Vol. 01, Iss. 2 (2024) 71-81, DOI: 10.61552/JSI.2024.02.003

JSDI

Journal of Sustainable Development Innovations

www.jsi.aspur.rs

Virtual One-day Trip Planner

Sridhar Iyer[a], Nirzari Parikh[a,*], Khushi Jobanputra[a], Gauri Bhosle[a], Anushka Pandit[a]

[a]*D.J. Sanghvi College of Engineering, Plot No. U-15, J.V.P.D. Scheme, Bhaktivedanta Swami Marg, Vile Parle (West), Mumbai – 400 056, Maharashtra, India.*

Keywords:

Virtual one-day trip planner, Recommendation system, Hybrid Recommendation System, User-based Collaborative Filtering System and Item-Based Collaborative Filtering System.

* Corresponding author:

Nirzari Parikh
E-mail: nirzariparikh291@gmail.com

Received: 9 March 2024
Revised: 17 April 2024
Accepted: 1 June 2024

A B S T R A C T

A virtual one-day trip planner is necessary to simplify trip planning for users who wish to explore different places within a limited span of time. A Recommender system is used to make various suggestions according to the budget and time. A Recommendation system is of utmost importance to provide recommendations to users on all the aspects using the technology. Different types of recommendation systems are available, i.e., Hybrid Recommendation System, User-based Collaborative Filtering System and Item-Based Collaborative Filtering System. The user-based filtering compares a user with other users who have similar preferences based on the identical items ratings they have provided. On the other hand, Item-based filtering takes into consideration the previously rated preference of the user to make further recommendations. A Hybrid System consists of both types of collaborative filtering to make the recommendation to the user.

1. INTRODUCTION

Tourism makes a huge contribution to foster the progress of a country in various aspects like tourism, food industries and transportation. People face a number of problems in planning trips, for example, places to visit, limited time for trips, specific budget trips, etc. In such cases, a virtual day trip planner is a boon for the travelers. The complicated task of planning a day trip is eased by day trip planners. These planners take into account various criteria such as individual's available, budget, preferred mode of transportation and places of interests. However, identifying a user's preferred locations can be a daunting and time-consuming process. Recommender systems address this challenge by offering personalized suggestions, thereby assisting users in efficiently finding relevant information for their trips.

2. LITERATURE SURVEY

A local survey was conducted in 2023 and according to the results of the survey, 53.1% of people have never used a one-day trip planner.

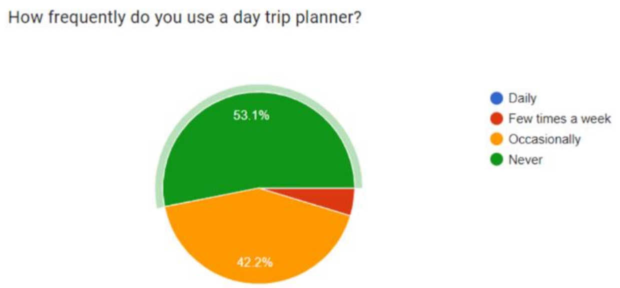

Fig. 1. Frequency of usage of a day trip planner.

Also, the survey concluded that nearly 42 out of 100 people cannot go to a one-day trip because of lack of information about the places that they can actually explore.

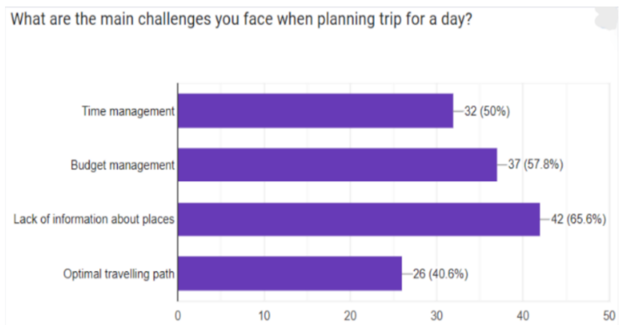

Fig. 2. Challenges faced for planning a day trip.

Furthermore, 56 out of 100 people have to compromise the place due to lack of time management for the one-day trip.

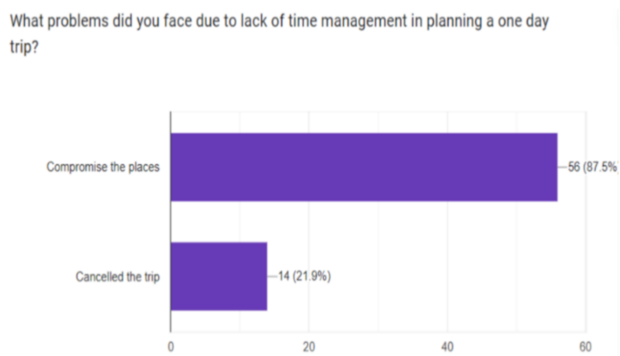

Fig. 3. Problems faced due to lack of time management.

The paper titled, "Coffee Shop Recommendation System Using an Item-Based Collaborative Filtering Approach" by Renita Astri, Ahmad Kamal, and Suaini Binti Sura, was used to conduct literature survey of the existing applications, addresses the need for a recommendation system in the context of increased coffee shop patronage during the COVID-19 pandemic. The study employs item-based collaborative filtering using adjusted cosine similarity to recommend coffee shops based on user ratings. This approach effectively identifies similar coffee shops and predicts user preferences, ultimately enhancing decision-making for coffee enthusiasts in Padang, Indonesia.

Another paper, "A Content Based and Collaborative Filtering Recommender System" by Vignesh Thannimalai and Li Zhang presents a novel recommendation system for tourist spots, integrating item-based collaborative filtering (CF) and content-based filtering (CBF) with a Naive Bayes Classifier (NBC). The system aims to provide personalized recommendations by utilizing user ratings and personal profiles, addressing the shortcomings of generic suggestions on most tourism websites. CF leverages user ratings to recommend top destinations, while CBF uses profiles for suggestions. The methodology section justifies these approaches, detailing the tools and resources used for development.

Testing and results analysis highlight CF's effectiveness in sparse content scenarios and CBF's ability to suggest based on user profiles. The implementation includes a web-based system built with HTML, CSS, JavaScript, and PHP, utilizing a MySQL database. CF predicts ratings using a cosine similarity formula, and CBF, combined with NBC, provides personalized recommendations.

Despite successful implementation, challenges like cold start and data sparsity remain. Future work suggests using hybrid systems and advanced machine learning techniques for improved performance. The paper emphasizes continuous adaptation to better meet user needs in tourism, proposing hybrid techniques and deep learning algorithms for enhanced accuracy and effectiveness.

Hence, a one-day trip planner would prove to be a useful tool for people who wish to explore different places in a very short span of time. It would help the user access the com- prehensive information about various destinations. Itinerary suggestions and time management tools to optimize their trip. Convenience and ease of planning, making it more likely for people to embark on one-day trips. Users can customize the entire trips according to their preferences providing flexibility to choose the duration of the trip, budget, other preferences and select specific activities or attractions. According to literature survey conducted on Coffee Shop Recommendation System Using an Item-Based Collaborative Filtering Approach through a research paper, an item based collaborative filtering system was built for people so that they could choose the best coffee shop in the locality to have a cup of coffee. The survey for the same was through manual questionnaire while one-day planner survey was conducted by circulating forms to people of various age groups like 15 - 30 year, 30 - 50 year and above 50 years. Most of the people were between the age group of 15 to 30 as they are the ones who face problems while planning a trip most of the time.

In addition to the existing literature, several other studies have highlighted the potential of recommendation systems in enhancing user experiences across various domains. For instance, a study by Smith et al. (2021) demonstrated how recommendation systems could be used to personalize museum visits by suggesting exhibits based on user interests and past visits. Similarly, Johnson and Brown (2022) explored the use of recommendation systems in healthcare, where personalized suggestions for lifestyle changes and medical treatments were provided based on patient data and preferences. These studies underscore the versatility and effectiveness of recommendation systems in diverse applications, further validating their use in trip planning.

3. PROBLEM STATEMENT

In today's fast-paced world, individuals seek immersive travel experiences. However, planning single-day trips can be time-consuming and difficult, often requiring consideration of personalized preferences, and options for optimized time and budgets. Existing virtual trip planning tools lack the ability to seamlessly integrate these diverse elements. The Virtual One-Day Trip Planner aims to ease the way individuals plan and experience one-day trips, offering a seamless and enjoyable experience. The Virtual One-Day Trip Planner eases the trip planning process, providing users with personalized plans and recommendations, customization options, and a user- friendly interface. The planner focuses specifically on one- day trip planning, aiming to enhance the travel experience by suggesting itineraries that meet individual preferences, time constraints, and budgets. Moreover, the lack of a robust one-day trip planner often results in missed opportunities for individuals to explore nearby attractions that they might not be aware of. This can lead to underutilization of local tourism resources and a diminished travel experience. By addressing these gaps, the Virtual One-Day Trip Planner aims to democratize access to travel planning tools, ensuring that even those with limited time and budget can enjoy fulfilling travel experiences. The system is designed to be intuitive and user-friendly, making it accessible to users of all ages and technical proficiency levels.

4. DATA COLLECTION

Data collection for various attractions was a comprehensive process, encompassing a wide range of places including malls, nature spots, theatres, restaurants, adventure parks, religious sites, clubs, and popular tourist destinations. Each of these attractions was meticulously cataloged with several key features such as the name of the attraction, its address, the type of attraction, user ratings, latitude, longitude and the nearest station for easy access. To ensure the accuracy and relevance of the data, the information was manually pre-processed. This involved verifying the details, standardizing the format, and cleaning up any inconsistencies or errors in the dataset. For ex- ample, names were standardized to avoid duplicates, addresses were verified for correctness, and ratings were normalized to ensure consistency across different attractions. The system is equipped with data, forming a solid foundation for generating personalized and accurate trip recommendations for users. This detailed and thorough approach to data collection and pre-processing ensures that users receive reliable and up-to-date information, enhancing their overall experience with the trip planning system. This extensive data collection effort is crucial for the accuracy and reliability of the recommendation system, enabling it to deliver high-quality, personalized suggestions to users.

5. PRODUCT MODEL

5.1 Flow of the Product

Initially upon Sign-up, the users will fill out a user interest form to indicate their preferences for various aspects of their trip, such as types of cuisines, etc. This information is used to build a detailed user profile that forms the basis for generating initial recommendations. The user interest form is designed to capture a comprehensive set of preferences to ensure personalized recommendations. After completing the user interest form, users will log into their accounts. The login process is secure and ensures that user data is protected. Upon logging in, users are directed to a personalized dashboard where they will be suggested initial recommendations based on the preferences indicated in their user interest form. The dashboard provides an intuitive interface, allowing users to easily navigate through various options and customize their trip plans.

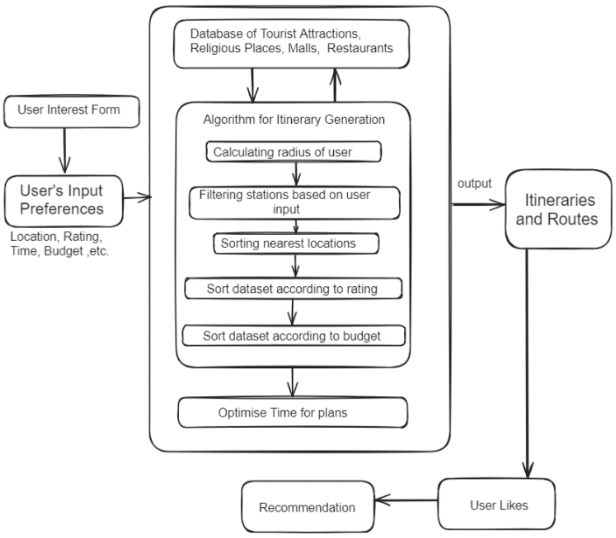

Fig. 4. Product Flow Model.

To plan a day trip, users can select the type of attractions they would like to visit, such as restaurants, malls, theatres, etc. Additionally, users can specify constraints such as the duration of the trip, budget limits, and preferred modes of transportation. The system takes these constraints into account and uses algorithms to generate and suggest optimized plans that best match the user's preferences and constraints After completing their trip, users have the option to provide feedback by liking the plans they enjoyed or rating various aspects of their experience. This feedback is crucial as it is used to update the user profile and refine future recommendations. The system continuously learns from user feedback, adapting and improving the accuracy and relevance of its recommendations over time. The ultimate goal of the system is to enhance user satisfaction by providing tailored trip recommendations that cater to individual preferences and constraints, thus improving the overall travel experience. Through its adaptive learning capabilities, the system strives to become more accurate and user-centric with each interaction.

5.2 High-Level System Architecture Diagram

Components:

User Interface (UI)

- User Interest Form: Upon signing up, users fill out this form to indicate their preferences for various aspects of their trip, such as types of cuisines, attractions, etc.
- Dashboard: After logging in, users are directed to the dashboard where initial recommendations are displayed. Users can select the type of attractions they want to visit and specify constraints like time and budget.
- Feedback: After the trip, users can provide feedback by liking the plans they enjoyed.

User Profile Manager

- Profile Creation: Based on the user interest form, an initial user profile is created, capturing the user's preferences.
- Profile Update: The user profile is continuously up- dated based on the user likes provided after each trip, improving the accuracy of future recommendations.

Recommendation Engine

- Initial Recommendation: Uses user profile to generate initial suggestions.
- Further Recommendation: Adjusts future recommendations based on user likes.

Planning Engine

- Trip Planning: Takes user-specified constraints (time, budget) into account to generate optimized trip plans.

Database

- User Data: Stores user profiles, preferences, and user likes.
- Liked Data: Stores liked plans.

Feedback System

- Like Mechanism: Allows users to like the plans they enjoyed, process likes to refine future recommendations.

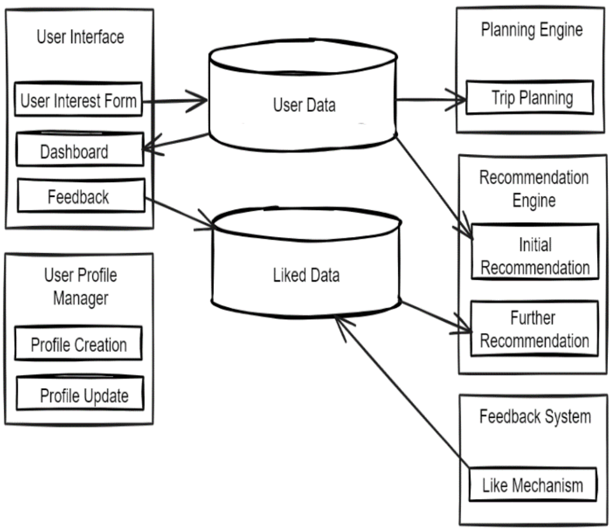

Fig. 5. System Architecture Diagram.

5.3 Recommendation Model

Recommendation Systems are software tools that are used to provide suggestions to user according to their preferences and requirements. Recommendation systems are a special type of information filtering systems which deals with the suggestion of items selected from a large collection that the user is likely to find interesting or useful.[18] Collaborative filtering-based recommendation system recommended objects that are selected on the basis of past evaluations of a large group of users.[17] Collaborative filtering is further sub-divided into User-based filtering and Item-based filtering. User-based method is when two users have similar ratings on some items, they will have similar ratings on the remaining items. In this approach recommendation for each active user is received by comparing with the preferences of other users who have rated the product in similar way to the active user.[17] Content- based filtering recommendation systems select items based on the correlation between the content of the items and the user's preferences. In content-Based filtering recommendations depends on A user's preference model and user's interaction log with the Recommendation system i.e. user's profile play an important role in Content-based filtering. Recent research has proved that a hybrid approach could be more effective in some cases. The main motive of hybrid approach is to aggregate collaborative filtering and content-based filtering to improve recommendation accuracy [18].

The product involves datasets covering various places like restaurants, clubs, malls, adventure parks, tourist spots, nature spots, religious places, and theatres. Each dataset includes details like name, rating, type, latitude, longitude, and station is collected. The product implements Hybrid Recommendation System containing User-based Collaborative Filtering System and Content-Based Filtering System.

User-based filtering compares a user with other similar users based on the preferences provided. On the other hand, Item-based filtering takes into consideration the previously liked preference of the user to make further recommendations. For content-based filtering a user profile is maintained which initially contains the users interests for a particular attraction (e.g. For Restaurants – Cuisines types like Chinese, Buffet, Fast Food, etc.). A new user is recommended initially according to the preferences collected through the user interest form. The profile of the user is then updated if the user likes a particular attraction. For User-based Collaborative Filtering System users are recommended based on users that have similar preferences e.g. For Restaurants users having similar preferences for cuisines will be recommended the same thing. A hybrid of both these recommendation models is considered and further the user is recommended according to the updated profile and the similar users. Similarity between two users is calculated and compared using Cosine similarity, Jaccard similarity, Manhattan distance and Euclidean distance.

The Cosine similarity is given by:

$$Cosine(A,B) = \frac{\vec{A}.\vec{B}}{\left|\left|\vec{A}\right|\right| \left|\left|\vec{B}\right|\right|} \tag{1}$$

The Jaccard similarity coefficient is defined as:

$$J(A,B) = \frac{|A \cap B|}{|A \cap B|} \tag{2}$$

The Manhattan distance between two vectors A and B is defined as:

$$d_{manhattan}(A, B) = \sum_{cuisine \in all\_cuisines} |A[cuisine] - B[cuisine]| \quad (3)$$

The Euclidean distance between two vectors A and B is defined as:

$$d_{euclidean}(A, B) = \sqrt{\sum_{cuisine \in all\_cuisines} (A[cuisine] - b[cuisne])^2} \quad (4)$$

```
Cosine vs Jaccard similarity correlation: 0.9964954748345323
Cosine vs Euclidean distance correlation: -0.37747335573549656
Cosine vs Manhattan distance correlation: -0.37747335573549656
Jaccard vs Euclidean distance correlation: -0.4172506637411647
Jaccard vs Manhattan distance correlation: -0.4172506637411647
Euclidean vs Manhattan distance correlation: 0.9999999999999998
```

Fig. 6. Comparisons of distance formulas.

Cosine vs Jaccard Similarity Correlation indicates high correlation coefficient which further indicates that the two measures are very similar in how they rank users. Cosine vs Euclidean/Manhattan and Jaccard vs Euclidean/Manhattan Distance Correlation has negative correlation coefficients which suggests that there is a weak negative relationship between them and that the Jaccard similarity might be a better measure of similarity compared to Euclidean/Manhattan distances and since Cosine vs Jaccard has high correlation coefficient either can be a better choice to choose.

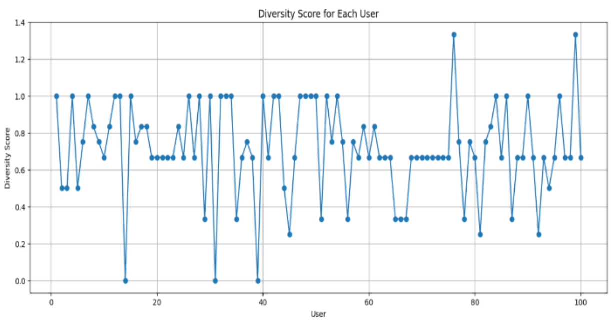

Fig. 7. Diversity scores Vs Users.

The graph illustrates the diversity scores of 100 users within the context of a restaurant recommendation system. Each user's diversity score represents the range and variety of restaurant preferences they exhibit.

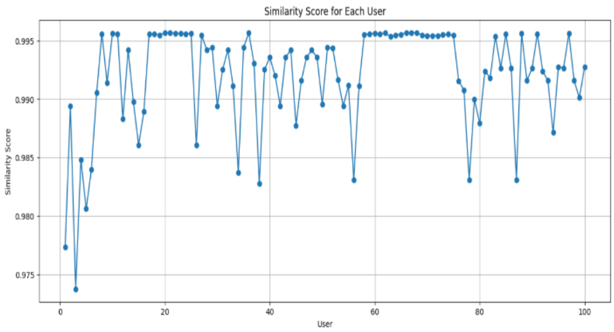

Fig. 8. Similarity scores Vs Users.

The graph depicts the similarity scores of 100 users in the context of a restaurant recommendation system. Each user's similarity score reflects the degree of overlap or similarity in their restaurant preferences with others.

After testing with 100 users, the average diversity score was found to be 0.75, indicating that, on average, the recommendations covered 75%. of the different types present in the initial recommendations. This suggests a good level of variety in the recommendations, which can be appealing to users who value diverse options. Additionally, the average similarity score was 0.97, indicating that the users' profiles were very similar to those of other users who received similar recommendations. This high similarity score suggests that the recommendations were well-targeted based on user profiles, aligning closely with users' preferences. In conclusion, the system achieved a good balance between offering diverse recommendations and aligning with users' preferences. The high similarity score indicates effective targeting, while the solid diversity score suggests a wide range of options for users.

The proposed recommendation system offers a comprehensive approach to trip planning, combining user input with optimization techniques to provide personalized and optimized trip plans. By incorporating user feedback into the recommendation process, the system aims to enhance user satisfaction and provide a seamless trip planning experience.

5.4 Planning Algorithm

A distance matrix according to stations is constructed and the data is sorted according to the distance input and the distance matrix. For calculating the distance matrix between stations haversine formula is used.

The Haversine formula is used to calculate the distance between two points on a sphere given their longitudes and latitudes. It is defined as follows:

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1) . \cos(lat_2) . \sin^2\left(\frac{\Delta lon}{2}\right) \quad (5)$$

$$c = 2 . \ \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (6)$$

$$d = R . c \quad (7)$$

where:

Δlat is the difference in latitude between the two points,

Δlon is the difference in longitude between the two points,

lat_1 and lat_2 are the latitudes of the two points,

R is the radius of the Earth (mean radius = 6371 km).

d is distance between the two points.

Inputs from user are time, budget, station, rating, type. Distance input is calculated using available time taken as an input from the user. Average speed for different modes of transport is considered walking = 5 km/hr, train = 33.5 km/hr, driving = 58 km/hr. The distance traveled by an object is given by the formula:

Further the data is sorted according to the budget and rating. Using this sorted data the final plans are suggested taking into consideration the distances calculated using haversine formula to optimize the time for the plans.

Algorithm 1 Calculate-Distance (speed, time)

1: **procedure** CALCULATE-DISTANCE (speed, time)
2: distance = speed × time
3: **return** distance
4: **end procedure**

The algorithm starts by determining the number of stations, denoted as n. It initializes a distance matrix of size n xn, setting all elements to zero. This matrix will store the computed distances. The core of the algorithm involves nested loops where the outer loop iterates over each station i from 1 to n, and the inner loop iterates over each subsequent station j from (i + 1) to n. For each pair of stations (i, j), the Haversine distance is calculated using their latitude and longitude coordinates. The calculated distance is then symmetrically stored in both distance_ matrix[i][j] and distance_matrix[j][i] to ensure the matrix remains symmetric. Finally, the algorithm returns the fully populated distance matrix. This approach efficiently handles the pairwise distance calculations by lever- aging the symmetric property of the distance matrix, reducing redundant computations and ensuring optimal performance.

Algorithm 2 Haversine-Distance (lat1, lon1, lat2, lon2)

1: **procedure** HAVERSINE-DISTANCE (lat1, lon1, lat2, lon2)
2: Convert latitudes and longitudes from degrees to radians.
3: $\Delta lat = lat2 - lat1$, $\Delta lon = lon2 - lon1$
4: $a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1) . \cos(lat_2) . \sin^2\left(\frac{\Delta lon}{2}\right)$
5: $c = 2 . \ \text{atan2}(\sqrt{a}, \sqrt{1-a})$
6: $d = R . c$
7: **return** d
8: **end procedure**

The HAVERSINE-DISTANCE procedure is a fundamental algorithm used to calculate the shortest distance between two points on the surface of a sphere, considering the curvature of the Earth. This method is particularly useful in geographic information systems (GIS), navigation, and spatial analysis. The procedure starts by converting the latitude and longitude coordinates from degrees to radians. This conversion is necessary because trigonometric functions in most programming languages and mathematical libraries operate on radian values rather than degrees. Next, the algorithm calculates the differences in latitude and longitude between the two points. These differences are then used in the Haversine formula, which determines the great-circle distance based on spherical trigonometry. The formula calculates the central angle between the two points, taking into account the spherical shape of the Earth. The Haversine formula involves trigonometric functions such as sine and cosine to accurately reflect the curvature of the Earth. This approach ensures that the calculated distance is more accurate than a simple Euclidean distance, which would ignore the Earth's curvature. The result of the Haversine calculation is a central angle, which is then multiplied by the Earth's radius to obtain the great-circle distance. This distance represents the shortest path between the two points on the surface of the sphere, providing a more realistic measure than straight-line distances for applications involving geographic locations.

Algorithm 3 Construct-Distance-Matrix (stations)

1: **procedure** CONSTRUCT-DISTANCE-MATRIX (stations)
2: n = number of stations
3: Initialize distance matrix[n][n] with zeros
4: **for** i = 1 to n **do**
5: **for** j = i + 1 to n **do**
6: d = Haversine-Distance(sts[i].lat, stns[i].lon, stns[j].lat, stns[j].lon
7: distance matrix[i][j] = d
8: distance matrix[j][i] = d
9: **end for**
10: **end for**
11: **return** distance matrix
12: **end procedure**

The procedure begins by determining the number of stations involved, denoted as *n*. A distance matrix of size *n xn* is then initialized with zeros. This matrix will store the distances between each pair of stations, with both rows and columns representing the stations. A nested loop structure is employed to compute the distances. The outer loop iterates over each station *i*, while the inner loop iterates over each subsequent station *j* (starting from *i* + 1). This approach ensures that each pair of stations is considered only once, optimizing the computational efficiency. For each pair of stations *i* and *j*, the HAVERSINE-DISTANCE function is called to calculate the great-circle distance between them. The function takes the latitude and longitude coordinates of the two stations as inputs and returns the distance, *d*, considering the curvature of the Earth. The calculated distance *d* is then stored symmetrically in the distance matrix. Specifically, the distance from station *i* to station *j* is the same as the distance from station *j* to station *i*, reflecting the symmetric nature of geographic distances. Therefore, *d* is assigned to both *distance_matrix*[*i*][*j*] and *distance_matrix*[*j*][*i*]. Once all pairs of stations have been processed, the completed distance matrix is returned. This matrix can be used for further analysis, such as route optimization, clustering, or network analysis, providing a foundational tool for understanding spatial relationships among the stations.

Algorithm 4 Filter-Stations-Within-Distance (stations, distances, max distance)

1: **procedure** FILTER-STATIONS-WITHIN-DISTANCE(stations, distances, max_ distance)
2: Initialize filtered_ stations as an empty list
3: **for** i = 1 to length(stations) **do**
4: **if** distances[i] ≤ max_ distance **then**
5: Add stations[i] to filtered_ stations
6: **end if**
7: **end for** –
8: **return** filtered stations
9: **end procedure**

This procedure is designed to filter a list of stations based on their distances from a reference point, using precomputed distance values. This filtering process is crucial in applications where selecting stations within a specified radius or distance threshold is necessary, such as location-based services, proximity searches, or network analysis. The procedure begins by initializing an empty list named *filtered stations*, which will store the stations that meet the distance criterion. This list will be populated during the iteration process. The procedure iterates through each station in the *stations* list, starting from index 1 up to the length of the list. For each station at index *i*:

- It checks if the precomputed distance stored in the *distances* list at index *i* (which corresponds to the distance of *stations[i]* from the reference point or another station) is less than or equal to the maximum distance threshold provided as an input parameter.
- If the distance condition (*distances[i] ≤ max distance*) is satisfied, the station *stations[i]* is added to the *filtered stations* list.

- If the distance condition is not satisfied, the station is not added to the *filtered stations* list.
- After iterating through all stations in the *stations* list and filtering those that meet the distance criterion, the *filtered stations* list contains all stations within the specified maximum distance from the reference point. Finally, the *filtered stations* list is returned as the output of the procedure, providing a subset of stations that are within the desired distance range. This subset can then be used for further analysis or processing as required by the specific application or algorithm utilizing this filtering mechanism.

Algorithm 5 Generate-Plans (user inputs, stations)

1: **procedure** GENERATE-PLANS (user_ inputs, stations)
2: Initialize parameters Earth's radius R, speed_walk, speed_ train, and speed_ drive.
3: distance = Calculate-Distance (speed, time)
4: dist_ mat = Construct-Distance-Matrix (stations)
5: fil stations = Filter-Stations-Within-Distance (stations, dist_ mat[station], distance)
6: sorted_ stations = Sort stations based on budget and rating (in ascending order of rating)
7: **return** sorted stations as optimized travel plans
8: **end procedure**

The GENERATE-PLANS algorithm is pivotal in creating optimized travel itineraries based on user preferences and station data. It begins by initializing essential parameters such as Earth's radius and different speeds for walking, train travel, and driving. These parameters are crucial for calculating distances and travel times, ensuring accurate itinerary planning. Next, the algorithm computes distances between stations using a distance matrix constructed through the Construct- Distance-Matrix procedure. This matrix facilitates efficient distance calculations using methods like the Haversine formula, establishing pairwise distances among all stations in the dataset. Once distances are established, the algorithm filters stations that fall within the maximum allowable distance from a reference point or another specified station. This filtering step, executed by Filter-Stations-Within-Distance, ensures that only relevant stations are considered for inclusion in the travel plans, optimizing for proximity. Following filtering, the algorithm sorts the remaining stations based on user-provided.

5.5 User Experience Design

The user experience (UX) design of the Virtual One-Day Trip Planner plays a pivotal role in its success. The system's interface is designed to be intuitive, visually appealing, and easy to navigate. Usability testing was conducted with a diverse test case to ensure that the interface meets the needs of various user demographics. Feedback from these tests was used to refine the design, resulting in a user-centric interface that enhances engagement and satisfaction.

5.6 Personalization Features

- User Preference Form: Upon logging into the Virtual One-Day Trip Planner, users are prompted to input their preferences regarding attraction types, preferred times for activities, and budget constraints. This input form ensures that the system can tailor recommendations to align with individual user interests and constraints. This personalization mechanism leverages user-specific data to enhance the relevance and satisfaction of the proposed itineraries.
- Optimized Plan Suggestions: Utilizing the preferences provided by users, the system employs an optimization algorithm to generate trip plans. These plans are customized to meet user-specific constraints and interests, such as selecting suitable attractions within the user's preferred time frames and budget limits. This approach aims to maximize user satisfaction by offering highly relevant and feasible trip options.
- User Feedback Loop: To continuously enhance the recommendation system, users are encouraged to provide feedback on their trip plans and experiences. This feed- back loop allows the system to learn from user inter- actions and preferences, thereby improving the accuracy and personalization of future

recommendations. By integrating user feedback, the system can adapt to changing preferences and provide more precise and satisfactory trip suggestions over time suggestions over time.

5.7 Future Scope

- Real-Time Collaboration: Enabling multiple users to plan trips together and make collective decisions on itineraries.
- Enhanced Personalization: Incorporating machine learning algorithms to provide more accurate and personalized recommendations based on user interactions.
- Social Features: Allowing users to share itineraries with friends and receive recommendations based on their social network's experiences.

6. CONCLUSION

The collaborative filtering method is just the right one to apply as it gives accurate results according to the user's needs. The highest rated place nearest to his location, that also falls under his budget is recommended to the user promptly. In the future, due to the increasing pace of lifestyle no one would want to waste hours deciding places to visit. The role of the one-day trip planner will become vital in the coming days and the fundamental principle of time management would be the key feature of the one-day trip planner. Also, budget trips, which in fact is one of the common problems faced by the people, would be resolved as they would know beforehand, what the trip would cost. Thus, keeping in mind the most critical aspects like time and money, the planner would suggest the best option for the user and help him enjoy the day to his fullest.

REFERENCES

[1] D. Sylejmani and A. Kadri, "A Survey on Tourist Trip Planning Systems," *International Journal of Arts & Sciences*, Gottenheim, Germany, pp. 13, 2011.

[2] N. Sun, T. Chen, W. Guo, and L. Ran, "Enhanced Collaborative Filtering for Personalized E-Government Recommendation," *Applied Sciences*, vol. 11, no. 24, p. 16, 2021. DOI: 10.3390/app112412119.

[3] P. A. Manjare, P. V. Ninawae, M. L. Dabhire, R. S. Bonde, D. S. Charhate, and M. S. Gawande, "Recommendation System Based on Tourist Attraction," *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, no. 4, pp. 877–881, Apr. 2016.

[4] X. Chen et al., "A Travel Recommendation Method Based on User Personalized Characteristics with Collaborative Fusion Matrix," in *Proc. 2020 5th Int. Conf. Mechanical, Control and Computer Engineering (ICMCCE)*, Harbin, China, 2020, pp. 1298-1301. DOI: 10.1109/ICMCCE51767.2020.00284.

[5] G. K. M., M. Haseeb, M. S. B., P. A. M. Zameel, and S. V. Raj, "Budget and Experience Based Travel Planner Using Collaborative Filtering," in *Proc. 2021 1st Odisha Int. Conf. Electrical Power Engineering, Communication and Computing Technology (ODICON)*, Bhubaneswar, India, 2021, pp. 1-7. DOI: 10.1109/ODICON50556.2021.9428978.

[6] A. Anjali, J. K. Sandhu, and D. Goyal, "User Profiling in Travel Recommender System using Hybridization and Collaborative Method," in *Proc. 2021 Int. Conf. Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, 2021, pp. 143-148. DOI: 10.1109/ICCCIS51004.2021.9397099.

[7] R. Astri, A. Kamal, and S. B. Sura, "Coffee Shop Recommendation System Using an Item-Based Collaborative Filtering Approach," in *IEEE Conference Publication*, 2022.

[8] W. Supanich and S. Kulkarineetham, "Personalized Tourist Attraction Recommendation System Using Collaborative Filtering on Tourist Preferences," in *Proc. 2022 19th Int. Joint Conf. Computer Science and Software Engineering (JCSSE)*, Bangkok, Thailand, 2022, pp. 1-6. DOI: 10.1109/JCSSE54890.2022.9836255.

[9] Y. Cui, C. Huang, and Y. Wang, "Research on Personalized Tourist Attraction Recommendation based on Tag and Collaborative Filtering," in *Proc. 2019 Chinese Control and Decision Conference (CCDC)*, Nanchang, China, 2019, pp. 4362-4366. DOI: 10.1109/CCDC.2019.8832961.

[10] X. Lv, "Analysis and Optimization Strategy of Travel Hotel Website Reservation Behavior Based on Collaborative Filtering," in *Proc. 2021 Int. Conf. Intelligent Transportation, Big Data & Smart City (ICITBS)*, Xi'an, China, 2021, pp. 362-365. DOI: 10.1109/ICITBS53129.2021.00096.

[11] Y. Cai, H. Gao, J. Liao, X. Li, Y. Xu, and J. Xiong, "A Personalized Recommendation Model based on Collaborative Filtering and Federated Learning for Cultural Tourism Attractions in Fujian-Taiwan," in *Proc. 2023 Int. Conf. Software and System Engineering (ICoSSE)*, Marseille, France, 2023, pp. 69-77. DOI: 10.1109/ICoSSE58936.2023.00020.

[12] T. Alenezi and S. Hirtle, "Normalized Attraction Travel Personality Representation for Improving Travel Recommender Systems," *IEEE Access*, vol. 10, pp. 56493-56503, 2022. DOI: 10.1109/ACCESS.2022.3178439.

[13] J. Wang, J. Wu, H. Wu, C. Wang, X. Zhang, and Y. Zhang, "Design and Research of Project Collaborative Participant Recommendation System Based on Collaborative Filtering Algorithm," in *Proc. 2022 Int. Symp. Advances in Informatics, Electronics and Education (ISAIEE)*, Frankfurt, Germany, 2022, pp. 187-190. DOI: 10.1109/ISAIEE57420.2022.00046.

[14] Li et al., "Research on the Application of Improved Collaborative Filtering Algorithm in Course Recommendation," in *Proc. 2023 Int. Conf. New Trends in Computational Intelligence (NTCI)*, Qingdao, China, 2023, pp. 90-93. DOI: 10.1109/NTCI60157.2023.10403738.

[15] R. G. Lumauag, "A Modified Memory-Based Collaborative Filtering Algorithm based on a New User Similarity Measure," in *Proc. 2021 Second Int. Conf. Innovative Technology Convergence (CITC)*, Sibalom, Philippines, 2021, pp. 69-73. DOI: 10.1109/CITC54365.2021.00020.

[16] "Comparative study of collaborative filtering algorithms," in *Proc. Int. Conf. Knowledge Discovery and Information Retrieval*, 2012. DOI: 10.5220/0004104001320137.

[17] B. Thorat, P. M. Goudar, and R. Barve, "Survey on collaborative filtering, content-based filtering and hybrid recommendation system," *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31–36, 2015. DOI: 10.5120/19308-0760.

[18] A. Rustamov and F. Bekkamov, "Recommender Systems: An Overview," *Scientific Reports of Bukhara State University*, vol. 5, no. 3, pp. 2–12, 2021. DOI: 10.52297/2181-1466/2021/5/3/1.

[19] "Trip planner," *SpringerReference*, Springer-Verlag, Retrieved Jun. 5, 2024. DOI: 10.1007/springerreference57952.

[20] "Comparative study of collaborative filtering algorithms," in *Proc. Int. Conf. Knowledge Discovery and Information Retrieval*, 2012. DOI: 10.5220/0004104001320137.