A Reproducibility Study of Personalized Item Frequency for Next Basket Recommendation

Sebastiaan Dijkstra[1] [0009-0002-6402-5816], Giray Keskin[1], and Bram Veenman[1] [0009-0009-2039-766X]

University of Amsterdam, Amsterdam, Netherlands
http://www.uva.nl/

Abstract. Next Basket Recommendation (NBR) is the task of predicting a user's future purchases based on purchasing history data. This task is of particular interest to grocery shopping retailers and other forms of e-commerce, but the underlying principles are also relevant for wider academic and industrial contexts. Although most recent work in the field focuses on neural-based networks, there is evidence that simple algorithms like k-nearest neighbors (kNN) can equal or even outmatch more complex methods. This paper analyzes TIFU-KNN, a kNN model that aims to integrate Personalized Item Frequency (PIF) information in the NBR task. To better understand the applicability and generalizability of this model, we reproduce the original study and expand on it in various ways: (1) We evaluate the model on data from other domains, (2) we alter the kNN method with different distance metrics, (3) we compare kNN with dense-based clustering algorithms. Our results indicate that ostensibly simple nearest neighbor-based models can perform on a similar or higher level as recurrent or dense neural networks, and that choosing the right model and clustering method significantly affects the quality of next-basket predictions.

Keywords: next-basket recommendation · k-nearest neighbors · item frequency · recurrent neural networks

1 Introduction

Recommender systems play a pivotal role across various domains, including academia [2, 19] and various retail sectors [7, 14]. One specialized task within this field is Next-Basket Recommendation (NBR), which seeks to predict a set of items (basket) that a user is likely to purchase based on the user's purchasing history. This task has particular relevance to grocery shopping, given the repetitive nature of consumer purchases [5, 16]. Nevertheless, the applicability of NBR extends beyond this context, and its principles are utilized in broader commercial and academic environments [9, 20].

Various models have been employed to encapsulate the NBR problem, including Markov Chains [21], Transformers [25], and Recurrent Neural Networks (RNN) [11, 26]. Each of these methods employs distinct techniques to create basket representations. However, they consider users' personal histories only implicitly, limiting their capacity to directly capture the *personal* frequency of item purchases.

This paper presents an investigation into temporal-item-frequency-based user-KNN (TIFU-KNN) [12], a model based on nearest-neighbor principles that harnesses personal item frequency (PIF) patterns for use in NBR contexts. Despite its ostensibly simple implementation, TIFU-KNN has demonstrated impressive performance in NBR tasks. However, the consistency of this performance across different scenarios remains to be fully explored.

To address the above, we first replicate the study of [12], verifying the results to establish a baseline for subsequent experiments. We then extend their work through three main expansions:

- We evaluate the TIFU-KNN model's performance in new domains, showcasing its generalizability and adaptability across diverse contexts.
- We experiment with alternative distance metrics (Cosine and Manhattan distances) in the kNN algorithm to assess their impact on the model's performance.
- We incorporate the density-based clustering algorithms DBSCAN and HDBSCAN as alternative approaches to kNN. These algorithms aim to leverage the inherent groupings in the data.

Our research delves into the feasibility and effectiveness of the TIFU-KNN model in different scenarios. Specifically, this study aims to answer the following research questions:

1. How well does the performance of the TIFU-KNN model generalize across diverse datasets and domains?
2. In what ways is the performance of the TIFU-KNN model impacted by adaptations of any changes to the basic kNN method?
3. What are the potential outcomes of integrating PIF information in other models, and how does this compare to the TIFU-KNN model?

The answers to these questions could advance the existing knowledge and contribute to developing more effective models for the NBR problem.

In the following sections, we initially provide background on NBR, PIF, and the TIFU-KNN model. We then discuss related research in the NBR field, including frequency-based, neighbor-based, and deep learning techniques. Following this, we explain in detail our methodology for replicating original results, assessing model generalizability, experimenting with alternative distance metrics, and incorporating density-based clustering. Each of these components supports our objective of extending the original research on TIFU-KNN, providing a more robust understanding of its performance in varied contexts.

2 Background

2.1 Next-basket Recommendation (NBR)

While the foundational concept of NBR has been addressed in the introduction, its formal definition provides further insight into its complexity.

Let us denote by $U = \{u_1, u_2, ..., u_n\}$ the set of users, and by $I = \{i_1, i_2, ..., i_m\}$ the set of items. For each user $u \in U$, there exists a sequence of shopping baskets $B_u = \{b_1, b_2, ..., b_t\}$, where each basket $b \in B_u$ is a set of items procured during a particular transaction, i.e., $b \subseteq I$.

The NBR task proposes a recommendation function $f : B_u \rightarrow I$. Given the sequence of past baskets B_u for a user u, this function is expected to predict the items in the subsequent basket b_{t+1}. The construction of f should be such that it optimizes the relevance of the recommended items according to the user's preferences, which are indicated in B_u.

This formal representation highlights the temporal dynamics inherent to the problem and the sequential nature of user shopping behaviors. These factors contribute to the complexity of NBR within the recommender systems domain.

2.2 Personalized Item Frequency (PIF)

For each user u and each item i, we define the PIF_{ui} as the number of instances in which user u has purchased item i throughout their shopping history. In this way, PIF quantifies the frequency with which users purchase an item and offers critical insights into their purchasing preferences and habits within NBR [12].

This metric provides a rich source of information that reflects two purchase patterns beneficial for high-accuracy NBR task predictions. Firstly, users frequently purchase the same items [4]. Simply recommending the most frequently purchased items can even outperform state-of-the-art neural models, as shown by [11]. The second pattern is collaborative: similar users tend to show similar purchase behavior. The authors of TIFU-KNN show that PIF captures both of these important patterns by analyzing real-world online grocery shopping data from services like Instacart [1].

Despite its significance, PIF is often insufficiently considered in conventional NBR approaches. [12] use synthetic data to demonstrate how Recurrent Neural Networks cannot learn vector addition sufficiently. Although RNNs can approximate any computable function [23], they prioritize capturing the more complicated sequential patterns in users' shopping history instead of simply aggregating vectors in the same way as vector addition. These findings align with other recent work that compares the performance of neural top-n recommendation algorithms with much simpler heuristics [10]. The observation of this phenomenon underscores the need for novel methods to integrate PIF information effectively into the NBR process.

[1] https://www.kaggle.com/c/instacart-market-basket-analysis

2.3 TIFU-KNN

Temporal **I**tem **F**requency-based **U**ser-KNN (TIFU-KNN) [12] is a k-nearest neighbors (kNN) [3] method that aims to capture personal item frequency information for Next Basket Recommendation directly. The algorithm is simple but surprisingly effective. Its goal is to make a prediction $\mathbf{P}$ over all items in the dataset so that the items with the largest values in $\mathbf{P}$ form the recommendation of the next basket. This prediction is calculated from two components: the target user's vector representation $\mathbf{u_t}$, which captures the personal repeated purchase pattern, and the average of the vectors of all the user's k nearest neighbors $\mathbf{u_n}$, which captures the collaborative purchase pattern.

These two components are balanced by a hyperparameter α to make the final prediction:

$$\mathbf{P} = \alpha \cdot \mathbf{u_t} + (1 - \alpha) \cdot \mathbf{u_n}$$

The user vector representations are created in two steps. First, the basket history is divided into equally sized groups. The vectors are multiplied by a weight factor $\mathbf{r}_b$ within each group. These weights are hierarchical and based on a time-decayed ratio, which allows the algorithm to process baskets of variable length and means that higher significance is given to more recent purchases.

The second step is to apply a different time-decayed weight $\mathbf{r}_g$ across groups. Using two separate weights makes the model capable of capturing how a user's interests change over time [13]. The final user vector $\mathbf{u}$ is then obtained by averaging over the group vectors.

3 Related work

Research in the NBR field has evolved significantly, adopting various innovative methodologies. These approaches can broadly be divided into frequency-based, neighbor-based, and deep learning techniques [15].

Frequency-based methodologies, such as the Top-n frequent (TopFreq) and the Personalized Top-n frequent (PersonTopFreq) methods, are commonly employed in NBR [12,15]. The former approach recommends items based on global popularity, while the latter focuses on individual user behavior, enhancing personalization in the recommendation system [15].

Neighbor-based techniques represent another facet of the NBR field, with the TIFU-KNN [12] and UP-CF@r [9] methods leading the way. TIFU-KNN captures essential PIF signals to accommodate repeat and collaborative purchase patterns. In contrast, UP-CF@r combines user-wise popularity, customer loyalty, and recency information to accommodate dynamic user preferences [9].

A significant portion of NBR research has also incorporated deep learning techniques, as exemplified by models like Sets2Sets [11] and DREAM [26]. The Sets2Sets approach uses an encoder-decoder structure with a set-based attention mechanism to predict subsequent sets of items. DREAM uses an RNN to capture sequential features among baskets and generate dynamic user representations [26].

The DNNTSP methodology [27] presents an interesting alternative by constructing co-occurrence graphs at the set level, enabling a more nuanced understanding of element relationships. It utilizes graph convolutions and an attention-based module to learn temporal dependencies, thereby enhancing prediction performance [15, 27].

Lastly, the Contrastive Learning Model (CLEA) [19] employs a Gumbel Softmax and a denoising generator to categorize the items in historical baskets. Its GRU-based context encoder and two-stage anchor-guided contrastive learning process facilitate relevance learning without requiring item-level supervision, thus improving the model's effectiveness [15, 19].

4 Methodology

4.1 Replication of Original Results

The execution of our reproducibility study was significantly simplified by the fact that the authors of TIFU-KNN made their code publicly available [2]. The documentation provided was decently comprehensive, facilitating a seamless initiation and execution of experiments without significant obstacles. Notably, the dataset was split in the same manner as in the original study, with the first 72 percent used for training and the remainder allocated for validation and testing.

Nevertheless, the codebase presented opportunities for enhancement. It needed more clarity and harbored a bug that inadvertently overlooked the first entry in each dataset. Our aim was not to correct shortcomings; instead, these points served as an invitation to refine the existing foundation and elevate the overall code quality, augment its readability, and strengthen its maintainability.

In order to implement these improvements, we restructured the code [3]. We partitioned it into autonomous modules, each accountable for a designated functionality. This modularity fosters an intuitive comprehension of the codebase, facilitates the isolation and resolution of bugs, and streamlines the integration of novel features. Further amendments were made to enhance the readability of the code: we adopted consistent naming conventions, inserted extensive annotations, and ensured a coherent code style throughout the repository. These alterations collectively render the codebase more accessible and beneficial for subsequent research endeavors.

4.2 Model Generalizability Assessment

The original paper tests the model on multiple datasets from the shopping domain. We expand the scope of these tests with new data from a different field: music recommendation. User behavior may be different in this new field, so experiments on these datasets will give a deeper understanding of the algorithm's

[2] https://github.com/HaojiHu/TIFUKNN

[3] https://github.com/SebastiaanJohn/knn-nbr-analysis

capabilities and might reveal cross-domain patterns that are not apparent in the single shopping domain.

However, music datasets are differently structured from those used for NBR: they do not have “baskets”, but might contain timestamps of when users listen to music. Additionally, some items only interact with a small number of users. In order to adapt these data to NBR, we pre-process them in two different ways. First, we ignore items used by less than a certain amount of users. This results in fewer items in the model and higher efficiency. We used a threshold of 250 for LFM-1k and 100 for LFM-1b. Second, to simulate baskets, we group users' actions into one month time intervals. The model predicts what the user will listen to in future time intervals. Detailed statistics are shown in Table 1.

We use the following datasets:

– LFM-1k [6]: A last.fm dataset that contains anonymized listening histories of approximately 1.000 users with 1.000.000 lines of user data of listening events. The dataset provides valuable insights into user preferences and listening patterns, making it a suitable resource for investigating music recommendations in NBR. Songs are considered as items.
– LFM-1b [22]: Another last.fm dataset, but with 1 billion lines. However, due to the dataset's size, computational limitations made it unfeasible to parse the whole dataset. We therefore ran our experiments with 20.000.000 randomly sampled lines from the first 100.000.000 entries. Artists are considered as items for efficiency.

To find the optimal parameters, we tested the following values in a grid search:

$k = \{300, 600\}$
$m = \{3, 7, 11\}$
$r_b = \{0.1, 0.5, 0.9\}$
$r_g = \{0.1, 0.5, 0.9\}$
$\alpha = \{0.3, 0.5, 0.7, 0.9\}$

4.3 Experimenting with Alternative Distance Metrics

The original paper by Hu et al. [12] utilized Euclidean distance to compute user similarity from their representations, where a smaller distance indicates higher similarity. The authors suggest exploring alternative similarity functions but do not do so themselves. Our reproducibility study extends the original research by experimenting with Cosine and Manhattan distances alongside the original Euclidean distance to evaluate their impact on the kNN algorithm performance.

Cosine distance, defined as $\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$, measures the (cosine of the) angle between two vectors. Because of its focus on an angle rather than vector magnitude, this measurement is suitable for high-dimensional spaces such as user-item interactions.

Table 1. Statistic information of all used datasets after pre-processing. Basket size shows the average basket size, Baskets/User shows the average amount of baskets per user, and Repetition shows the average count of items bought or listened to by the same user.

Dataset	Items	Users	Basket Size	Baskets/User	Repetition
ValuedShopper	7,907	10,000	8.71	56.85	2.22
Instacart	8,000	19,935	8.97	7.97	2.24
Dunnhumby	4,997	36,241	7.33	7.99	1.36
TaFeng	12,062	13,949	6.27	5.69	1.11
LFM-1k	12,589	990	136.92	26.53	5.2
LFM-1b	21,199	9,346	50.71	32.10	10.18

Manhattan distance, defined as $\sum_{i=1}^{n} |A_i - B_i|$, calculates the sum of the absolute differences of coordinates. It is particularly relevant, and often more effective than Euclidean distance, in high-dimensional spaces [1], due to the curse of dimensionality [24].

4.4 Incorporating Density-Based Clustering: DBSCAN and HDBSCAN

Hu et al. [12] utilized kNN for next-basket recommendations by leveraging PIF information. However, kNN, being a supervised learning algorithm, may not always capture natural groupings in data.

As an extension, our study integrates the density-based clustering algorithms DBSCAN [8] and HDBSCAN [17, 18] as alternatives to kNN. DBSCAN groups data points based on density, defined by two parameters: ϵ (maximum distance between samples in a cluster) and τ (minimum number of samples for a core point). It can detect clusters of varying shapes and is robust to noise.

HDBSCAN, an enhancement of DBSCAN, performs hierarchical clustering by running DBSCAN over a range of ϵ values and combining the results to find stable clusters across scales. Like DBSCAN, it uses τ to determine whether a point is a core point. In addition, HDBSCAN introduces a new parameter μ, which allows the algorithm to define the smallest size grouping that should be considered a cluster.

Our approach uses DBSCAN and HDBSCAN to cluster customers based on the density of their history vectors, which contain PIF information. The algorithms precompute cluster indices and iterate through test customer IDs while they cluster labels. For each test customer, they retrieve the history vectors and initialize an array for the nearest neighbors within the cluster.

If a test customer is part of a valid cluster, the algorithm collects history vectors of the training customers from the same cluster as the nearest neighbors. Otherwise, it uses the test customer's history vectors. The final prediction follows the methodology of the original paper.

To identify the optimal parameters for DBSCAN and HDBSCAN, we performed a grid search for ϵ, τ, and μ on the validation set using the following values:

$$\epsilon = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$$
$$\tau = \{1, 2, 3, 4, 5, 10, 15, 20, 25, 30\}$$
$$\mu = \{2, 3, 4, 5, 10, 15, 20, 25, 30\}$$

Table 2 provides a breakdown of the optimal parameters identified for each dataset. During this search process, all remaining parameters were held constant using the same values as [12].

Table 2. Optimal parameters for DBSCAN and HDBSCAN for each dataset as determined by grid search. The DBSCAN parameters are the maximum distance between samples in a cluster (ϵ) and the minimum number of samples for a core point (τ). The HDBSCAN parameters are the minimum number of samples for a core point (τ) and the minimum cluster size (μ), the smallest size grouping that should be considered a cluster.

Dataset	**DBSCAN** (ϵ, τ)	**HDBSCAN** (τ, μ)
ValuedShopper	0.5, 2	1, 3
Instacart	0.9, 2	1, 2
Dunnhumby	0.9, 2	1, 3
TaFeng	0.4, 1	1, 2

5 Results

5.1 Replication

Table 3 presents the original and reproduced results of the TIFU-KNN algorithm using identical datasets, hyperparameters, and metrics as [12]. The replicated outcomes are largely consistent with the original findings, albeit with minor variations. For instance, in the ValuedShopper dataset, there is a marginal increase in all metrics, the highest being 2.5% in Recall@10. In contrast, the Recall@10 and Recall@20 slightly decrease for the Instacart dataset, while NDCG@10 remains almost unchanged. The Dunnhumby dataset shows a similar trend, with a slight decrease in Recall@10 and Recall@20 and a negligible increase in NDCG@10. The TaFeng dataset presents mixed results, with a notable increase of 4.6% in Recall@20, while other metrics exhibit minor fluctuations.

5.2 Model Generalizability Assessment

Table 4 shows the results of our experiments on different datasets with various hyperparameters.

Table 3. Side-by-side comparison of the results originally reported and those obtained in our replication attempt using identical hyperparameters. 'Metric' identifies the evaluation measure, 'Original' and 'Reproduced' show the respective results, and 'Difference' gives the percentual change between them. A negative 'Difference' indicates a decrease in the replicated results.

Dataset	Metric	Original	Reproduced	Difference (%)
ValuedShopper	Recall@10	0.2162	0.2216	2.5
	Recall@20	0.3028	0.3036	0.3
	NDCG@10	0.2171	0.2217	2.1
	NDCG@20	0.2589	0.2617	1.1
Instacart	Recall@10	0.3952	0.3814	-3.5
	Recall@20	0.4875	0.4857	-0.4
	NDCG@10	0.3825	0.3831	0.2
	NDCG@20	0.4384	0.4382	-0.1
Dunnhumby	Recall@10	0.2087	0.2073	-0.7
	Recall@20	0.2692	0.2675	-0.6
	NDCG@10	0.1983	0.1986	0.2
	NDCG@20	0.2302	0.2270	-1.4
TaFeng	Recall@10	0.1301	0.1291	-0.8
	Recall@20	0.1810	0.1893	4.6
	NDCG@10	0.1011	0.0983	-2.8
	NDCG@20	0.1206	0.1211	0.4

Table 4. Results of new datasets after hyperparameter tuning with best parameters based on recall@10. Parameter k represents the number of nearest neighbors, r_b represents the decay rate within a group, r_g represents the decay rate between groups, α represents the weight of the group, and m is the group size.

Dataset	k	m	r_b	r_g	α	Recall@10	NDCG@10
LFM-1k	300	3	0.5	0.9	0.5	0.0911	0.1327
LFM-1b	300	3	0.9	0.5	0.3	0.2164	0.2099

Even though there are no baseline models to compare with, we can see that TIFU-KNN can perform satisfiably with data from the field of music recommendation. The metrics are lower than for the shopping datasets, but this is expected, since there are more items in these datasets.

5.3 Alternative Distance Metrics

Table 5 compares of three distance metrics, namely Euclidean, Cosine, and Manhattan distances, in the context of kNN algorithm performance on various datasets. The datasets used include ValuedShopper, Instacart, Dunnhumby, and TaFeng. The evaluation metrics employed are Recall@10 and NDCG@10. The results indicate that the Cosine distance metric outperforms the Euclidean and Manhattan metrics in terms of Recall@10 and NDCG@10 for the ValuedShopper, Instacart, and TaFeng datasets. However, for the Dunnhumby dataset, the Manhattan distance metric slightly surpasses the other two metrics.

Table 5. Comparison of different distance metrics across the datasets using the same hyperparameters as the original paper. 'Metric' identifies the evaluation measure, while 'Euclidean', 'Cosine', and 'Manhattan' columns show the results obtained using the respective distance metrics. Numbers in **bold** represent the best-achieved results.

Dataset	**Metric**	**Euclidean**	**Cosine**	**Manhattan**
ValuedShopper	Recall@10	0.2216	**0.2232**	0.2198
	NDCG@10	0.2217	**0.2230**	0.2212
Instacart	Recall@10	0.3814	**0.3830**	0.3816
	NDCG@10	0.3831	**0.3841**	0.3828
Dunnhumby	Recall@10	0.2073	0.2063	**0.2075**
	NDCG@10	0.1986	0.1916	**0.1988**
TaFeng	Recall@10	0.1291	**0.1312**	0.1287
	NDCG@10	0.0983	**0.1003**	0.0987

5.4 Density-Based Clustering

Table 6 compares DBSCAN, HDBSCAN, and kNN performance on the datasets. The results indicate that kNN consistently outperforms DBSCAN and HDBSCAN on almost all metrics and datasets. For instance, in the ValuedShopper dataset, kNN achieves the highest scores in Recall@10, Recall@20, NDCG@10, and NDCG@20. Similar trends are observed in the Instacart and Dunnhumby datasets. However, in the TaFeng dataset, DBSCAN slightly outperforms kNN in NDCG@10, whereas kNN remains superior for the other metrics.

6 Discussion

Table 3 demonstrates the robustness of the original implementation of the TIFU-KNN algorithm. Our replication study reveals minor discrepancies between the original and replicated results, which can be attributed to factors such as random initialization and computational environment. Interestingly, the algorithm shows a significant increase in Recall@20 on the largest dataset, TaFeng, indicating its sensitivity to dataset characteristics.

Comparing the performance of the kNN algorithm using different distance metrics (Table 5) indicates that the choice of distance metric can have a small

Table 6. Comparison of DBSCAN, HDBSCAN, and kNN performance on the ValuedShopper, Instacart, Dunnhumby, and TaFeng datasets, evaluated using Recall@10, Recall@20, NDCG@10, and NDCG@20 metrics. Numbers in **bold** represent the best-achieved results.

Dataset	Metric	DBSCAN	HDBSCAN	kNN
ValuedShopper	Recall@10	0.2198	0.2174	**0.2216**
	Recall@20	0.2742	0.2647	**0.3036**
	NDCG@10	0.2208	0.2195	**0.2217**
	NDCG@20	0.2385	0.2318	**0.2617**
Instacart	Recall@10	0.3797	0.3787	**0.3814**
	Recall@20	0.4828	0.4798	**0.4857**
	NDCG@10	0.3791	0.3785	**0.3831**
	NDCG@20	0.4378	0.4342	**0.4382**
Dunnhumby	Recall@10	0.1964	0.1903	**0.2073**
	Recall@20	0.2623	0.2417	**0.2675**
	NDCG@10	0.1785	0.1790	**0.1986**
	NDCG@20	0.2170	0.2045	**0.2270**
TaFeng	Recall@10	0.1287	0.0974	**0.1291**
	Recall@20	0.1832	0.1251	**0.1893**
	NDCG@10	**0.0986**	0.0839	0.0983
	NDCG@20	0.1194	0.0958	**0.1211**

effect on the algorithm's performance across different datasets. The Cosine metric outperforms the other distances on ValuedShopper, Instacart, and TaFeng, which could be attributed to its properties being well-suited for the underlying distribution and structure of these datasets. On the other hand, the Manhattan metric exhibits better performance on the Dunnhumby dataset. Although these findings could indicate important dataset characteristics in metric selection, the performance differences are insignificant enough to alter the model's behavior significantly.

Table 4 shows that the model is suitable for the task of music recommendation, where patterns are similar to basket recommendation. Especially the LFM-1b dataset, with artists considered as items, yielded surprisingly strong results. These results demonstrate the capability of the TIFU-KNN model on data where users display more repetitive customer behavior, as can be understood from Table 1. Even though the datasets are very different, the effects of changing parameters shows the same trends. These parameters may be tuned on an even lower scale to further explore this observation. From Table 1 and 3, we can see that the metrics on larger datasets are lower than on the smaller datasets, but this is expected as there is more variety of items to make predictions about.

Table 6 reveals that the kNN algorithm is more effective at capturing PIF information than DBSCAN and HDBSCAN. This could mean that the relevant data patterns for the NBR task align better with a simple nearest neighbors approach than with a density-based clustering approach. Surprisingly, HDBSCAN consistently underperforms compared to DBSCAN across all metrics and datasets. Its hierarchical approach may not be optimal for this problem domain, potentially because it introduces unnecessary complexity without enhancing performance. DBSCAN, however, slightly outperforms kNN in NDCG@10 for the TaFeng dataset, presumably due to dataset-specific characteristics. Finally, it is worth mentioning that, aside from its robust performance, kNN has a superior computational speed. This is a substantial practical advantage in real-world applications and leads us to believe kNN is the overall better clustering approach for NBR.

7 Conclusion

Through a reproducibility study and expansions upon the original method, we have attempted to explore the robustness and generalizability of the TIFU-KNN algorithm [12], a nearest neighbors-based model for the Next Basket Recommendation task. The original paper's authors provide a somewhat undocumented but functioning codebase, which aids in both reproducibility and fairness assessment. Our results show that the model can perform well in different scenarios and on datasets from various domains. The model also does not seem to be significantly impacted by changes in the inner functioning of the KNN method. Furthermore, a comparison with dense clustering algorithms indicates that simple nearest-neighbor models are surprisingly effective and that selecting the correct model for the NBR task is an essential factor for the final prediction quality.

Our study has a few limitations. Firstly, we kept the same hyperparameters as the original study when varying the distance metrics. Hyperparameter choice may influence these metrics, and performance differences may appear under certain scenarios we did not consider. Another limitation of our study is how we evaluate model performance. We heavily emphasized Recall and NDCG, but other evaluation metrics might rank models differently. Future work must investigate the models with other evaluation metrics to better understand their performance. Moreover, we could only explore a small percentage of the LFM-1b dataset due to computational limitations. Lastly, we suggest future research to look more into the properties of PIF to see how this information can be leveraged in different kinds of models.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Database Theory—ICDT 2001: 8th International Conference London, UK, January 4–6, 2001 Proceedings 8. pp. 420–434. Springer (2001)
2. Aggarwal, C.C., et al.: Recommender systems, vol. 1. Springer (2016)
3. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician **46**(3), 175–185 (1992)

4. Anderson, A., Kumar, R., Tomkins, A., Vassilvitskii, S.: The dynamics of repeat consumption. In: Proceedings of the 23rd international conference on World wide web. pp. 419–430 (2014)
5. Ariannezhad, M., Jullien, S., Li, M., Fang, M., Schelter, S., de Rijke, M.: Recanet: A repeat consumption-aware neural network for next basket recommendation in grocery shopping. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1240–1250 (2022)
6. Celma, O.: Music Recommendation and Discovery in the Long Tail. Springer (2010)
7. Chen, X., Chen, H., Xu, H., Zhang, Y., Cao, Y., Qin, Z., Zha, H.: Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 765–774 (2019)
8. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96, pp. 226–231 (1996)
9. Faggioli, G., Polato, M., Aiolli, F.: Recency aware collaborative filtering for next basket recommendation. In: Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization. pp. 80–87 (2020)
10. Ferrari Dacrema, M., Cremonesi, P., Jannach, D.: Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM conference on recommender systems. pp. 101–109 (2019)
11. Hu, H., He, X.: Sets2sets: Learning from sequential sets with neural networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1491–1499 (2019)
12. Hu, H., He, X., Gao, J., Zhang, Z.L.: Modeling personalized item frequency information for next-basket recommendation (2020)
13. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 447–456 (2009)
14. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 1419–1428 (2017)
15. Li, M., Jullien, S., Ariannezhad, M., de Rijke, M.: A next basket recommendation reality check. ACM Transactions on Information Systems **41**(4), 1–29 (2023)
16. Liu, Y., Lee, H., Achananuparp, P., Lim, E.P., Cheng, T.L., Lin, S.D.: Characterizing and predicting repeat food consumption behavior for just-in-time interventions. In: Proceedings of the 9th International Conference on Digital Public Health. pp. 11–20 (2019)
17. McInnes, L., Healy, J.: Accelerated hierarchical density based clustering. In: Data Mining Workshops (ICDMW), 2017 IEEE International Conference on. pp. 33–42. IEEE (2017)
18. McInnes, L., Healy, J., Astels, S.: hdbscan: Hierarchical density based clustering. The Journal of Open Source Software **2**(11), 205 (2017)
19. Qin, Y., Wang, P., Li, C.: The world is binary: Contrastive learning for denoising next basket recommendation. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. pp. 859–868 (2021)
20. Raeder, T., Chawla, N.V.: Market basket analysis with networks. Social network analysis and mining **1**, 97–113 (2011)

21. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 811–820 (2010)
22. S, M.: The lfm-1b dataset for music retrieval and recommendation. International Journal of Multimedia Information Retrieval p. 103–110 (2016)
23. Siegelmann, H.T., Sontag, E.D.: On the computational power of neural nets. In: Proceedings of the fifth annual workshop on Computational learning theory. pp. 440–449 (1992)
24. Weber, R., Schek, H.J., Blott, S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: VLDB. vol. 98, pp. 194–205 (1998)
25. Ying, H., Zhuang, F., Zhang, F., Liu, Y., Xu, G., Xie, X., Xiong, H., Wu, J.: Sequential recommender system based on hierarchical attention network. In: IJCAI International Joint Conference on Artificial Intelligence (2018)
26. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 729–732 (2016)
27. Yu, L., Sun, L., Du, B., Liu, C., Xiong, H., Lv, W.: Predicting temporal sets with deep neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1083–1091 (2020)

Table 7. Individual contributions to the project.

Member	**Extension Concept**	**Implementation**	**Report**	**Poster/Presentation**
S. Dijkstra	30%	70%	40%	10%
G. Keskin	50%	30%	20%	10%
B. Veenman	20%	–%	40%	80%
Total	100%	100%	100%	100%