electronics

MDPI

Article

Temple Recommendation Engine for Route Planning Based on TPS Clustering CNN Method

Dasarada Rajagopalan Thirupurasundari [1], Annadurai Hemlathadhevi [2], Amit Kumar Gupta [3], Ruchi Rani Garg [4] and Mangal Sain [5,*]

[1] Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research, Selaiyur, Chennai 600073, India
[2] Department of Computer Science and Engineering, Panimalar Engineering College, Poonamallee, Chennai 600123, India
[3] Department of Computer Applications, KIET Group of Institutions, Delhi-NCR, Ghaziabad 201206, India
[4] Department of Applied Science, Meerut Institute of Engineering & Technology, NH-58 Delhi-Roorkee Highway, Bag Pat Bypass Road, Meerut 250005, India
[5] Division of Computer and Information Engineering, Dongseo University, Busan 47011, Korea
* Correspondence: mangalsain1@gmail.com

Abstract: There are no restrictions on religious or cultural practices in India. India's temples are becoming an ideal platform for Hindu groups to express their ideals in a global context. For the sake of devotees, temples now require widespread accessibility and participation by a wide range of individuals on major holidays. A pilgrim may be unable to determine which site to visit, or where to stay, due to a variety of considerations such as cost, location, and the interests of each individual user. A user's preferences are taken into consideration when a personalized recommendation list is generated. A large number of systems use Collaborative Filtering to produce user recommendations. In order to generate user-specific recommendations, this system uses a filtering method dubbed the "hybrid approach." The Proposed OTPS Cluster technique is used to determine TPS (Time, Place, and Service). Users' interests and TPA recommendations are taken into account. Users can forecast the location of the temple based on the temple's history. Collaborative Filtering and Material-Based Filtering were used to propose sites based on comparable users and content, respectively. Testing shows that the algorithm is capable of solving difficulties in standard tour routing and providing a temple visit route that is tailored to each individual's preferences. This study makes use of data from the South Indian city of Temple in the form of temples.

Keywords: recommendation systems; place recommendation; time optimization; TPS clustering

Citation: Thirupurasundari, D.R.; Hemlathadhevi, A.; Gupta, A.K.; Garg, R.R.; Sain, M. Temple Recommendation Engine for Route Planning Based on TPS Clustering CNN Method. *Electronics* **2022**, *11*, 2630. https://doi.org/10.3390/electronics11162630

Academic Editors: Antonio Rios Navarro and Gianvito Urgese

Received: 14 July 2022
Accepted: 17 August 2022
Published: 22 August 2022

1. Introduction

Since the beginning of time, Hinduism has been practiced as a way of life. The macrocosm (the universe) and the microcosm (the individual) are both represented in Hindu temple construction, which is linked to astronomy and sacred geometry. To ensure the spiritual and social well-being of the community, temples are always located at the heart of Hindu towns. Recommendation systems, or "recommenders systems," are a major research area. By analyzing the habits of its customers, it recommends products that are likely to be of value. In general, recommendation lists are generated based on user preferences, item features, user–item past interactions, and some other additional information such as temporal (e.g., sequence-aware recommender) and spatial (e.g., POI recommender) data. Recommendation models are mainly categorized into collaborative filtering, the content-based recommender system, and the hybrid recommender system based on the types of input data.

The following are the steps of recommendation systems [1]. To begin, people converse and discuss their preferences via the user interface they are using. Second, the consumers' implicit and explicit data is collected. Recommendation engines are then used to make recommendations from the obtained data. Preprocessing, learning, evaluating, and predicting are all components of the recommendation engine [2]. Without a recommendation system, users would spend a lot of time choosing the right destination from the many routes available. This can frustrate users and discourage them from using the web search any longer. Recommendation systems are defined as the techniques used to predict the rating one individual will give to an item or social entity [3]. Finally, the most accurate predictions are sent out to the customers as shown in Figure 1.

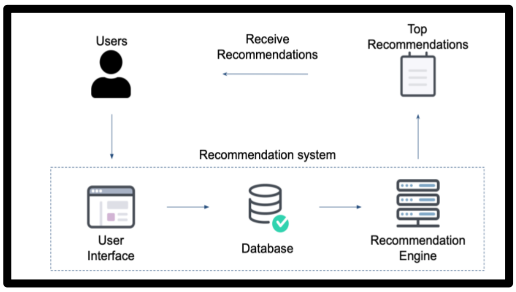

Figure 1. Recommendation system steps.

The recommendation system (RS) has gained its popularity in many domains such as movies, books, music, search queries, and social tags due to its capability of predicting future preference with a set of items for users. Collaborative filtering is a technique or method of predicting user preferences and finding things that users may like based on information gathered from various other users with similar likes or preferences. It takes into account the fundamental fact that if MX and MY respond to certain items to a certain extent, they may have the same opinion on other items. Collaborative filtering tends to find similar users and recommend what they like. Rather than recommending the use of item features, this type of recommender system classifies users into similar cluster types and recommends each user based on their cluster preferences.

Planning a vacation to an unfamiliar location can be unnerving, especially for tourists with physical restrictions or language barriers, as there is little or no information on site. Websites provide information about points of interest (POI) based on ratings provided by other users. It may not suit everyone's tastes. The proposed Temple recommendation engine uses personal information to provide recommendations and attempts to obtain travel-related attributes such as user mentions, media attachments, etc. OTPS cluster technology is used to determine the TPS (Time, Place, and Service).

2. Literature Survey

This work uses clustering to find relevant routes for the user prior to route optimization so that route ranking can be accomplished. Recommender systems (RSs) were generally defined as expert systems which are used to recommend products or services to the users. Figure 2 portrays the working of a traditional recommender system. Objects are grouped together in clusters where they are more closely related to one another than they are to the objects in other groups [4]. Machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics are some of the most common applications of exploratory data mining and statistical data analysis. Timeliness is one of the most important challenges. Factors such as the very short duration of news articles, recency, popularity, trends, and the large number of news articles delivered per second are taken into account. Another major challenge in the news space is the very dynamic behavior of users. Personalization is a useful feature of NRS, as it provides information based on reader preferences and interests [5]. A single algorithm is not needed to perform cluster analysis; rather, it is a generic activity carried out by a range of algorithms, each with a somewhat different take on what constitutes a cluster and how to find them quickly [6]. The model-based filtering methodology describes how machine learning algorithms can be implemented for movie recommendation purposes, and how to predict unrated movie ratings and branch or sort movies based on viewer preferences. The CF algorithm is a very efficient technique for applying MRS [7]. It is common for clusters to be made up of groups with a little distance between each other, dense regions of data space, and certain statistical distributions. In this way, clustering can be treated as an optimization problem. A clustering algorithm and parameter settings should be chosen based on the specific data collection and the intended use of the resulting clustering data. The idea of social network data mining is similar to GPS trajectory data mining. In GPS trajectory data mining, the main applications include association rules, abnormal behavior, travel mode, and GPS trajectory recommendation [8]. Problems with recommender systems using AI and improvements to these systems using AI approaches such as fuzzy techniques, transfer learning, genetic algorithms, evolutionary algorithms, neural networks and deep learning, and active learning will be examined [9]. The Knowledge Base Graph Integration module extracts core entities from images, text and maps them as knowledge base entities. Then, they extract the subgraphs closely related to the central entity and transform the subgraphs into low-dimensional vectors to realize the subgraphs' embedding [10]. RS-based users' behavior and collaborative location and tracking [11] uses Twitter data to personalize place of interest (POI) recommendations. Their model takes into account tweets related to travel. User preferences for the different categories are first identified [12]. In social network trajectory data mining, applications mainly include location recommendation, path recommendation, and behavior preference recommendation [13]. As a process of knowledge discovery or an interactive multiobjective optimization involving trial and error, cluster analysis is not often seen as an automatic operation [14]. A latent factor (LF)-based approach becomes highly popular when implementing a recommender system. However, current LF models mostly adopt single distance-oriented Loss, such as an L2 norm-oriented one, which ignores target data's characteristics described by other metrics, such as an L1 norm-oriented one [15]. A novel end-to-end framework called KBRD, which stands for Knowledge-Based Recommender Dialog System, integrates the recommender system and the dialog generation system. The dialog system can enhance the performance of the recommendation system by introducing knowledge-grounded information about users' preferences, and the recommender system can improve that of the dialog generation system by providing recommendation-aware vocabulary bias [16]. Moreover, as a representation of global information, the Knowledge Graph itself contains rich semantic information. The Knowledge Graph can increase the interpretability of recommendation results as well as improve the accuracy of recommendations and the diversity of recommendation results [17]. Sometimes it is necessary to tweak things like data preprocessing and model parameters to get the results you want. A survey of collaborative filtering (CF)-based social recommender systems provide a brief overview over the task of recommender systems and traditional approaches that do not use social network information, but a recommender system that suggests information to users based on their current mood or emotion [18]. In this model, the facial expression recognition technique uses a convolutional neural network (CNN) to extract features from the facial image and then artificial neural networks [19].

It is a difficult job for recommender systems to provide suggestions based on user preferences. Based on the Travel Recommender and POI Recommender, the location recommendation and social media recommendation algorithms are implemented.

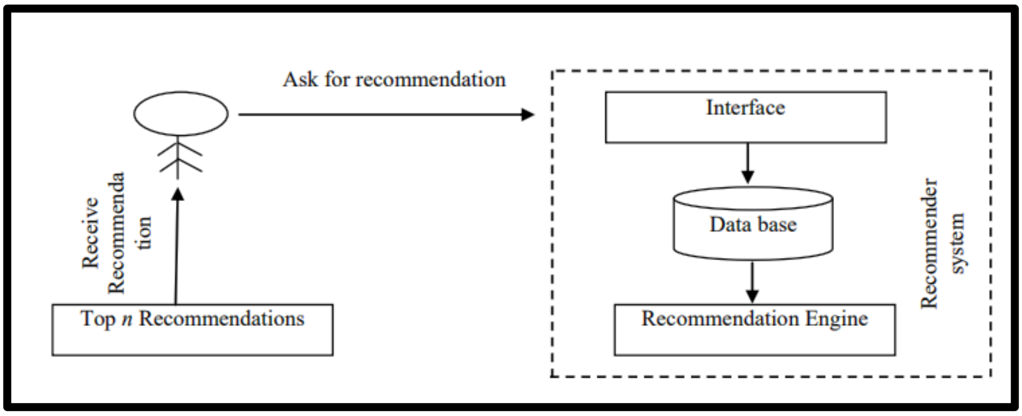

Figure 2. Existing Recommender System.

3. Proposed System

Data flow in the news recommendation system will be discussed in this section. Recommendations should arrive in real time, which is an important consideration. The system should be able to capture and recommend fresh news articles to the user as soon as they enter the system [20]. The information is passed along from one component to the next until it reaches the user in the form of a list of suggested readings. The process go in the following order through the system as shown in Figure 3:

1. Scraper: Access content from a variety of news sources. Initiates database entry of articles.
2. Filterer: Identifies articles that are not duplicates of existing articles and matches the standards, such as size and language, to be classed as articles legitimate enough to be processed. Filtered news is marked as such and is saved in a separate database from the scraped material.
3. Classifier: Filtered data is seen, and articles are assigned NER and LDA classifications. The LDA classifications are applied to the filtered articles, and the items are then tagged as classified.
4. Node Manager: A VMM-Tree is constructed from the most recently processed articles' cumulative and time-decayed subject ratings. The tree is serialized and stored in the database in this manner.
5. Recommender: For each user, the system finds a subject set that it believes is relevant, and then recommends articles from those topics using a rating algorithm when they interact with the system.

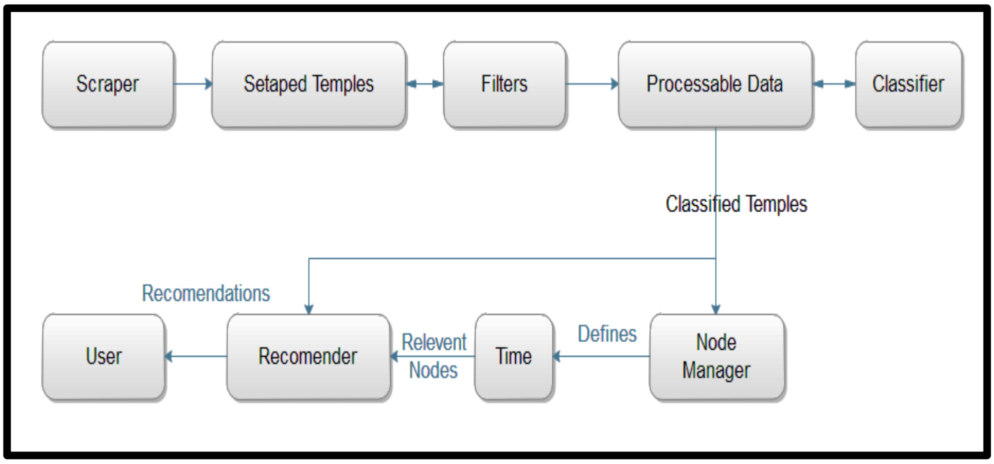

Figure 3. Proposed Recommendation System.

Our embedding layers Figure 4 include dense embedding, sparse embedding, and positional embedding. Positional embedding is used for attention modules, while dense embedding is used for numerical data types [21]. There are three inputs, two dense and one sparse, in the dataset. Time and service are dense inputs, but place is sparse. Due to their differences in characteristics, these two inputs are each treated differently. Using MLP, for example, the dense layer is embedded, while embedding lookup transforms sparse features into numerical vectors. Sequential hyperinteraction is then applied to each dense and sparse input to retrieve the historical pattern. Lastly, nonsequential hyperinteraction can be used to learn more about user–item interaction than can be learned from sequential data [22]. The two losses, regression and classification, are then calculated.

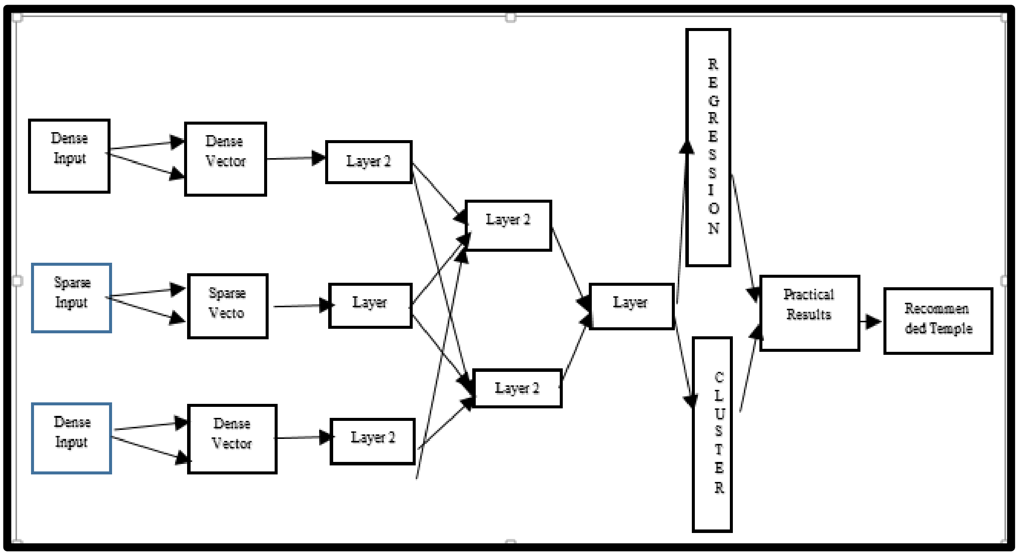

Figure 4. Proposed embedding layers.

3.1. Location Recommendation System

The location-based social network was used to predict the recommended destinations for the user in the proposed recommendation model. The socially relevant trust walker algorithm uses the existing rating scores for related location categories to determine the rating score for each location [23]. As location-based technologies (such as GPS) become available, social networks use location data to operate in different ways. For example, users can share their current location on websites, upload geotagged photos to social media platforms, and write reviews about places they have visited. Location is therefore considered one of the key elements of user context, and in-depth knowledge of user behavior and interests can be derived from user location data [24].

Locations having a higher relevance score are recommended to users after calculating the location category's rating score. In order to make recommendations more useful, a social trust walker algorithm was implemented. Figure 5 depicts the proposed recommendation model. There are two primary components in the proposed location recommender system: user interface and ratings prediction. Rating prediction is based on the user inputs from various locations and the reviews received.

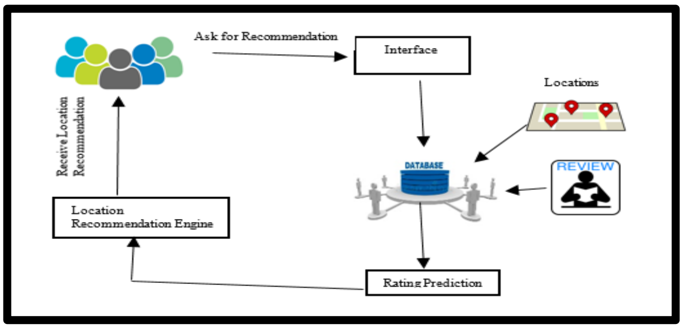

Figure 5. Block Diagram for Location Recommender System.

- *Location Mining*:

In general, a travel itinerary is a series of attractions in a specific order. For example, for a left-to-right order, the leftmost point in the sequence is considered the start point and the rightmost point is considered the end point, the last stop.

Route extraction is the first step in the recommendation approach, which handles the process of retrieving data from data sources and recombining that data to generate routes for further data processing or storage. It generates route sets by extracting more features such as geolocation, route information, etc., from the travel dataset beyond individual attractions.

- *Powerful route generation*:

Given a set of frequent roots, R, the requirement when a root can be added to the robust roots is to find all nonempty subsets that satisfy the minimum condition. Similar to frequent root generation, we also need to look for efficient ways to generate robust roots. This generates robust routes by merging two routes that share the same prefix in the consequent part of the rule.

In this method, we applied user-based collaborative filtering to calculate similarity scores between users and local experts. It provided users with location recommendations around a given geographical location based on their preferences. A user's location history and local expert social opinion may share similar interests. Compared to user profiles and user location histories, user paths contain a richer set of information, such as the order of movement between locations and the time spent at each location. Therefore, route data can be used to more accurately estimate user preferences.

3.2. Collaborative Filtering Architecture

After doing an evaluation of the algorithm's effectiveness, we observed that it was 78 percent successful. Given that we do not know anything about the objects or their users, we make the assumption that persons who give the same item equivalent ratings are also similar. Collaboration filtering is a strategy used by recommendation systems to make their recommendations [25,26]. Ant colony optimization performed well in solving the traveling salesman problem (TSP), is prone to local optima, and has limited convergence speed. To solve this problem, multi-colony ant colony optimization based on a generalized Jaccard similarity recommendation strategy [27].

Ratings and reviews of tourist places are based on collaborative filtering. Users rate tourist destinations for various purposes. Users specify reviews of places and hotels they have visited and are used to find ratings for specific places and hotels. Uses collaborative filtering data mining techniques using three data mining algorithms such as pattern matching, clustering, and association [28,29]. Collaborative filtering is a generic term that refers to the process of filtering information or patterns using techniques that necessitate collaboration across diverse agents, views, data sources, and other factors. For a variety of reasons, including the ability to analyze large datasets, collaborative filtering is employed. Collaborative filtering outperforms content-based filtering in terms of accuracy [30]. With the use of collaborative filtering architecture Figure 6, we can divide the suggestion process into three parts:

- User assessments are taken into account while creating a visual representation of temple information.
- We may then use our own collaborative filtering method to determine the degree of similarity between tourists based on their previous visit history [9] and other relevant data.
- The creation of tourism attraction recommendations. Recommendations for top attractions, restaurants, and hotels are generated by comparing your tastes with those of other visitors.

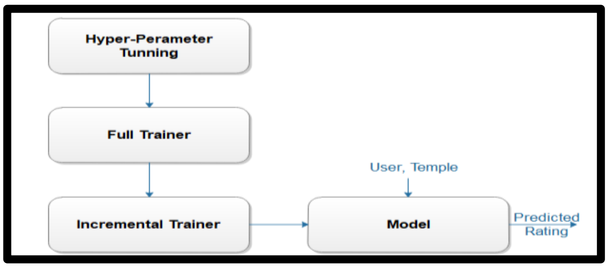

Figure 6. Collaborative filtering architecture.

To put it another way, we refer to two things as similar if one user gave them the same rating. A weighted average of reviews from this user's most X similar items is then used to determine how likely a user is to purchase an item. Item-based CF has the advantage of being more stable than human-based CF in that the ratings for a given item do not change dramatically over time [28]. When comparing two vectors, the concept of similarity states that this is equal to 1 if they are identical, and 0 if they are orthogonal. A number between 0 and 1, indicating the degree to which the two vectors resemble one another, is what we mean by the term "similarity".

Temple recommender systems predict AU (active user) preference for items in real time. The proposed system creates a model based on AU historical data in a tree-like data structure to reduce computational complexity. Recommendations are triggered when the item the AU is looking for is detected. If there is an item rating record in the AU history, the parameters are predicted based on tree structure. If there is no article history, the proposed system predicts the priority based on the similarity.

Collaborative filtering does not require known features on items or users, which is suitable for different sets of objects. If you want an advisor to recommend a new temple to someone who has just visited a temple that is not on the list, try adding the same to the recommendation spell when selecting for cooperation. Collaborative filtering is very commonly used in recommendations, but challenges in using it include: collaborative filtering can cause issues such as new items added to the list to cold start. They do not recommend them until someone reviews them. Lack of data can affect the quality of user-based recommendations and exacerbate the cold start issues mentioned. With a simple implementation, you will find that recommendations are usually already popular, and items in the long tail section can be skipped.

4. Results and Discussion

4.1. Time Recommendation System Algorithms

Personalized TPS recommendations are contrasted to other route-planning systems. Figure 7 Plots the evaluation on different method discussed in this section.

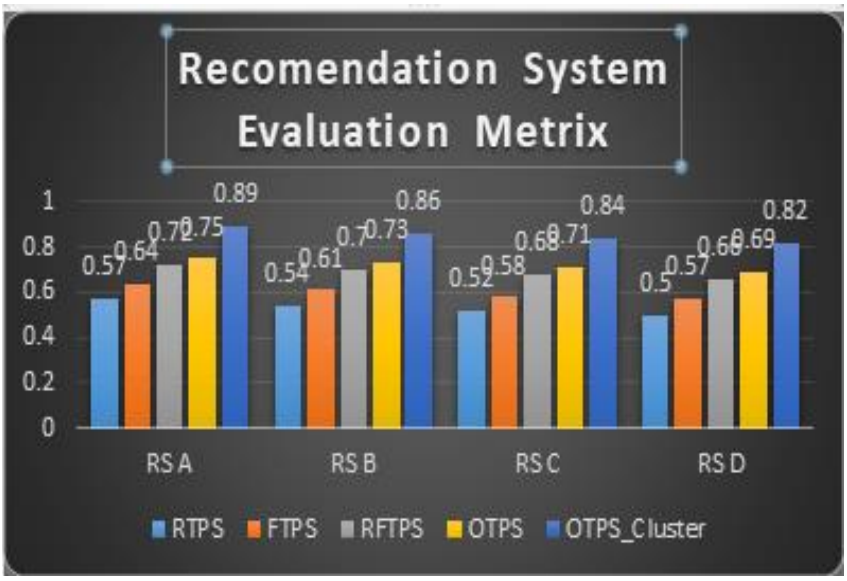

Figure 7. Plot for Evaluation on different method for the time recommendation system (RS).

The history of the trip information is extracted. The trips that have same origin/destination place are put together in chronological order. Then, we use an average on all durations for these trips whose start time is within the same time slot to represent the travel time of the two locations during the time slot. A temple's trip contains the information beginning at the entrance of the temple (TT) and lasting until encountering a finish-off event. The target of the data preparation is to derive the sample training data from original temple's trace. We put similar trips together to derive a sample observation.

Definition 1. *(temple trip): a temple trip TTr is a quaternion containing the following four items: start location (TTr.sl), start time (TTr.st), finish location (TTr.fl), and duration (TTr.du).*

The start location, the end location, and the start time of a temple are the three basic features of each trip. If two trips have basic features similar to each other, they are similar. We use the start region and end region to replace the points.

4.2. Random Time Planning System (RTPS)

This method constructs time schedule by randomly selecting five POIs.

We will use a linear recurrence relation as follows,

$$X_i = mX_{\mathrm{i}-1} + p \tag{1}$$

where m = 2 and p = 5 (m and n are chosen integer)

$$U_i = X_i / p \tag{2}$$

$$U_i = 2/5 = 0.4$$

4.3. Famous Time Planning System (FTPS)

However, this strategy does not allow us to prioritize or optimize the most popular time schedules. All users get the same POI recommendations calculated by Jaccard Similarity.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

4.4. Ranked Famous Time Planning System (RFTPS)

This technique proposes the most popular routes based on a user's time preferences calculated by Pearson correlation coefficient, but does not optimize for the user's time constraints.

$$r = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2} X \sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2}} \tag{4}$$

4.5. Optimized Time Planning System (OTPS)

A travel route is a sequence consisting of multiple travel POIs, denoted as R = {p1, p2, . . . , pN}, where pi is the temple location included in the route and N is the number of locations.

The travel time required by the user from $POI\ P*\ POI\ Px\ to\ POI\ Py\ POI\ Py$ can be defined as:

$$TTravel(px, py) = Dist(px, py)/speed$$

Route advice, including time ranking, is included in this algorithm. The following metrics are used to evaluate the above-mentioned route recommendation techniques.

- Rating prediction metrics
- Ranking recommendation matrix
- Rating prediction metrics

The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{5}$$

Ranking Recommendation Matrix

With the input of users' ratings on the temple locations, we would like to predict how the users would rate the temple locations so the users can get the recommendation based on the prediction.

Assume we have the ranking table of five people and five temples, and the ratings are integers ranging from 1 to 5, the matrix is provided by the Table 1.

Table 1. Rating matrix.

	Temple 1	**Temple 2**	**Temple 3**	**Temple 4**	**Temple 5**
person 1		5		3	2
person 2	4		3		4
person 3		3			
person 4	4		5	2	
person 5	3	5			4

Define a set of people (P), temples (D), R size of $\mid P \mid$, and $\mid D \mid$. The matrix $\mid P \mid * \mid D \mid$ includes all the ratings given by users.

Given with the input of two matrices A $(|P| * k)$ and $(|D| * k)$, it would generate the product result R.

$$R = A * QT = R\hat{} \tag{6}$$

Optimized Time Planning System Using Clustering (OTPS_Cluster)

In the outlined above, when using fuzzy clustering, each data point can be included in multiple clusters at the same time. Data points are assigned to groups in such a way that those belonging to the same group are as similar as feasible, and those belonging to separate groups are as dissimilar as possible, in order to do clustering or cluster analysis. Clusters are detected using similarity measurements, such as distance, connectedness, and intensity, which are used to identify them. Data or application specific similarity measures might be used.

Fuzzy C Mean Algorithm

Step 1: Initialize the data points into desired number of clusters randomly.

Table 2 below represents the values of the data points along with their membership (gamma) in each of the cluster.

Table 2. Values of the data points along with their membership.

Cluster	(1, 2)	(2, 4)	(3, 6)	(7, 8)	(7, 9)
1	0.6	0.5	0.2	0.1	0.4
2	0.3	0.4	0.5	0.9	0.7
3	0.5	0.3	0.4	0.6	0.1
4	0.4	0.8	0.3	0.7	0.9

Step 2: Find out the centroid.

The formula for finding out the centroid (V) is:

$$V\{ij\} = \frac{\left(\sum 1\_n \ \int_1^n £\{ik\}^m \ * x_k\right)}{\sum_{k=0}^{n}\left(\frac{n}{k}\right)x^k \ \int_1^n £\{ik\}^m} \tag{7}$$

Step 3: Find out the distance of each point from centroid.

Step 4: Updating membership values.

$$£ = \sum_{k=1}^{n}\left(\frac{n}{k}\right)d\_\frac{\frac{kj^2}{m}1}{j} \tag{8}$$

Step 5: Repeat the steps (2–4) until the constant values are obtained for the membership values.

Table 3 below represents the Evaluation on different method for the time recommendation system (RS).

Table 3. Evaluation on different method for the time recommendation system (RS).

Performance	RTPS	FTPS	RFTPS	OTPS	OTPS_Cluster
RS A	0.57	0.64	0.72	0.75	0.89
RS B	0.54	0.61	0.70	0.73	0.86
RS C	0.52	0.58	0.68	0.71	0.84
RS D	0.5	0.57	0.66	0.69	0.82

This allows the objects to belong to many clusters simultaneously with different degrees of membership. Fuzzy clustering methods using fuzzy clustering rather than hard clustering can be more intuitive in many cases. Membership degrees between 0 and 1 are assigned to objects on the boundary between many classes, rather than requiring them to fully belong to one of the classes. The hard partitioning's discrete character also makes it problematic for algorithms based on analytic functions, since these functions are not differentiable. This paper introduces the Map Reduce framework to address the scalability issue in fuzzy clustering.

Preprocessing Using MapReduce

Before you begin the clustering process, you need to clean up your data. Removed from the database are duplicates and nulls. A location ID with the user's name and rating is established after preprocessing. Algorithm 1 MapReduce is used to produce user rating pairs. After the program has been preprocessed, a separate file containing the location's name, users' names, and ratings is generated.

Algorithms 1: MapReduce

```
BEGIN
  procedure map (LongWritable key, Text value)
      String s=value.toString();
      a   -   Array of String type
      String a[]=s.split(“,”)
      If (a.length)
         t1.set(a[1]+”\t“)
         t2.set(a[0]+”,”+a[2])
      End if
      context.write(t1,t2);
END
```

5. Conclusions

In today’s online world, recommendation engines are essential to any company’s success. To produce appropriate recommendations in real time, a good recommendation system must have strong correlation capabilities that go beyond the product. If you are running an e-commerce business, recommender systems can be a strong asset. Moreover, as technology advances, their worth will only grow. Social media data is used to implement the POIs and route recommendation methodologies. The textual description also yields information about a user’s potential interest in travel topics, such as Time, Place, and Service (TPS). A Users preferences can be taken into consideration when a recommendation is given to other user. In order to optimize revenues and communicate with customers based on the information acquired based on their preferences, RS is critical in many enterprises and other sectors. The CNN approach, which has been developed by a number of academics, has shown promising results and performance over previous RS approaches. When it comes to planning a pilgrimage, pilgrims have a hard time narrowing down the list of temples they want to see. This recommender system provides recommendations for pilgrims to find suitable shrines, and the system provides information on the location of temples and finds the distances to the destinations. The OTPS provides more accurate results by providing recommendations based on the user’s interests. Following the same concept, applications can also be developed that can address the following issues: query processing and weather recommendations

Author Contributions: Conceptualization, D.R.T. and A.H.; Investigation, D.R.T., A.H. and A.K.G.; Methodology D.R.T. and R.R.G.; Writing—Original draft, D.R.T.; Software, A.H.; Project Administration, Resources and Supervision, A.K.G. and M.S.; Formal Analysis, Validation and Visualization, R.R.G.; Writing—review and Editing, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Dongseo University, “Dongseo Cluster Project” Research Fund of 2022 (DSU-20220006).

Conflicts of Interest: The author declares that there is no conflict of interest regarding the publication of this paper.

References

1. Cao, B.; Zhao, J.; Lv, Z.; Yang, P. Diversified Personalized Recommendation Optimization Based on Mobile Data. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 2133–2139. [CrossRef]
2. Arif, Y.M.; Harini, S.; Nugroho, S.M.S.; Hariadi, M. An Automatic Scenario Control in Serious Game to Visualize Tourism Destinations Recommendation. *IEEE Access* **2021**, *9*, 89941–89957. [CrossRef]
3. Turnip, R.; Nurjanah, D.; Kusumo, D.S. Hybrid recommender system for learning material using content-based filtering and collaborative filtering with good learners’ rating. In Proceedings of the 2017 IEEE Conference on e-Learning, e-Management and e-Services (IC3e), Kuching, Sarawak, Malaysia, 16–17 October 2017; pp. 61–66.
4. Nitu, P.; Coelho, J.; Madiraju, P. Improvising personalized travel recommendation system with recency effects. *Big Data Min. Anal.* **2021**, *4*, 139–154. [CrossRef]

5. Raza, S.; Ding, C. News recommender system: A review of recent progress, challenges, and opportunities. *Artif. Intell. Rev.* **2021**, *55*, 749–800. [CrossRef] [PubMed]
6. Migliorini, S.; Carra, D.; Belussi, A. Distributing Tourists among POIs with an Adaptive Trip Recommendation System. *IEEE Trans. Emerg. Top. Comput.* **2019**, *9*, 1765–1779. [CrossRef]
7. Thakker, U.; Patel, R.; Shah, M. A comprehensive analysis on movie recommendation system employing collaborative filtering. *Multimed. Tools Appl.* **2021**, *80*, 28647–28672. [CrossRef]
8. Yin, P.; Ye, M.; Lee, W.C.; Li, Z. Mining GPS data for trajectory recommendation. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Tainan, China, 13–16 May 2014; Springer: Cham, Switzerland, 2014; pp. 50–61.
9. Zhang, Q.; Lu, J.; Jin, Y. Artificial intelligence in recommender systems. *Complex Intell. Syst.* **2021**, *7*, 439–457. [CrossRef]
10. Zheng, W.; Yin, L.; Chen, X.; Ma, Z.; Liu, S.; Yang, B. Knowledge base graph embedding module design for Visual question answering model. *Pattern Recognit.* **2021**, *120*, 108153. [CrossRef]
11. Celdrn, A.H.; Prez, M.G.; Clemente, F.J.G.; Prez, G.M. Design of a recommender system based on users' behavior and collaborative location and tracking. *J. Comput. Sci.* **2016**, *12*, 83–94. [CrossRef]
12. Martinkus, P.; Madiraju, P. *Personalizing Places of Interest Using Social Media Analysis*; Milwaukee: Brookfield, WI, USA, 2015.
13. Shen, Y.; Zhao, L.; Fan, J. Analysis and Visualization for Hot Spot Based Route Recommendation Using Short-Dated Taxi GPS Traces. *Information* **2015**, *6*, 134–151. [CrossRef]
14. Chen, Q.; Lin, J.; Zhang, Y.; Ding, M.; Cen, Y.; Yang, H.; Tang, J. Towards knowledge-based recommender dialog system. *arXiv* **2019**, arXiv:1908.05391.
15. Wu, D.; Shang, M.; Luo, X.; Wang, Z. An L1-and-L2-norm-oriented Latent Factor Model for Recommender Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–14. [CrossRef]
16. Al Fararni, K.; Nafis, F.; Aghoutane, B.; Yahyaouy, A.; Riffi, J.; Sabri, A. Hybrid recommender system for tourism based on big data and AI: A conceptual framework. *Big Data Min. Anal.* **2021**, *4*, 47–55. [CrossRef]
17. Shao, B.; Li, X.; Bian, G. A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. *Exp. Syst. Appl.* **2021**, *165*, 113764. [CrossRef]
18. Yang, X.; Guo, Y.; Liu, Y.; Steck, H. A survey of collaborative filtering based social recommender systems. *Comput. Commun.* **2014**, *41*, 1–10. [CrossRef]
19. Iniyan, S.; Gupta, V.; Gupta, S. Facial Expression Recognition Based Recommendation System. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 5669–5678.
20. Huang, F.; Xu, J.; Weng, J. Multi-Task Travel Route Planning with a Flexible Deep Learning Framework. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 3907–3918. [CrossRef]
21. Yochum, P.; Chang, L.; Gu, T.; Zhu, M. Linked Open Data in Location-Based Recommendation System on Tourism Domain: A Survey. *IEEE Access* **2020**, *8*, 16409–16439. [CrossRef]
22. Qu, B.; Yang, W.; Cui, G.; Wang, X. Profitable Taxi Travel Route Recommendation Based on Big Taxi Trajectory Data. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 653–668. [CrossRef]
23. Symeonidis, P.; Ntempos, D.; Manolopoulos, Y. Location-Based Social Networks. In *Recommender Systems for Location-Based Social Networks*; Springer: New York, NY, USA, 2014; pp. 35–48.
24. Gao, K.; Yang, X.; Wu, C.; Qiao, T.; Chen, X.; Yang, M.; Chen, L. Exploiting Location-Based Context for POI Recommendation When Traveling to a New Region. *IEEE Access* **2020**, *8*, 52404–52412. [CrossRef]
25. Manjare, P.A.; Ninawe, P.V.; Dabhire, M.L.; Bonde, R.S.; Charhate, D.S.; Gawande, M.S. Recommendation System Based on Tourist Attraction. *Int. Res. J. Eng. Technol.* **2016**, *3*, 877–881.
26. Díaz-Ramírez, J. Gamification in engineering education—An empirical assessment on learning and game performance. *Heliyon* **2020**, *6*, e04972. [CrossRef] [PubMed]
27. Zhang, D.; You, X.; Liu, S.; Yang, K. Multi-Colony Ant Colony Optimization Based on Generalized Jaccard Similarity Recommendation Strategy. *IEEE Access* **2019**, *7*, 157303–157317. [CrossRef]
28. Shen, J.; Cheng, Z.; Yang, M.; Han, B.; Li, S. Style-Oriented Personalized Landmark Recommendation. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9768–9776. [CrossRef]
29. Shao, X.; Tang, G.; Bao, B.-K. Personalized Travel Recommendation Based on Sentiment-Aware Multimodal Topic Model. *IEEE Access* **2019**, *7*, 113043–113052. [CrossRef]
30. Ahmad, S.; Ullah, I.; Mehmood, F.; Fayaz, M.; Kim, D. A Stochastic Approach Towards Travel Route Optimization and Recommendation Based on Users Constraints Using Markov Chain. *IEEE Access* **2019**, *7*, 90760–90776. [CrossRef]