NextStop: A Travel Destination Recommendation System

Hasanabbas Momin*, Ujjwal Agarwal†, Rishi Vakharia‡, and Varad Badhe§
*†‡§*Student at International Institute of Information Technology, Bangalore, India*
Email: {Hasanabbas.Momin, Ujjwal.Agarwal, Rishi.Vakharia, Varad.Badhe}@iiitb.ac.in

***Abstract*—We have created a Travel Destination guide that suggests cities for users to visit based on the cities visited by the user previously. This system helps to narrow down the possible options to go for tourism, and saves users their precious time which is spent on research for possible destination. We have used the paradigms of collaborative filtering and content-based filtering for recommending the cities. In collaborative filtering, both item-item similarity and user-user similarity are tried. The datasets used in the project are described in the 'dataset' section. Finally, we conclude by analysing which method works best for this problem.**

***Index Terms*—Travel recommendation system, Travel history, Collaborative filtering, Content-based filtering**

1. Introduction

Our team has developed a Travel Destination guide that utilizes a user's previous travel history to suggest potential cities for future visits. This system streamlines the process of researching and selecting travel destinations, ultimately saving users valuable time. By leveraging a user's past travel experiences, our guide effectively narrows down the vast array of possible destinations, providing personalized recommendations that align with the user's interests and preferences. Our system represents a significant advancement in the field of travel recommendation systems, offering a practical and efficient solution for travelers seeking to optimize their tourism experiences. To provide accurate and relevant travel recommendations, our Travel Destination guide utilizes both collaborative filtering and content-based filtering techniques. Collaborative filtering involves analyzing a user's past travel history and identifying patterns and similarities with other users to make recommendations. Content-based filtering, on the other hand, involves analyzing the characteristics and attributes of a user's past travel destinations to suggest similar destinations that align with the user's interests. By applying these techniques individually, our system is able to provide personalized recommendations that are tailored to each individual user's travel history and preferences. This approach represents a significant contribution to the field of travel recommendation systems, offering a practical and efficient solution for travelers seeking to optimize their tourism experiences.

2. Dataset Creation

Our dataset consists of 5,000 unique users and their travel history in 3,000 different cities. As user information is not publicly available, we sourced our data from NomadList, a website that provided the necessary information. Our data is structured as a user and the list of cities visited by that user. We did not focus on ratings of the cities visited by the user, but rather on the places they have visited. In preprocessing, we removed all non-city entries from the user's travel history, such as beaches, resorts, airports, etc.

Additionally, we have a separate dataset that contains attributes for all 3,000 cities in our dataset. For example, the city of Bangalore has attributes such as Technology, Nightlife, and Startups. This attribute dataset was publicly available and provides valuable content information for each city, enabling us to make recommendations for new cities as well. By utilizing both of these datasets, we are able to effectively apply both collaborative filtering and content-based filtering techniques to provide personalized travel recommendations to our users.

3. Approach

3.1. Collaborative Filtering

Given the small size of our dataset, collaborative filtering is a practical and effective choice for building our Travel Destination guide. Collaborative filtering has the capability to identify patterns and similarities among both users and items, and then utilize this information to make personalized recommendations for the next item to the user. By leveraging user-item interactions, collaborative filtering can effectively capture the preferences and interests of individual users, ultimately leading to more accurate and relevant recommendations. Therefore, we have chosen to utilize collaborative filtering as one of the primary techniques in our Travel Destination guide, as it offers a powerful and efficient solution for providing personalized travel recommendations to our users.

3.1.1. Based on Item-Item Similarity . In item based collaborative filtering, recommendations are made to a user based on the similarity of items to which they have been associated to in the past. For implementing item-item collaborative filtering, the system first calculates the similarity between each pair of items and stores it somewhere. In our case, we have a vector corresponding to each city which contains the information about the users who have visited that particular city. If a user has visited a particular city, their corresponding index in the vector is marked as 1, while if they haven't visited, the index is marked as 0. We compute the pairwise similarity between cities using their vector representations and store the similarity scores. Whenever a user inputs their travel history by specifying the cities they have previously visited, we retrieve the top cities that are most similar to the user's input cities(using the similarity scores that were previously computed) and use this information to generate recommendations. The similarity score used is cosine-similarity.

3.1.2. Based on User-User Similarity. In user-user collaborative filtering, the system recommends items to a user based on the preferences of other users who have similar interests. If we want to recommend cities to a given user based on their travel history, we calculate the jacard similarity between that user and all other users in the dataset as we have binary data, we don't have ratings if we would have ratings we would have used cosine or pearson similarity. Then, we iterate through each city and identify the users who have visited that city, adding up their similarity scores with respect to the given user. Finally, we recommend the top cities based on the calculated similarity scores. The similarity score used is jaccard-similarity.

3.2. Content Based Filtering

Content-based filtering considers the cities that are visited by a person and tries to recommend similar cities. To perform this task, we'll need to compute the similarity between any 2 cities. To help us, we have a data-set which has the city name and its top 3 attributes for around 3,000 cities distributed around the globe. We'll use this data-set to find the similarity between cities. For this, we convert the attributes which are English words into numerical vectors with the help of "GloVe-Twitter-25" model. Now, each city is represented by a set of 3 '25-dimensional' vectors. To compute the similarity between two cities which have vectors $[u_1, u_2, u_3]$ and $[v_1, v_2, v_3]$ resp., we compute $u_1 \cdot v_1 + u_1 \cdot v_2 + u_1 \cdot v_3 + u_2 \cdot v_1 + u_2 \cdot v_2 + u_2 \cdot v_3 + u_3 \cdot v_1 + u_3 \cdot v_2 + u_3 \cdot v_3$ and use this as a measure of similarity between the cities. Why don't we take the average of u_1, u_2, u_3 and do its dot-product with the average of v_1, v_2, v_3? This can be an approach but averaging will lead to some amount of data loss and can lead to different cities being classified as similar. That's why we use the above approach to compute similarity.

Now, if a user says the he likes a city 'A', we can recommend the cities that are similar to 'A' by computing similarities of all cities with 'A' and taking the cities with

Figure 1. Disadvantage of taking dot-product of average of vectors - data loss

highest similarity.

But the input we can get for a user can contain more that one city. In that case, we combine the attributes of all visited cities and remove the attributes that occur more than once. Thus, we get set of vectors $[u_1, u_2, ...u_m]$ which represents the attributes of cities visited by user. Now, we find the similarity of this set of vectors with every other city which is represented by the set of vectors let's say $[v_1, v_2, v_3]$. We'll again not take the average of vectors and take dot product, but instead consider every possible pair of vectors between the sets $[u_1, u_2, ...u_m]$ and $[v_1, v_2, v_3]$, take their dot product and sum it all up to get the similarity between the sets. The argument against taking average is once again the case of data-loss and imperfect classification of similar cities. Next again we can take the cities that have highest similarity score and recommend them.

Thus in this way, given the input as the cities that a user has visited, we can recommend them similar cities to visit.

3.3. Comparision

Content based filtering performs better in the cold start scenario, as it can utilize item attributes and characteristics to make recommendations even when there is limited data about items or users. Collaborative filtering faces challenges in the cold start scenario due to its reliance on user-item interactions.So if we are using collaborative filtering then we cannot recommend places for a new user without their travel history while in content based approach if we get the user interest we can recommend places to visit. Content based filtering also performs better in terms of scalability as it can handle a larger number of users and items efficiently as it focuses on item attributes while the collaborative based approach takes polynomial time to calculate similarity which is not feasible for large number of users and items.

But Based on our problem we will find that the collaborative filtering will recommend better than the content based filtering. We will see this in Result Analysis.

4. Challenges

Dataset: Since we required user-specific details of their travel history, this type of data was not publicly available due to privacy concerns. In applications like Trip Advisor, we typically only have access to user reviews for specific places, making it difficult to extract information that we require as we need the cities visited not places in a particular city. Therefore, creating the dataset posed the biggest challenge for us.

Neural Network: We attempted to apply a neural network (NN) for our approach. Our initial step was to obtain city embedding using word2vec. We then calculated user embedding by taking the mean of the city embedding for cities visited by each user. Next, we trained the neural network by inputting the concatenation of the user embedding with the city embedding and predicting the embedding of the next city. However, we encountered difficulty in retrieving the city name from the output embedding.

5. Model And Result Analysis

We evaluated the accuracy of our Travel Destination guide by using one entry from the cities dataset. Specifically, we provided the first four cities in the dataset as the actual data point of the user: [Paris, Sicili, Amsterdam, London, Instanbul, Athens, Beirut, Dubai]. We then used our model to make recommendation for the remaining cities in the dataset and compared these recommendation to the actual data. The input for this evaluation was the first four cities in the dataset: [Paris, Sicili, Amsterdam, London]. By conducting this analysis, we were able to assess the accuracy and effectiveness of our Travel Destination guide in providing personalized travel recommendations to our users

We found that user-user based collaborative filtering recommended the following cities for the user: Berlin, Barcelona, New York City, Lisbon, San Francisco, Prague, Bangkok, Budapest, Rome, and Canggu in order. In contrast, item-item based filtering recommended the following cities: Barcelona, Beirut, Rome, Berlin, Brussels, New York City, Athens, Dubai, and Istanbul in order. It is important to note that the user had previously visited Istanbul, Athens, Beirut, and Dubai. By comparing the recommendations from both techniques, we were able to assess the effectiveness of each approach in providing personalized travel recommendations to our users

Based on our analysis, we found that the Item-Item based method was more effective than the user-user based method in recommending cities that the user had previously visited. Specifically, the Item-Item based method recommended all four of the other cities that the user had visited (Istanbul, Athens, Beirut, and Dubai), while the user-user based method was unable to recommend any of these cities. However, it is important to note that the user-user based method's recommendations were not random and appeared to be intuitive. This suggests that both methods have their strengths and weaknesses, and the choice of which method to use may depend on the specific needs and preferences of the user.

We found that the content-based filtering method recommended the following cities for the user: Minneapolis, Karlsruhe, Darmstadt, San Jose, and Katowice. While these cities did not match the cities visited by the user, we observed that they were similar to the cities provided as input in terms of their attributes. This suggests that the content-based filtering method is effective at identifying cities that share similar characteristics and attributes to those that the user has previously visited. While this approach may not always provide recommendations that match the user's exact travel history, it can still offer valuable insights and suggestions for new and interesting travel destinations.

Expanding the attributes available for each city in our dataset, as well as incorporating information on the contribution of each attribute to the city's fame, and the country in which the city is located, would significantly enhance the effectiveness of our content-based filtering method. By incorporating additional data points and attributes, we could further refine our recommendations and provide more accurate and relevant suggestions for our users. This highlights the importance of continually updating and expanding our dataset to improve the effectiveness of our Travel Destination guide. By leveraging a more comprehensive and detailed dataset, we can offer a more personalized and tailored experience for our users, ultimately leading to greater satisfaction and engagement with our platform

6. Future Scope

- These model could be combined to create a hybrid, which will show better recommendation.
- We can ask the user their current location and climate conditions, we can consider these metrics to give extra weightage to nearby and optimal weather destination.
- RNNs can be applied to solve this problem. We can see this problem as a sequence modelling task as the order in which a person visits the destinations also holds some information. Based on the already visited places by the user, we can build a context of the user using RNNs and then use the context to make prediction about where the user will go next.

The authors would like to thank

References

[1] *Nomadlistscraper Repository*. [Online]. Available: https://github.com/julianprester/nomadlistscraper/tree/main/data/interim

[2] *Item-Item Collaborative Filtering with Binary Data*. [Online]. Available: https://medium.com/radon-dev/item-item-collaborative-filtering-with-binary-or-unary-data-e8f0b465b2c3

[3] *Building a City Recommender for Nomads*. [Online]. Available: https://www.datarevenue.com/en-blog/building-a-city-recommender-for-nomads