Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi

Bilecik Seyh Edebali University Journal of Science

12(1), 233-241, 2025

DOİ: 10.35193/bseufbd.1513170

e-ISSN: 2458-7575 (https://dergipark.org.tr/tr/pub/bseufbd)

Araştırma Makalesi - Research Article

Enhancing Hotel Recommendations through Feature-based Clustering

Özellik Tabanlı Kümeleme ile Otel Tavsiyelerinin Geliştirilmesi

Ömer Arifoğulları[1*], Gülfem Işıklar Alptekin[2], Günce Keziban Orman[3]

Geliş / Received: 09/07/2024 ***Revize / Revised: 31/08/2024*** ***Kabul / Accepted: 18/09/2024***

ABSTRACT

This paper addresses the challenge of sparse interaction data in recommendation systems for the hotel industry. Due to the infrequent nature of hotel stays (often once or a few times annually), customer-product interaction data is typically sparse, hindering the effectiveness of traditional collaborative filtering techniques. We propose a novel hybrid recommendation framework specifically designed for this scenario. Unlike conventional systems that rely solely on user preference similarity, our framework leverages hotel clustering based on binary attributes to segment the product space. User interactions are analyzed within these clusters, leading to a more refined recommendation process. We take advantage of several clustering and feature reduction techniques and assign the final recommendation through ballot scoring. The experiments are performed on a real-world hotel sales data set including both sales information and hotel attributes. We evaluate our methodology and demonstrate significant improvements over baseline approaches which is the case of not using the found clusters for recommendation. The proposed framework achieves a two-fold increase in both the number of users receiving recommendations and the number of correct recommendations. These results highlight the potential of cluster- based recommendations for mitigating sparsity issues in tourism recommender systems.

Keywords- Recommendation System, Collaborative Filtering, Hybrid Recommendation System, Ballot Score

ÖZ

Bu makale, otel endüstrisi için öneri sistemlerinde seyrek etkileşim verilerinin yarattığı zorlukları ele almaktadır. Otel konaklamalarının genellikle yılda bir veya birkaç kez olması, müşteri-ürün etkileşim verilerini seyrek kılar. Bu da geleneksel işbirlikçi filtreleme tekniklerinin etkinliğini engeller. Bu senaryo için özel olarak tasarlanmış yeni bir hibrit öneri çerçevesi öneriyoruz. Yalnızca kullanıcı tercihi benzerliğine dayanan geleneksel sistemlerin aksine, çerçevemiz ürün uzayını bölümlere ayırmak için ikili özniteliklere dayalı otel kümelemesinden yararlanmaktadır. Kullanıcı etkileşimleri bu kümeler içinde analiz edilerek daha rafine bir tavsiye süreci ortaya çıkar. Çeşitli kümeleme ve özellik azaltma tekniklerinden yararlanıyor ve nihai tavsiyeyi oylama puanlaması yoluyla atıyoruz. Deneyler, hem satış bilgilerini hem de otel niteliklerini içeren gerçek dünya otel satış veri seti üzerinde gerçekleştirilmiştir. Sonuçlara göre metodolojimizi kümeleme kullanmayan temel yaklaşımlara göre önemli gelişmeler gösteriyor. Önerilen çerçeve, hem tavsiye alan kullanıcı sayısında hem de doğru tavsiye sayısında iki kat artış sağlıyor. Bu sonuçlar, turizm tavsiye sistemlerindeki seyreklik sorunlarını hafifletmek için küme tabanlı önerilerin potansiyelini vurgulamaktadır.

Anahtar Kelimeler- Öneri Sistemi, Işbirlikçi Filtreleme, Hibrit Öneri Sistemi, Oylama Puanı

[1*]Corresponding Author Contact:mr.arifogullari@gmail.com (https://orcid.org/0009-0002-6146-264X)
Department of Computer Engineering, Galatasaray University, 34349, Istanbul
[2]Contact: gisiklar@gsu.edu.tr (https://orcid.org/0000-0003-0146-1581)
Department of Computer Engineering, Galatasaray University, 34349, Istanbul
[3]Contact: korman@gsu.edu.tr (https://orcid.org/0000-0003-0402-8417)
Department of Computer Engineering, Galatasaray University, 34349, Istanbul

I. INTRODUCTION

Following a significant recovery in 2023, the initial UNWTO World Tourism Barometer of the year indicated that international tourism nearly returned to its pre-pandemic levels, achieving 88% of its former volume with an estimated 1.3 billion international arrivals [1]. Similarly, Turkey experienced a significant upturn in tourism revenue, marking a 16.9% increase from the previous year to reach 54,315,542,000, with accommodation spending also rising by 27% according to the National Statistical Report [2]. This revitalization is vital for economies worldwide, as tourism significantly generates revenue, creates jobs, and fosters cultural exchange. The increasing shift towards online search and booking further highlights the critical role of advanced recommendation systems in maintaining competitive advantage by effectively utilizing the extensive data collected by travel agencies.

The literature on recommendation systems categorizes them into three primary types: collaborative filtering, content-based filtering, and hybrid filtering. Collaborative filtering recommends items based on user behavior and patterns [3-5]. It examines user activity to find comparable users and their preferences. Once similar users are identified, the system proposes items that previous customers with similar preferences have enjoyed or bought. Content-based filtering techniques analyze the items and a user's behavior [6]. The user's previous preferences are stored in a user profile. An item is analyzed to determine the similarities with the user's profile and previous choices so that the system can recommend items that the user has already liked. The hybrid filtering technique combines recommendation techniques to profit from the strength of different approaches to obtain better recommendations and overcome the setbacks of one algorithm [7].

Recommendation systems face two primary analytical challenges: diversity and sparsity [8, 9]. Sparsity mostly arises from infrequent user-item interactions leading to weaker recommendations [10]. On the other hand, diversity is difficult to achieve without users' demographic values such as gender, socioeconomic status, and education level. To address these issues, one viable approach is the clustering of users or items, followed by designing a recommendation system for each cluster. This strategy helps mitigate the effects of sparsity and enhances the diversity of recommendations. Nonetheless, a significant obstacle is the often-limited availability of detailed user features in user databases, while collecting item features tends to be easier.

Our paper focuses on precisely clustering datasets describing hotel features through binary variables. Relevant literature includes studies on hotel clustering and the importance of feature analysis [11,12]. A hotel recommendation system based on the link prediction method is introduced by [13]. They constructed a customer-hotel bipartite network using the data crawled from TripAdvisor.com. Another point of view has been to cluster travelers according to their past choices and their personality traits [14,15]. We encounter that Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is commonly used for hotel recommendation systems. A user-defined weights to prioritize hotel features and suggest optimal choices is employed in [16-18] presents a fuzzy and nonlinear programming approach that captures the dynamic nature of user preferences and contextual factors, optimizing weights based on real-time user choices for personalized recommendations. Nevertheless, these methods are not tailored for high sparsity of interactions. Besides, we encountered a work that explores the influence of customer satisfaction on hotel recommendations, analyzing the relationship between satisfaction features and recommendation likelihood, emphasizing the importance of price-quality ratio, customer service, and facilities [19]. This approach differs from our proposal at the point that it focuses on user satisfaction, so that the feedback is important in these cases. However, in our proposal, we do not build an explicit framework. We propose to take advantage of clustering for overcoming the sparsity of sales data. Utilizing a real-life dataset from a leading Turkish tourism company covering over 61% of registered tourist facilities, our work aims to develop an accurate hybrid recommendation system that considers both item features and user similarities.

Contributions of our study include:

i. Developing a hybrid recommendation framework that takes into account the item features from the perspective of content-based filtering and user similarities for collaborative action.

ii. Implementing this framework on a real-world hotel sales dataset characterized by high sparsity.

iii. Experimentally showing the superior performance of our hybrid model compared to traditional collaborative approach.

iv. Employing sales data for hotel recommendations as opposed to relying on customer reviews or ratings.

v. Providing detailed methodologies and pseudo-codes for the hybrid recommendation system, which may be re-applicable for any other datasets with binary item features.

The structure of this article is outlined as follows: Section II examines the cluster-based hybrid recommendation model proposed by this study. In Section III, we present the experimental setup, alongside the results derived from these experiments. Finally, Section IV offers concluding remarks and encapsulates the findings of the paper.

II. CLUSTER-BASED HYBRID RECOMMENDATION FRAMEWORK

In the context of our study, the products are 2,562 hotels, and most users visit only one or two of them. Addressing this challenge, we grouped hotels into clusters based on their features. We then identified for each user, other users who had visited hotel clusters most similar to the ones they visited.

A. Hotel Clustering

The inherent challenge in analyzing a sparse binary matrix of hotel features necessitates sophisticated analytical methodologies. Our prior work introduced a comparative analysis of various clustering and dimension reduction techniques applied to this dataset [20]. Given that most clustering algorithms are primarily designed for datasets with numerical features, an initial step involved converting the dataset in question from binary to numerical format. In our prior work, we specifically focus on Sparse Principal Component Analysis (SPCA) [21] and Non-Negative Matrix Factorization (NMF) [22]. Dimension reduction is essential for making complex datasets more manageable and deriving meaningful insights with less computational resource expenditure. In conventional PCA, the principal components are derived as linear combinations of all original features [23]. Conversely, SPCA generates principal components from linear combinations of a select subset of the original features. The primary objective of SPCA is to determine a minimal subset of features that explain the maximum variance within the data, effectively excluding noise or irrelevant information. NMF is also a dimensionality reduction technique similar to PCA. Unlike PCA, NMF models offer straightforward interpretability. Nonetheless, NMF is not universally applicable across all datasets; the sample features must be non-negative, like in our hotel dataset.

After implementing dimension reduction techniques, we employed four well-known distinct clustering algorithms for further analysis: K-means, hierarchical agglomerative, DBSCAN, and OPTICS [20]. The effectiveness of two distinct dimension reduction methodologies, alongside the performance of four clustering algorithms, is thoroughly evaluated in [20]. Our analysis focuses on two critical criteria of the resultant cluster structure: firstly, achieving a clustering configuration characterized by high intra-similarities, and secondly, ensuring that the clusters are well-separated, with significant inter-cluster distances. This whole clustering efforts are encapsulated within the 'Hotel Clustering' process. Utilizing optimally performing clusters identified in [20], we aim to formulate more accurate hotel recommendations for customers.

B. Recommendation Engine

The methodology for this recommendation engine process is outlined in *Algorithm 1 Recommendation Engine* shown in Table 1. Our recommendation engine takes parameters the *weight*, which is used in ballot scoring which is described in the following part, and the *recommendation_number*, which express the number of items which will be recommended to each user, outputting a set of *recommendations* for each user. To evaluate the success of our system, we conducted preliminary supervised learning experiments, with the setup detailed in Section III. The process begins by taking the training data from the dataset (*line #1*). We then take the clusters of the found hotels obtained from the application of the clustering methodology described previously to group hotels in the previous section in (*line #2*). Hotels are then replaced with their cluster equivalents in the dataset (*lines #3* and *#4*), thus creating a modified dataset for recommendations. Each row in this set represents a user's previously purchased hotel cluster and other details related to that purchase. Utilizing the *getUsers(...)* function (*line #5*), we construct the *userlist* matrix, where each row corresponds to a user vector. The size of these vectors is equal to the total count of hotel clusters.

Table 1. Pseudo-code of Recommendation Engine

```
Algorithm 1 Recommendation Engine
Require: weight, recommendation_number
Ensure: recommendations, a set including recommendation_number recommendations
for each customer
1: Set data as the extracted training data set from the revised sales table
2: Set clusters as the found hotel clusters from the hotel binary features set
3: Match hotel IDs from the training data set with hotel IDs in clusters
4: Replace the hotel ID of each sales record in the training data set with the
corresponding cluster ID's after matching
5: userlist ← getUsers(data)
6: cosines  ← getCosineSimilarity(userlist)
7: for user from userlist do
8:     user rec ← getRecommendations(user, cosines,
weight, recommendation_number)
9:     recommendations ← aggregate(recommendations, user_rec)
10: end for
```

It is important to highlight that utilizing hotel clusters instead of individual hotels significantly impacts the dimensionality of the vectors in question. If individual hotels were used, the vector size would correspond to the total number of hotels within the system. Considering that the number of clusters is much smaller than that of hotels, the resulting *userlist* matrix demonstrates a significant decrease in sparsity. This reduction translates to fewer zeros in each vector of the *userlist* matrix. While this decrease in zeros is noteworthy, it's essential to recognize that there might also be a decrease in ones. This is because multiple visits by a user to hotels within the same cluster are aggregated into a single cluster representation. However, if users in the system have interacted with different hotels, depending on the level of diversity, there may be interactions with the number of clusters equal to the number of hotels. Generally, various users exist in real-world systems, resembling both the first and second scenarios. Therefore, the decrease in the number of ones may not be as dramatic as the decrease in the number of zeros. Cosine similarity, calculated in *line #6*, is used to determine the similarity between user vectors within the *userlist*. The equation for cosine similarity is provided in Eq. 1 for two binary vectors *x* and *y* at the identical size.

$$\mathrm{cosine}(x, y) = \frac{\sum_i x_i \cdot y_i}{\sqrt{\sum_i x_i} \cdot \sqrt{\sum_i y_i}} \quad (1)$$

Cosine similarity was selected for its widespread use and reliability in computing similarity between binary vectors. A value of 1 indicates identical user vectors and a value of 0 signifies no common elements between the vectors. Subsequent steps in Algorithm 1 involve a straightforward sub-procedure to generate recommendations for each user. We detailed this sub-procedure in *Algorithm 2 get Recommendations* shown in Table 2. Specifically, *line #1* of this sub-procedure filters users according to their cosine scores, choosing the top *recommendation_number* users that are most similar for each target user. The core concept behind our recommendations is to propose hotels that the user had not visited before but were visited by users most similar to them. Nonetheless, the process of selecting *recommendation_number* users introduces additional complexities.

Table 2. Pseudo-code of getting recommendations through ballot scoring

```
Algorithm 2 getRecommendations
Require: user,cosines,weight, recommendation_number
Ensure: user_rec, a set including recommendation_number recommendations for
given user
1: Set similar_users, similarities ← extractSimilarUsers(user,
   cosines, recommendation_number)
2: Set neighbors, itemNumbers ← ballotScores(similar_users, similarities,
weight, recommendation number)
3: Get hotel IDs as many as the number of items of each neighbor from neighbors,
itemNumbers, ensuring these IDs are distinct from those already purchased by
the user.
```

Let us consider the scenario where the *recommendation_number* is set to 3, and for a given user, the similarities of the top 3 similar users are 0.9, 0.1, and 0.1, respectively. In this case, while making recommendations, it is necessary to receive more recommendations from the first user and perhaps fewer or no recommendations from the other two. To achieve this, we applied a technique, widely utilized in voting systems known as "ballot scoring" [24].

Ballot scoring is a method that facilitates the distribution of n preferences among $k>=n$ candidate demands based on the quality of those demands. In this scoring framework, not all k candidates may receive a share from the preferences; rather, candidates receive shares based on their quality scores. This method uses a weighting parameter to adjust the distribution strength based on quality. In our context, the n preferences correspond to the recommendations to be made, ultimately totaling the *recommendation_number*. The k options, in our case, are the similar users identified, also equal to the number of recommendations. If each user receives exactly one recommendation, the preferences are evenly distributed among all users. This system enables a dynamic distribution where more recommendations can be allocated to users with higher similarity scores. The allocation of recommendations per user is determined using a straightforward formula, as illustrated in Formula 2. This equation is obtained by rounding the product of the user's similarity score, *s*, and the weight, *w*, to the nearest integer.

$$\mathrm{ballot}(w, s) = \begin{cases} \lfloor w \cdot s \rfloor, \text{if the fraction part of } w \cdot s \ < 0.5 \\ \lceil w \cdot s \rceil, \text{otherwise} \end{cases} \tag{2}$$

The ballot count for each similar user, as determined in *line #2* of Algorithm 2 using Eq. 2. This procedure facilitates the allocation of recommendation rights up to the specified *recommendation_number*, prioritizing users with the highest ballot counts. Although it is possible under this method that some similar users may not contribute recommendations, it primarily ensures that recommendations are predominantly sourced from users exhibiting the greatest similarity. Consequently, our approach can be characterized as a hybrid recommendation engine, since it seamlessly merges the evaluative strengths of collaborative filtering—through the assessment of user similarities based on their preferences—with a content-based approach that categorizes these preferences by their attributes.

III.EXPERIMENTS AND RESULTS

We employed two distinct datasets for our analysis: an extensive collection of hotel features, and anonymized sales data, which documented transactions executed by the sales department from 2019 to 2022. The hotel feature dataset includes properties that have been visited at least once by customers, incorporating 2,562 distinct hotels each characterized by 27 features. These features, which are all binary, indicate whether the relevant hotel has the relevant feature or not. Some examples of features are "child-baby friendly","ski-hotel","pool","sandy beach","next to sea shore". Despite the challenges that sparse and binary datasets pose for clustering [25], such features are advantageous for content-based filtering systems by providing clear indications of attributes that align with user preferences or item descriptions.

Our second dataset contains sales data (40,599 sales records with 32 features) that indicates which hotel was chosen and bought by which customer. Unlike many hotel recommendation systems that rely on customer reviews or ratings from platforms like Tripadvisor or Trivago [11,13], our approach utilizes sales data, offering direct insights into customer preferences and outcomes. This leads to a dataset characterized by sparser customer-hotel interactions but provides a basis for more reliable evaluations. The unmodified raw hotel dataset was imported ensuring the Hotel ID column contained no null values, and each binary feature column had at least one '1' value. After verifying the uniqueness of the Hotel ID column and removing duplicate rows, we finalized a hotel feature table with 2,562 rows and 27 columns. The sales dataset was subject to comprehensive cleaning, including the removal of suspicious records, such as those with incorrect regions or room types, and the rectification of inconsistencies, such as sales records without corresponding hotel feature entries, resulting in the exclusion of 9,212 rows. These adjustments culminated in a streamlined sales dataset comprising 28,439 records, ready for further analysis. The detailed results of the clustering step on this data set are previously reported in [20]. Here, we mainly focus on the recommendation step and detail the results of ballot scoring.

A. Setup and Performance Measurement

We conducted a supervised experiment to evaluate the efficiency of our proposed approach. We partitioned 10% of each user's records from the main sales dataset into a dedicated test set while retaining the remainder for training purposes. Notably, due to the infrequent nature of hotel bookings for many users, a significant portion of the dataset consisted of users with only a single transaction, rendering them ineligible for inclusion in the test set. Consequently, out of the 2588 users, only 600 were incorporated into the test set, with the remainder categorized as cold-start users.

In the training phase, we ran both the cluster-based approach proposed in Algorithm 1 and a baseline method, differing solely in the exclusion of steps pertaining to hotel ID vector similarities (omitting *lines #2, 3*, and *4* in Algorithm 1). Each method was configured to generate 10 recommendations per user. The ballot scoring we use tells us how many recommendations we can get from each similar user for a total of 10 recommendations. However, there is a risk that hotels already recommended by different users will be recommended again, failing to produce any recommendations altogether. In some cases, similar users visit only the same hotels as the user for whom the recommendations are generated. This may also cause not to produce any single recommendation for some users. Such scenarios, common in datasets characterized by rare transactions such as hotel bookings, necessitated the inclusion of additional performance metrics beyond correct recommendations. That is why; we included the number of users for whom the recommendation could be produced as performance metrics besides the correct recommendations. To evaluate the performance of our methods, we employed the following metrics:

- Number of distinct users for whom recommendation could be generated in the training set
- Number of distinct users for whom a recommendation could be made in the test set
- Number of correct recommendations for the test set
- Ratio of correct recommendations made for the test set to the expected number of complete recommendations.

Given that our test set comprises 600 users, each associated with only one sales data point allocated to the test set, the anticipated number of complete recommendations stands at 600. We systematically varied the ballot score weight across a range of values from 1 to 10, with the results evaluated accordingly.

B. Results

We first evaluate the impact of ballot scoring on the recommendation distribution for selected users to reveal the behavior of ballot scoring. Table 3 illustrates the distribution of similar users, their cosine similarity, and corresponding ballot scores generated by executing Algorithm 1 with *recommendation_number* set to 10 and *weight* adjusted to 2 and 5, respectively. For instance, when recommending 10 items to customer ID 171, employing a *weight* of 2 results in recommendations sourced from 9 distinct similar users. Conversely, when the *weight* is increased to 5, recommendations originate from 3 different users. The results reveal a trend where a lower *weight* increases the diversity and number of recommending users.

Table 3. Cosine similarity and ballot scores of similar customers for target customer ID 171 across different weights

Recommendation Order	Similar Customer	Cosine Similarity Score	Ballot Score for w=2	Ballot Score for w=5
1	47617	0.816	**2**	**4**
2	17208	0.730	**1**	**4**
3	1299804	0.707	**1**	**2**
4	17132	0.707	**1**	1
5	17907	0.707	**1**	1
6	19228	0.707	**1**	1
7	19133	0.707	**1**	1
8	17082	0.707	**1**	1
9	75469	0.707	1	1
10	23012	0.707	1	1

The evaluation results of our proposed cluster-based approach and the baseline method, utilizing various weights in the ballot scoring scheme, are depicted in Figure 3. In our study, the training dataset comprises 2588 users. These plots provide a comprehensive visual representation of the impact of weight variation on the effectiveness of the ballot scoring mechanism in both our proposed approach and the baseline method.

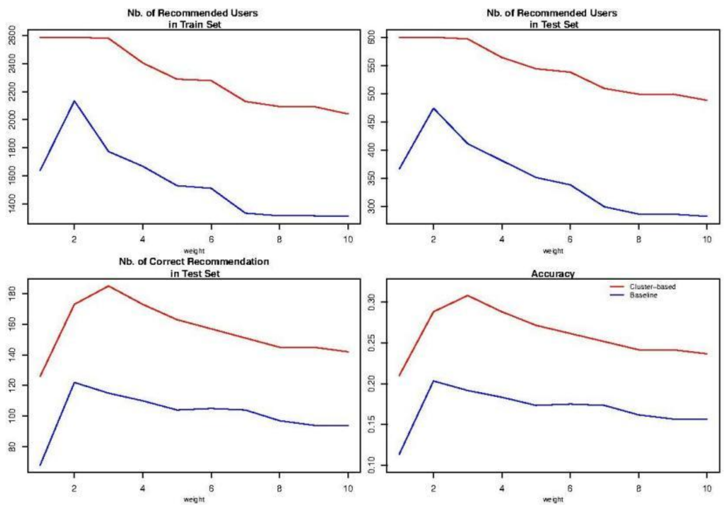

Figure 3. The change of the number of total recommended and correctly recommended users in Train and Test sets with the change of ballot weight

Our proposed approach demonstrates the capability to provide recommendations to all users across various weight settings, including *weight* of 1, 2, and 3. This inclusivity extends to users in the test set as well. However, as the *weight* parameter increases, the number of users eligible for recommendations gradually decreases. Notably, our cluster-based approach exhibits twice the efficacy of the baseline method in terms of the number of users that can be recommended across all *weight* configurations. After a certain *weight* value, the reason for the decrease in the number of users who can be recommended is the large number of recommendations received from a single user. This problem may not occur in datasets with hundreds of transactions per user when the *weight* is high, but as mentioned before, most of the users visited only one or two hotels in our data. Based on our observations, our dataset's optimal *weight* parameter emerges as 3. In the plots depicting the number of correct recommendations and their ratio relative to the anticipated total recommendations, our proposed cluster-based approach consistently outperforms the baseline, achieving double the success rate. This underscores the effectiveness of our approach in enhancing correct recommendation and completeness. Across all methods evaluated, this *weight* setting consistently yields the highest levels of success.

IV. CONCLUSION

In this research, we have developed a novel hybrid recommendation framework tailored for hotel transactions, addressing the challenge of sparse customer-product interactions. Our framework introduces a unique methodology by segmenting hotels based on their attributes and evaluating customer similarities through interactions with these hotel clusters. A comprehensive clustering phase was conducted to determine optimal segmentation, utilizing cosine similarity to quantify customer similarities. Subsequently, customers were prioritized based on these similarities, with recommendations generated via a weighted ballot scoring system that dictated the allocation of recommendations from each user. Our findings were benchmarked against a baseline approach that considers individual hotels, revealing that our method resulted in two times more success both in terms of producing the number of recommended users and the success of recommendations.

The core insight from this study indicates the efficacy of employing item segmentation to enhance correct recommendation. Furthermore, the adaptability of our framework permits its application across diverse datasets, offering flexibility in selecting clustering tools for segmentation and integrating various collaborative filtering techniques within the recommendation engine phase. This initial study was intentionally focused on establishing and algorithmically detailing our framework, foregoing a broader range of experimental conditions. Future research will explore variations in experimental approaches, expanding upon the groundwork set by this research. For now, the data set we have used consists of only hotel sales and does not reflect if the customers liked the hotels. That is why we cannot measure the success of customer satisfaction about recommended hotels. As a novel perspective, this work can be extended to measuring the performance according to the pleasant hotels for a system including customer feedback.

ACKNOWLEDGEMENT

This work is supported by the Scientific and Technological Research Council of Türkiye (TÜBITAK) under the grant number 122N701 and by the Galatasaray University Research Fund (BAP) within the scope of project number FBA-2024-1240, and titled 'Bağlantı Tahmini Yöntemleri ve Sınıf Dengesizliği Sorununa Çözüm Yaklaşımları'. The original data set used in the experiments in this study was provided by Setur Company. The clustering results of this work were previously published at a conference with reference [20].

REFERENCES

[1] Unwto world tourism barometer and statistical annex, january 2023. https://www.e-unwto.org/doi/epdf/10.18111/wtobarometereng.2023.21.1.1?role=tab. Accessed: 2024-02-01.
[2] Turizm İstatistikleri, ekim-aralık, (2023). *https://data.tuik.gov.tr/Bulten/Index? p=Turizm-Istatistikleri-IV.-Ceyrek:-Ekim—Aralik,-2023-53661*. Accessed: 2024-02-01.
[3] Liu, L.L., & Mpcm, J. (2023). *Multi-modal user portrait classification model based on collaborative learning. Information Technology and Control*, 52(4):867–877.
[4] Lee, M., & Kim, H.J. (2023). *A collaborative filtering model incorporating media promotions and users' variety-seeking tendencies in the digital music market. Decision Support Systems*, 174:114022.
[5] Iftikhar, A., Mustansar A. G., Mubbashir A., Zahid M., & Muazzam M. (2020). *An improved product recommendation method for collaborative filtering. IEEE ACCESS*, 8:123841– 123857.
[6] Jozani, M., Liu, C.H., & Choo, KK. R. (2023). *An empirical study of content-based recommendation systems in mobile app markets. Decision Support Systems*, 169:113954.
[7] Liu, D.R, Lai, C.H., & Lee, W.J. (2009). *A hybrid of sequential rules and collaborative filtering for product recommendation. Information Sciences*, 179(20):3505–3519.
[8] Wihartiko, F. D., Nurdiati S., Buono, A., & Santosa, E. (2023). *Multi-objective entropy optimization model for agricultural product price recommendation problem. Engineering Letters*, 31(4),
[9] Li, G.H.C.M.Z., Wu, J., & Chen, Y. (2023). *Multi-hypergraph neural network with fusion of location information for session-based recommendation. IAENG International Journal of Applied Mathematics*, 53(4):1389–1398.
[10] Bobadilla, J., Ortega, F., Hernando, A. & Gutiérrez, A. (2013). *Recommender systems survey. Knowledge-Based Systems*, 46:109–132.
[11] Akyol, M. (2021). *Clustering hotels and analyzing the importance of their features by machine learning techniques. Bilgisayar Bilimleri ve Teknolojileri Dergisi*, 2(1):16–23.
[12] Rodrıguez-Victoria, O.E., Puig, F. & González-Loureiro, M. (2017). *Clustering, innovation and hotel competitiveness: evidence from the colombia destination. International Journal of Contemporary Hospitality Management*, 29(11):2785–2806.
[13] Kaya, B. (2020). *A hotel recommendation system based on customer location: a link prediction approach. Multimedia Tools Appl.*, 79(3–4):1745–1758.
[14] Chen, T. (2020), *A fuzzy ubiquitous traveler clustering and hotel recommendation system by differentiating travelers' decision-making behaviors. Applied Soft Computing*, 96:106585.
[15] Lee, T. H., & Jan, F.H. (2023). *How does personality affect covid-19 pandemic travel risk perceptions and behaviors? evidence from segment analysis in taiwan. Sustainability*, 15(1).
[16] Forouzandeh, S., Berahmand, K. Nasiri, E. & Rostami, M. (2021). *A Hotel Recommender System for Tourists Using the Artificial Bee Colony Algorithm and Fuzzy TOPSIS Model: A Case Study of TripAdvisor. International Journal of Information Technology & Decision Making (IJITDM)*, 20(01):399–429.
[17] Kwok, P.K., & Lau, H.Y.K. (2019). *Hotel selection using a modified topsis-based decision support algorithm. Decision Support Systems*, 120:95–105.
[18] Chen, T., & Chuang. Y.H. (2018). *Fuzzy and nonlinear programming approach for optimizing the performance of ubiquitous hotel recommendation. Journal of Ambient Intelligence and Humanized Computing*, 9(2).
[19] Lee, S.H., Yun, J.J., Diaz, M.M. & Duque, C.M. (2021). *Open innovation through customer satisfaction: A logit model to explain customer recommendations in the hotel sector. Journal of Open Innovation: Technology, Market, and Complexity*, 7(3):180.
[20] Arifogullari, O., & Orman, G.K. (2023). *On experimental study of hotel clustering. In Proceedings of the International MultiConference of Engineers and Computer Scientists 2023*, IMECS 2023, pages p. 82–87, Hong Kong.
[21] Johnstone, I. M., & Yu, L. A. (2009). *On consistency and sparsity for principal components analysis in high dimensions. Journal of the American Statistical Association*, 104(486):682–693.
[22] Inderjit, S. D., & Suvrit, S. (2005). *Generalized nonnegative matrix approximations with bregman divergences. In Proceedings of the 18th International Conference on Neural Information Processing Systems, NIPS'05*, page 283–290, Cambridge, MA, USA, MIT Press.
[23] Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A, 374(2065):20150202–20150202, 2016.
[24] Taebel, D.A. (1975). *The effect of ballot position on electoral success. American Journal of Political Science*,

19(3):519–526.
[25] Ye, M., Zhang, P., & Nie, L. (2018). *Clustering sparse binary data with hierarchical bayesian bernoulli mixture model. Computational Statistics Data Analysis*, 123:p. 32–49.