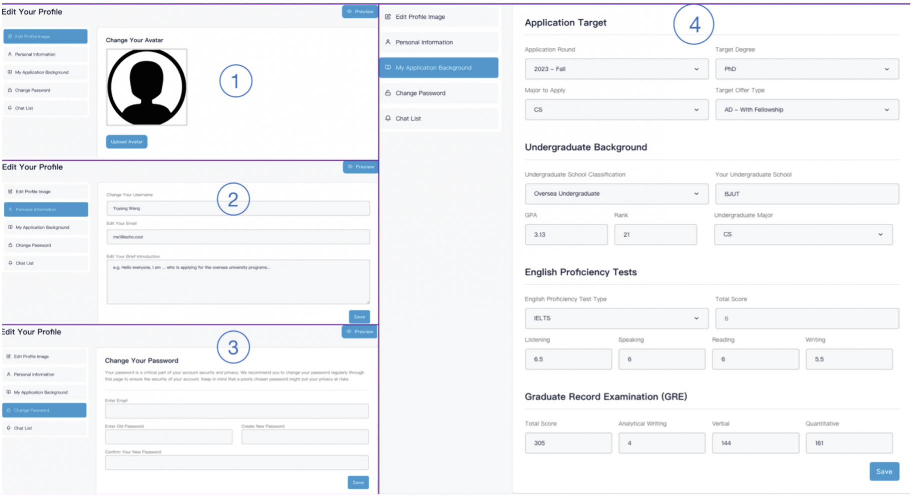

Fig. 5. Personal Information

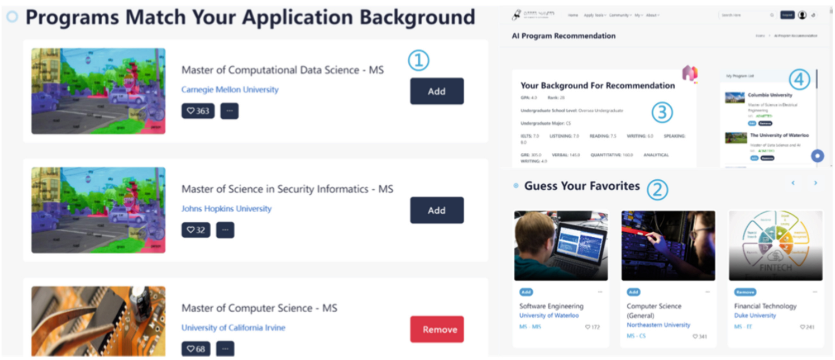

Fig. 6. Recommendation System

Report Application. (Figure 7) Users can upload and change their application status. Only applications with offers will be recorded in our database. However, the approach to guarantee the authenticity of users' data is still a problem that needs to be solved.

Comparison. Figure 8 directs users to compare two different programs selected from their programs list, which display the basic information and average background about this pair of programs.

By incorporating this comparative functionality, users gain deeper insights into the programs of interest, enabling them to make more informed decisions beyond solely relying on the recommendations provided by the system. This interactive and user-driven comparison process empowers applicants to assess

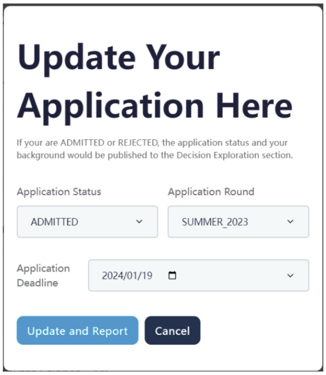

Fig. 7. Report Application

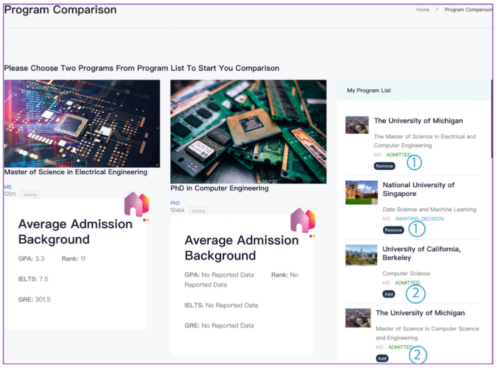

Fig. 8. Comparison

the programs based on their personal preferences and priorities, fostering a more engaged and personalized decision-making experience. By offering this level of control and transparency, our platform enhances the user's understanding and engagement in the college selection process, promoting greater satisfaction and confidence in their final decisions.

Dicision Exploration. Within our platform, applicants have access to an online chat feature that facilitates communication with other applicants. Moreover, applicants can explore detailed information about the schools they have applied to, providing them with comprehensive insights into the institutions of interest. Furthermore, applicants have the capability to review the application background of the decision report's owner. This feature empowers applicants to gauge the credibility and relevance of the decision report.

Dataset. The dataset used in our website was constructed through a two-fold approach. Firstly, we collected relevant data from Open-Source Datasets, ensuring a diverse and comprehensive pool of information. Additionally, we sourced program-related data from OpenCS.app, further enriching our dataset.

Moreover, our platform encourages active user participation, enabling users to contribute to the expansion of our dataset. Users have the option to upload their admission status and application backgrounds, providing valuable real-world data that enhances the accuracy and relevance of our system. This collaborative data-sharing approach ensures the dataset remains up-to-date and reflective of real user experiences, ultimately contributing to the overall effectiveness and reliability of our recommendation system.

Test of Accuracy. To evaluate the accuracy of our recommendation system, we conduct 50 tests using the background of 50 students whose final choice is known, which we simulate as input from website users. In each test, we evaluated the effectiveness of our recommendation system by examining whether the final choices of these simulated users were adequately covered within our generated recommendation list (Table 1).

Table 1. Test of accuracy

Hit Rate Level	Number	Rate
In the first recommendation	7	14%
In the top three recommendations	29	58%
In the top five recommendations	37	74%
In the top ten recommendations	44	88%

Our recommendation system achieved great results in accuracy tests. Specifically, 58% of the actual final choices were successfully included within the top three recommendation predictions, while 74% of the results were captured within the top five recommendation predictions. Furthermore, the overall hit rate of our recommendation list stood at an impressive 88%.

We speculate that the unsuccessful predictions may have been influenced by other factors, such as economic conditions and region. These factors were not taken into account in our recommendation system, but they could be important factors affecting some students.

5 Conclusion

Our recommendation system efficiently filters internet information, saving users time and addressing information overload. It generates a curated list of colleges based on multiple factors, ensuring objectivity and high success rates. Users can obtain recommended institutions without searching elsewhere, streamlining the application process. Our website offers user-friendly services and comprehensive information to empower applicants in making well-informed decisions for higher education. It aims to be a holistic and effective tool catering to diverse user needs.

We present a recommendation system using KNN-Based Collaborative Filtering to predict suitable programs and universities for users. Our system utilizes data from OpenCS as the foundational dataset and continually updates the database with information from platform users for real-time relevance. To accommodate larger data volumes and further optimize the usability of our recommendation system, we integrate Hadoop-based distributed clustering collaborative filtering. This approach enables efficient and parallel processing of vast datasets, leading to enhanced scalability and improved overall system performance.

Our recommendation system has shown positive outcomes in accuracy testing experiments, proving its effectiveness and usefulness. The system provides relevant and precise recommendations, assisting both graduates and current university students in their college selection process. It effectively addresses the challenges and time costs caused by data overload on the Internet.

Despite the decent accuracy of our recommendation system, there are still challenges to address. Ensuring the authenticity of user-uploaded results is crucial, and we propose using big data techniques for initial identification and manual processing of anomalous data. Additionally, economic costs and school location were found to be significant factors in decision-making. Further research and development are needed to incorporate these factors into our recommender system to improve its comprehensiveness and accuracy.

References

1. Adeniyi, D.A., Wei, Z., Yongquan, Y.: Automated web usage data mining and recommendation system using k-nearest neighbor (KNN) classification method. Appl. Comput. Inf. **12**(1), 90–108 (2016)
2. Adomavicius, G., Zhang, J.: Stability of collaborative filtering recommendation algorithms. Citeseer **10**(1.221), 7584 (2012)
3. Aggarwal, C.C.: Recommender Systems. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29659-3
4. Anand, S.S., Mobasher, B.: Intelligent techniques for web personalization. In: Mobasher, B., Anand, S.S. (eds.) ITWP 2003. LNCS (LNAI), vol. 3169, pp. 1–36. Springer, Heidelberg (2005). https://doi.org/10.1007/11577935_1
5. Anwar, T., Uma, V., Hussain, M.I., Pantula, M.: Collaborative filtering and KNN based recommendation to overcome cold start and sparsity issues: a comparative analysis. Multimedia Tools Appl. **81**(25), 35693–35711 (2022)
6. Awan, M.J., et al.: A recommendation engine for predicting movie ratings using a big data approach. Electronics **10**(10), 1215 (2021)
7. Baskota, A., Ng, Y.K.: A graduate school recommendation system using the multi-class support vector machine and KNN approaches. In: 2018 IEEE International Conference on Information Reuse and Integration (IRI), pp. 277–284. IEEE (2018)
8. Bawden, D., Holtham, C., Courtney, N.: Perspectives on information overload. In: Aslib Proceedings. vol. 51, pp. 249–255. MCB UP Ltd (1999)
9. Burke, R.: Hybrid recommender systems: survey and experiments. User Model. User-Adap. Inter. **12**, 331–370 (2002)
10. Chung, N.C., Miasojedow, B., Startek, M., Gambin, A.: Jaccard/tanimoto similarity test and estimation methods for biological presence-absence data. BMC Bioinf. **20**(15), 1–11 (2019)
11. Edmunds, A., Morris, A.: The problem of information overload in business organisations: a review of the literature. Int. J. Inf. Manage. **20**(1), 17–28 (2000)
12. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN model-based approach in classification. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) OTM 2003. LNCS, vol. 2888, pp. 986–996. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39964-3_62
13. Hardt, D.: The oauth 2.0 authorization framework. Tech. rep. (2012)
14. Isinkaye, F., Folajimi, Y., Ojokoh, B.: Recommendation systems: principles, methods and evaluation. Egypt. Inf. J. **16**(3), 261–273 (2015)
15. Kajdanowicz, T., Indyk, W., Kazienko, P.: Mapreduce approach to relational influence propagation in complex networks. Pattern Anal. Appl. **17**, 739–746 (2014)

16. Koren, Y., Bell, R.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 77–118. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_3
17. Kupisz, B., Unold, O.: Collaborative filtering recommendation algorithm based on hadoop and spark. In: 2015 IEEE International Conference on Industrial Technology (ICIT), pp. 1510–1514. IEEE (2015)
18. Li, G., Zhang, J.: Music personalized recommendation system based on improved KNN algorithm. In: 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 777–781. IEEE (2018)
19. Lipkus, A.H.: A proof of the triangle inequality for the tanimoto distance. J. Math. Chem. **26**(1–3), 263–265 (1999)
20. Lodderstedt, T., Bradley, J., Labunets, A., Fett, D.: Oauth 2.0 security best current practice. IETF Web Authorization Protocol, Tech. Rep. draft-ietf-oauth-security-topics-16 (2020)
21. Lynn, N., Emanuel, A.: A review on recommender systems for course selection in higher education. In: IOP Conference Series: Materials Science and Engineering, vol. 1098, p. 032039. IOP Publishing (2021)
22. Mahdi, M.N., Ahmad, A.R., Ismail, R., Natiq, H., Mohammed, M.A.: Solution for information overload using faceted search-a review. IEEE Access **8**, 119554–119585 (2020)
23. Nguyen, L.V., Vo, Q.T., Nguyen, T.H.: Adaptive KNN-based extended collaborative filtering recommendation services. Big Data Cogn. Comput. **7**(2), 106 (2023)
24. Owen, S., Friedman, B.E., Anil, R., Dunning, T.: Mahout in Action. Simon and Schuster (2011)
25. Patro, S.G.K., et al.: A hybrid action-related k-nearest neighbour (HAR-KNN) approach for recommendation systems. IEEE Access **8**, 90978–90991 (2020)
26. Pawlicka, A., Pawlicki, M., Kozik, R., Choraś, R.S.: A systematic review of recommender systems and their applications in cybersecurity. Sensors **21**(15), 5248 (2021). https://doi.org/10.3390/s21155248
27. Schneider, S.C.: Information overload: causes and consequences. Hum. Syst. Manag. **7**(2), 143–153 (1987)
28. Sharma, V., Trehan, T., Chanana, R., Dawn, S.: StudieMe: college recommendation system. In: 2019 3rd International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE), pp. 227–232. IEEE (2019)
29. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**, 421425 (2009)
30. Subramaniyaswamy, V., Logesh, R.: Adaptive KNN based recommender system through mining of user preferences. Wireless Pers. Commun. **97**, 2229–2247 (2017)
31. Tan, H., Guo, J., Li, Y.: E-learning recommendation system. In: 2008 International Conference on Computer Science and Software Engineering, vol. 5, pp. 430–433. IEEE (2008)
32. Tarus, J.K., Niu, Z., Mustafa, G.: Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. Artif. Intell. Rev. **50**, 21–48 (2018)
33. Willett, P.: Similarity-based approaches to virtual screening. Biochem. Soc. Trans. **31**(Pt 3), 603–606 (2003). https://doi.org/10.1042/bst0310603. PMID: 12773164
34. Willett, P.: Similarity-based virtual screening using 2D fingerprints. Drug Discovery Today **11**(23–24), 1046–1053 (2006)
35. Xiaojun, L.: An improved clustering-based collaborative filtering recommendation algorithm. Clust. Comput. **20**, 1281–1288 (2017)

36. Yan, L., Yin, C., Chen, H., Rong, W., Xiong, Z., David, B.: Learning resource recommendation in e-learning systems based on online learning style. In: Qiu, H., Zhang, C., Fei, Z., Qiu, M., Kung, S.-Y. (eds.) KSEM 2021. LNCS (LNAI), vol. 12817, pp. 373–385. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-82153-1_31
37. Zhang, X., Liu, H., Chen, X., Zhong, J., Wang, D.: A novel hybrid deep recommendation system to differentiate user's preference and item's attractiveness. Inf. Sci. **519**, 306–316 (2020)
38. Zhao, Z.D., Shang, M.S.: User-based collaborative-filtering recommendation algorithms on hadoop. In: 2010 Third International Conference on Knowledge Discovery and Data Mining, pp. 478–481. IEEE (2010)
39. Zheng, Y., Mobasher, B., Burke, R.: CARSKit: a java-based context-aware recommendation engine. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), pp. 1668–1671. IEEE (2015)