DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

A Recommender System for Planning Composite City Trips Based on Travel Mobility Analysis

Rinita Roy

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

A Recommender System for Planning Composite City Trips Based on Travel Mobility Analysis

Ein Empfehlungsdienst zur Planung von Städtereisen basierend auf Mobilitätsanalysen von Reisenden

Author:	Rinita Roy
Supervisor:	Prof. Dr.-Ing. Jörg Ott
Advisor:	Linus W. Dietz
Submission Date:	15.05.2020

Abstract

Location-based social networks (LBSNs) are rich sources of studying travel mobility of people. With more users sharing updates about their activities in LBSNs, there is availability of enough data to learn about their travel mobility patterns (TMPs). Design of destination recommender systems (DRSs), in turn, can benefit from better understanding of travellers' mobility patterns. We research on wide-range literature in the field of TMPs and DRSs, and also discover no such system that recommends personalised city trips to different users by employing data-driven approaches. In this thesis, we study the TMPs of people using trips extracted from Twitter check-ins and design a DRS that computes personalised city trips for users based on their travel preferences. We come up with 10 prototype traveller types, having similar and dissimilar features with each other. We also divide the world into 10 regions, and segregate $93,955$ trips based on the home regions of travellers. The partitioned trips are then clustered, which in turn, also group the travellers around the world, conforming to the previously modelled traveller types. Finally, we develop a prototype web application, *TripRec* to test the functionalities of our algorithm that recommends destinations from a set of 138 cities in the database. The application accepts user information and preferences like home region, destination region, traveller type, maximum travel duration and fondness for different types of venues in a city, as inputs. Satisfying the user preferences and constraints, a suitable trip including an ordered list of cities with duration of stay at each is determined, to be recommended to the user. This thesis, thus, reveals a novel data-driven approach to build a recommender system application for planning composite city trips, personalised to user requirements.

Contents

1. Introduction

Tourism can be described as the act of travelling to somewhere away from home for leisure, business or other purposes. Travel and tourism industry contributed approximately 2.9 trillion U.S. dollars to the Gross Domestic Product (GDP) in 2019, globally [25]. It supports in the growth of a range of other industries, some broad categories being transportation, accommodation, food & beverages, entertainment and recreation. Thus, it is very important to continue the research, finding different ways to satisfy the customers of the industry, the tourists, and keep tourism flourish for building the economy of all countries.

Information technology (IT) has played a very significant role to revolutionise the tourism industry in the recent years [10]. Destination recommender systems (DRSs) are one of the noteworthy applications developed for travellers to assist them discover destinations to travel to. Depending on the type of data utilised, a recommendation model can be collaborative-filtering or content-based. The former model is typically based on user feedback — explicit or implicit. Content-based recommender systems use the characteristic features of the items and the preferences of a user before generating the right recommendations for them. The design of recommender systems (RSs), which earlier relied only on intuition-based models, is now employing more data-driven approaches [17]. The latter involves analysis of large sets of data, interpreting and incorporating them for building better decision-making strategies.

City tourism, also known as urban tourism involves travelling to the urban cities of different countries. It facilitates the development of the cities to attract tourists. Moreover, with more than half of the world population staying in urban areas [54], city tourism is important for the economy as it brings employment to numerous individuals. On the other hand, this is mainly interesting for tourists who prefer to visit locations including architectures & monuments, pubs & bars, restaurants & cafés, ski and skydiving areas. However, it is difficult for people to determine desirable top destination cities to be visited for their next trip. City recommender systems become useful in this context.

Google Trip[1] collects data from Gmail account of a user and combines it with other features like crowd-sourced reviews about destinations for suggesting trips to her. However, this is not personalised for users without prior Gmail accounts. There is no application in the market that uses data-driven approaches for determining personalised, composite city trips for all users, motivating us to start working further in that direction. In this master thesis, we discover travel mobility patterns (TMPs) in the data retrieved from location-based social networks (LBSNs), and utilise them for composing composite city trips, personalised to user preferences.

In our work, trips extracted from Twitter check-ins are segregated based on home locations of their travellers and each partition is further grouped using their characteristic mobility features. This also divides the corresponding travellers into different types, whose typical behaviours are described by us. We design a content-based recommendation algorithm for utilising users' travel preferences to generate personalised city trip recommendations for different types of travellers similar to each other. With 138 cities in our database, a prototype web application, *TripRec* is developed thereafter, to examine the functionalities of our algorithm and check the feasibility of our idea to recommend composite city trips using Twitter check-ins. The thesis has the following research questions:

RQ1: What is the relevant research and which are the applications involving the development of data-driven DRSs?

RQ2: How can trips generated from the LBSNs be characterised?

RQ3: Which are the types of travellers in different regions of the world?

RQ4: How can composite city trips be planned for different types of users, given their travel preferences and constraints?

RQ5: How can a suitable user interface be designed for the composite city trip recommender system?

In the upcoming chapters, we seek for the answers to the above questions. Chapter *Related Work* reviews the research in the fields relevant to our research topic. We study the areas like the human mobility patterns (HMPs), destination characterisation, and the manifold tourist recommender systems developed using various techniques.

In chapter *Data Engineering for Content-based Recommendation*, we discuss how we characterise cities and also the trips extracted from Twitter check-ins. We explore our datasets for cities and trips, and prepare them for further analysis. We also

[1] https://www.google.com/travel/

discuss dividing the world into 10 customised regions. Chapter *Identification of Regional Traveller Types* discovers the different types of travellers found in these regions using clustering techniques. We describe their characteristics and identify similarities with each other.

Chapter *Algorithmic Aspects of Recommending Composite City Trips* presents an algorithm to plan the composite city trip for a user, including the duration of stay at each recommended city. The recommendation is based on the preferences of the user and satisfies her constraints, answering RQ4. This is followed by chapter *TripRec — an Application for Recommending Composite City Trips*, which discusses the different aspects in the development of our prototype application. This web application (a) accepts user inputs, (b) utilises the designed algorithm to recommend an ordered list of cities to the user, along with duration of stay at each of them, and (c) displays a feedback form to be filled up by the user.

The chapter *User-centric Evaluation of TripRec* analyses the user data collected during the user study, when people utilise our web application to get trip recommendations and provide feedback about their experiences. We inspect the user data to study their behaviours, services provided to them, and the shared feedback. We summarise this master thesis, discuss the limitations and present scope for future work in chapter *Conclusion & Future Work*.

Thus, after a thorough literature survey (RQ1), we engineer check-in data from LBSNs to study travel mobility for characterising trips (RQ2), and identify patterns in them (RQ3). They are utilised to build a content-based recommender system that plans composite city trips for travellers based on their preferences and constraints (RQ4). A web application is finally built as a prototype to test the functionalities of the RS (RQ5).

2. Related Work

In this chapter, we discuss the literature on HMPs and RSs, focusing on the tourism domain. We further elaborate the related work on composite tourist recommender systems (CTRSs), focusing on the tourist trip design problem (TTDP). The duration of stays, customisation and the context factors associated with destination recommendations are also explored. We look for clustering or classification problems involved in designing RS applications. Finally, we explore how to characterise destinations, and what are the different data sources and features involved in the characterisation, followed by discussing functional regions.

2.1. Human Mobility Patterns (HMPs)

Studying HMPs has been a popular area of research. Different data sources like mobile traffic data [30, 57], check-in data from LBSNs including Foursquare [45] and Twitter [34], and vehicle GPS data [32, 72] have been used for studying the same. Sevtsuk and Ratti used mobile data to prove the regularity in urban mobility at different hours, days, and weeks [57]. However, earlier, Hanson and Huff noted the irregularity or existence of more than one routine in human's daily or weekly mobility patterns [33]. The human movement patterns across different cities are not universal, as was also shown by Noulas et al. [45]. However, they also revealed that instead of the physical distances, if the rank-distances between places were considered as factor to analyse the human mobility in different cities, they displayed universal patterns. Ranks accounted for the number of places of interest between an origin and a destination.

Analysing TMPs is an important subset of studying HMPs. Analysing TMPs can improve composite trip recommendations in future, considering explicit factors such as time and money, and prior travelling patterns [18]. González et al. drew similarities in the travel patterns of anonymous mobile phone users using their mobile communication data [30]. They characterised individuals with time-dependant characteristic travel distance and returning probabilities to different locations and human trajectories were thus shown to follow temporal and spatial regularities. Others [18, 19, 34] used LBSNs to analyse TMPs. Dietz et al. mined check-in data of Foursquare users and proposed a model for identifying different metrics to check if a traveller's check-in behaviour was adequate to derive TMPs [18]. This was followed by utilising the derived trips to characterise trips and travellers [19], and then the travel regions around the world [56]. Hawelka et al., on the other hand, analysed geo-located messages from Twitter to find similarity patterns of tourists globally, followed by comparing the result with worldwide tourism statistics and commonly-used human mobility models [34].

We analyse the TMPs of people after extracting trips out of Twitter check-ins and characterise the trips to find the type of cities included in them.

2.2. Recommender Systems (RSs)

RSs are used to suggest items of interests to users. Adomavicius and Tuzhilin [1] defined recommender systems as follows:

"More formally, the recommendation problem can be formulated as follows: Let C be the set of all users and let S be the set of all possible items that can be recommended ... Let u be a utility function that measures the usefulness of item s to user c, that is, $u : C \times S \to R$*, where R is a totally ordered set (for example, nonnegative integers or real numbers within a certain range). Then, for each user* $c \epsilon C$ *, we want to choose such item* $s \epsilon S$ *that maximizes the user's utility."*

2.2.1. Recommending Destinations

Destination recommendation can be further divided into recommending countries [67], cities [20], point of interests (POIs) [6, 12, 43], activities [55] or events [51].

Different ways to recommend social events from mobile phone location data and evaluation of the recommended lists were done by Quercia et al. [51]. They found that amongst the different ways used, recommending the nearby events proved to be the worst, whereas recommending events that were popular among the residents of the area, where the person who was being recommended belonged to, worked the best. They emphasized that sharing attendance at social events can lead to receiving better recommendations of future events. Roy and Dietz rather presented a model (Figure 2.1) that utilized the health parameters, such as heart rate, body temperature and blood pressure, measured by wearable sensors to infer the physiological conditions like stressed, active, or hungry of a user [55]. These were then availed to derive the type of activities like arts & entertainment, outdoor & recreation, or food to be recommended to the tourists [55].

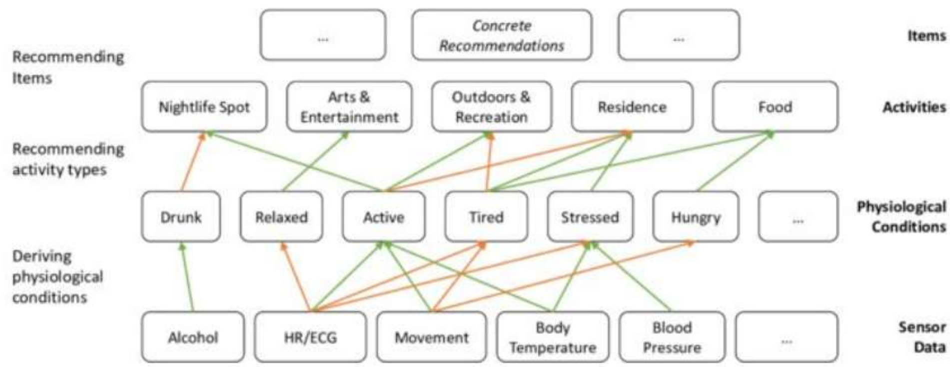


Figure 2.1.: Model for recommending tourist activities by Roy and Dietz [55]

New services can be provided after knowing the current location and predicting the next locations of the user [7]. Massimo and Ricci proposed a recommender system [43] that generated relevant next POI to be visited to tourists, after analysing POI visit trajectories and learning tourist behaviours via inverse reinforcement learning. The recommender systems developed by Bao et al. [6] and Cao et al. [12] recommended top-k destinations to tourists using two common types of recommender systems, collaborative filtering and case-based recommender systems, respectively. Bao et al. [6] determined the similarities between local experts and a user, and the similarity scores fed into collaborative-filtering-based model helped calculating the ratings of different locations for the user. The tourism recommender system designed by Cao et al. [12], on the other hand, asked users to choose keywords or images to describe their interests. This was followed by ranking the locations based on the similarities between characteristics of query images and tags, and that of representative images and tags of different clusters of the photos, determined earlier. The higher ranked locations were recommended to the corresponding users in both the cases [6, 12].

2.2.2. Tourist Packages

RSs can recommend single or multiple items. A composite recommender system (CRS) can be considered to recommend packages consisting of sets of single items. Xie et al. designed a CRS arranging the items in a value-sorted order, each item having a value and a cost associated with it [70]. It is a variant of the knapsack problem (KP) with the restriction of items to be accessed in non-increasing order of their values.

Different models for personalised travel package recommendations were developed by Liu et al. [40]. The key characteristics of different travel packages are analysed to develop the Tourist-Area-Season Topic (TAST) model, which found the inter-relations between tourists and spatial (locations) and temporal (travel seasons) characteristics of landscapes. This model was then utilised to develop a cocktail approach on personalised travel package recommendation by considering other factors like the seasonal behaviours of tourists, prices of packages, applying collaborative filtering on them, and dealing with cold start problems. TAST model was also extended to the tourist-relation-area-season topic (TRAST) model, that helps understand the reasons for tourists to form a travel group and find the latent relationships between the tourists in the group [39]. Figure 2.2 shows the illustration of the TAST and TRAST models.

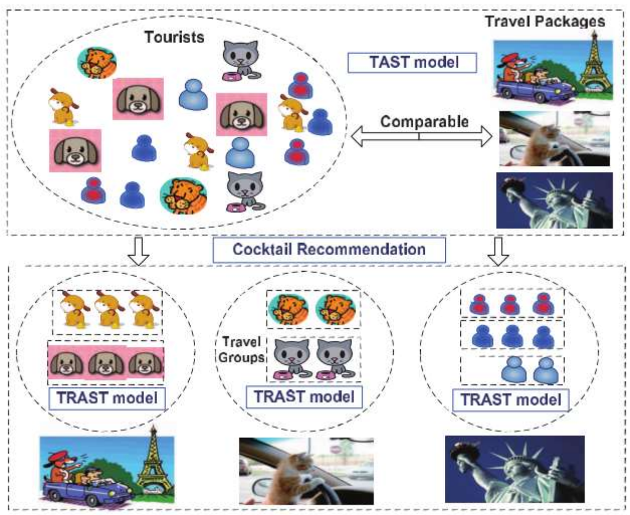


Figure 2.2.: Illustration of TAST and TRAST models [39]

2.2.3. Tourist Guide

Designing a tourist guide is another related interesting problem. Cyberguide [41] was one of the earliest context-aware mobile tourist guides that provided several services to tourists based on their position and orientation. It was designed to work into modules for assisting the tourists by providing important services to them. The services were visualised by the authors as the roles of a cartographer, a librarian, a navigator and a messenger. Cyberguide provided the service of a cartographer by means of maps, that showed locations of different sights. It provided descriptions about interesting sights or associated people, helped in finding locations of fellow tourists, and could also be used for communication by means of exchanging electronic messages. Refer to Figure 2.3 for the map and information views of Cyberguide.

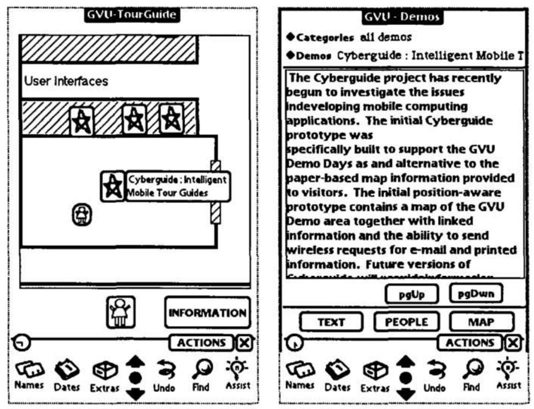


Figure 2.3.: Map view and information view of Cyberguide [41]

An electronic tourist guide can support tourists with information and navigation requirements for visiting a city [14, 38, 46] or in and around a monument [48]. Oppermann and Specht [48] described how an electronic guide could be used for a visit to a museum — collecting information about contents, exhibition organisation, location, and opening hours, followed by planning and executing visits. Others [14, 38, 46] described their own variations of tourist agents acting as dynamic tour guides of a city. They allowed tourists to choose attractions for planning personalised tours, and helped them in navigating and providing information based on the location or point of interest. Other than providing these basic services, the context-aware electronic tourist guide application, GUIDE [14] also facilitated other services like communication using electronic messaging service and booking of an accommodation. Finally, Vansteenwegen and Oudheusden talked about a next generation mobile tourist guide (MTG) that modelled the TTDP as an extension of the orienteering problem (OP) [66]. The authors discussed the important functionalities a next generation MTG should have to produce and manage personalised, integrated trip plan for an entire holiday period.

The RS developed by us recommends destinations, or more specifically, multiple cities to a user.

2.3. Composite Tourist Recommender Systems (CTRSs)

A composite trip consists of a sequence of travel destinations. CTRSs deal with choosing a number of travel destinations, selecting the sequence of visit, and determining suitable duration of stay in each of the destination.

Researchers in the past developed CTRSs recommending multiple countries [36, 67], or point of interests (POIs) [68, 71]. Dietz and Weimert used crowdsourced trips in wOndary for developing trip itineraries [22]. They presented how by using the platform, users were allowed to organise personal trip itineraries, collaborate with other travellers and also share details of their personal itineraries, which could later be looked up and followed by others. CTRSs are mostly otherwise designed by solving TTDP, discussed in the next subsection.

2.3.1. Tourist Trip Design Problem (TTDP)

Gavalas et al. [27] defined TTDP as follows:

"The tourist trip design problem (TTDP) refers to a route-planning problem for tourists interested in visiting multiple points of interest (POIs). TTDP solvers derive daily tourist tours, i.e., ordered visits to POIs, which respect tourist constraints and POIs attributes."

Gavalas et al. [27] presented a survey of the available models or approaches concerned to solve the TTDP. The models mostly dealed with finding points of interests to be recommended to tourists according to their preferences, maximising their satisfaction, and following a range of constraints. Many open issues on the related area, with few or no much relevant work on them, were also pointed out in the paper [27]. Herzog et al. discussed how TTDP could be modelled using different mathematical problems like travelling salesman problem (TSP), OP and its variants, and KP [35].

A web-based application was developed by Rani et al. [52] that constructed travel itinerary plans using TSP and k-means clustering. A user had to predefine the origin POI and the destination POI(s) from displayed POI lists, and also specify her desired travel duration. The user's preferred destinations were clustered into as many clusters as the number of specified travel days. For each cluster of POIs, all possible visit orders were considered, and the one with the smallest total distance for each cluster was selected. The recommended itineraries for each day kept the origin same as the final destination, as is the case of TSPs. An application designed for visiting all the national parks in U.S. showed another example of modelling a TTDP using TSP [47]. This road trip was designed to form an enclosed loop (Figure 2.4), around U.S. to include all the national parks. This was done so that the trip can be started from any point and proceeded in either direction.

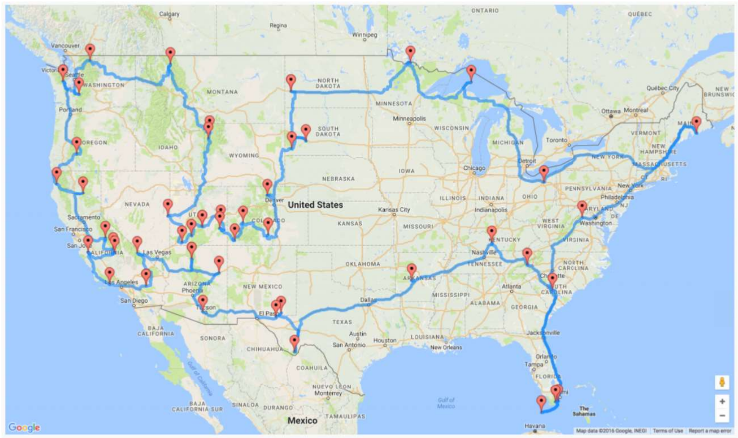


Figure 2.4.: An optimal route for a road trip to visit U.S. National Parks [47]

A case-based recommender system, that recommended multiple travel regions to a user for individual travelling, was developed by Herzog and Wörndl, where the TTDP was modelled as a KP [36]. It selected the travel regions with the maximum values, keeping users' time and money constraints into account. This was also extended by affecting the presence of a region in the recommended composite trip with the presence or absence of other regions. Wörndl presented a web application [67] that also recommended travel regions for independent travellers. The travel regions in both of these applications corresponded to mid-sized countries in general [36, 67].

In another application developed by Wörndl et al. [68], Foursquare-based POIs were combined to generate interesting routes. This was done using constraint-free and constraint-based variants of a procedure based on Dijkstra's algorithm [24]. Sun and Lee [64], on the other hand, used GIS (geographic information system) spatial analysis functions and a heuristic search algorithm to develop a tourist route model. The model considered spatio-temporal behaviour of tourists and recommended routes to them. Each route consisted of a series of connected, geographically close road segments, having high attraction values. POIs, along with detailed maps and also routes for guided walks could be shown by an adaptive system called SAMAP [13], developed to help tourists visit different cities. After user modelling and finding activities to be done in a city, a suitable plan was generated by the system for a user.

Trip planning requires scheduling and recommending point of interests to be visited by tourists. Vansteenwegen et al. [65] discussed the functionalities offered by such systems involved in planning a trip. They also compared systems described in related papers, based on the presence or absence of these functionalities. Finally, a system was presented, that used the basic OP model having most of the discussed trip planning functionalities [65].

It is not enough for tourists to get suggestions of the POIs to visit. Sometimes, they want to receive the whole travel itinerary scheduling the visits. Ardissono et al. developed a tourist information server called INTRIGUE [3], that helped users to schedule their itinerary to plan a tour themselves. Sources like geo-temporal photo check-ins by users in Flickr [16] or crowdsourced trips in wOndary [22] were also used earlier for developing trip itineraries. In the application developed by Choudhury et al. [16], the photo streams of each user were extracted with estimated duration of stay at each place, followed by aggregating all users' photo streams into a POI graph. Itineraries were then constructed from the graph considering the popularity of the POIs and the time and location constraints of the user.

The CTRSs that model the TTDP in terms of KPs for recommending itineraries end up planning schedules, that are very unbalanced across different days. The best locations are all scheduled to be visited in some days, and the destinations of much lower qualities are left for the other days. Moreover, some neighborhoods are visited multiple times in different days. The orienteering algorithm developed by Friggstad et al. [26] focussed on providing high-quality, multi-day trip itineraries, by addressing and trying to solve these problems.

The CTRS designed by us also uses the approach of solving TTDP with the maximum travel duration constraint to recommend one or more cities to tourists.

2.3.2. Preference Elicitation and Customisation

The user might not like to just follow the whole of a suggested trip itinerary by a recommender system. Soo and Liang presented a travel agent [62], that allowed users to elicit their travel preferences and constraints. If the travel agent failed to find a valid trip plan satisfying those, the user was asked to adopt a solution liberating any of her constraints, like removing one of the spots she wanted to visit or increasing her budget. After the violations are resolved with negotiations, a valid trip plan was shown to be followed by the user.

Other than customising with negotiations, users can already be provided with recommendations personalised to themselves. For the tourist guide by Kramer et al. [38], tourist profiles gathered according to tourist interests were checked for similarities to find whether construction of individual tours with varied tourist interests was necessary. Otherwise, if the tourist profiles were not diverse enough, developing a few standard numbers of tours equal to the number of prototypical tourist interest profiles would be sufficient. It was shown that a great number of profiles were different from each other, which showed the importance of personalisation. Moreover, tourist tracking showed that despite being attractive, some areas were rarely visited by people, while most of the tourists visit the same places. Thus, they argued that proper marketing and management could be used to distribute tourists at a destination, again pointing at the importance of personalisation.

Zheng and Xie worked on two types of travel recommendations — generic and personalised [73]. The generic one modelled other users' location histories using a tree-based hierarchical graph and recommended the most interesting travel locations popularly visited by people in a geo-spatial region. The personalised RS took the concerned user's travel preferences into account. INTRIGUE [2] was developed for recommending varied sightseeing destinations and itineraries, considering the possibly varied interests of tourist groups, like families with children and elderly. Other than the unique listing, the suggestions for adults, children or impaired could be separately listed by the system (Figure 2.5). The user could also get some generic recommendations of tourist attractions, if she preferred not to register [3].

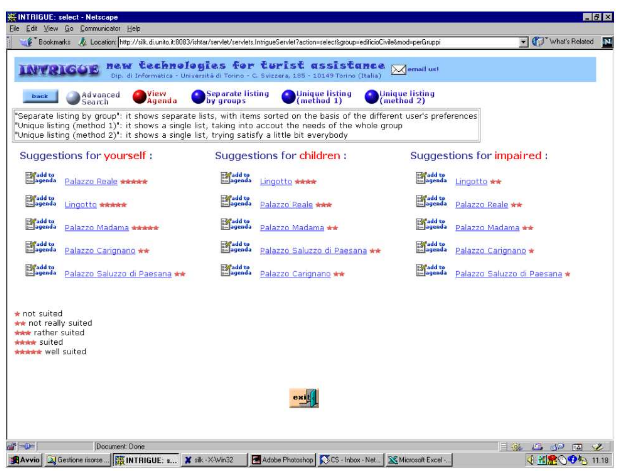


Figure 2.5.: An illustrative separate recommendation listings for different groups [2]

A way of generating personalised recommendations for a tourist is by getting their travel preferences. Yu and Chang proposed a prototype [71] to plan personalised tour for a tourist allowing them to elicit a range of preferences. The travel preferences included preferred food type, lunch and dinner timing, maximal number of visiting spots, ending time for daily activities, and hotel class. They were utilised to find matching recommendations for restaurant, hotel and sightseeing, all required to plan a tour.

In our application, a user is able to elicit her preferences by specifying the type of cities she wants to visit, regions of the world she would restrict herself to go to, and also the maximum duration she can afford for the whole trip. Destinations are then recommended after fulfilling the preferences and constraints.

2.3.3. Context Factors

Herzog et al. argued that while solving the TTDP, it is not enough to find the best solution from mathematical point of view [35]. Other than choosing the sequence of destinations with the best scores, they pointed on the need for considering the contextual factors as well while finding a solution. It included aspects like weather, time of day, and previously visited POIs. Sometimes, amongst the POIs displayed in the itinerary, some might be temporarily closed, or the current time might not be apt to visit there due to different reasons like bad weather. Descriptions can be provided if some of the selected attractions can not be included in the itinerary due to unavailability or other reasons [3]. Wu et al. took weather forecast into account while composing tour schedules, making them adaptive to probabilistic changing weather [9]. They designed an approximation algorithm based on greedy search and neighborhood search algorithms, taking user preferences of origin and final destinations, start and end time, and time restriction for visiting each location. A schedule tree was formed having the starting point as its root. The next destinations shown by the subsequent nodes of the tree were based on the weather at the time when visiting the current destination was finished. In other words, there were n possible paths to take from one location, where n was the number of weather types taken into account. Figure 2.6 shows an exemplar schedule tree adaptive to weather change.

The sophisticated tourist guide system, Guliver's Genie [46], designed by O'Hare and O'Grady, collected user information from their profiles and utilised user contexts like position and orientation to deduce the mental state of the tourists in terms of their beliefs, desires and intentions. The personalised, multi-media-enhanced information about various attractions in a timely manner was then wirelessly transmitted and displayed on personal digital assistants of the tourists. Cheverst et al. also took into account users' personal contexts like the current locations and the refreshment preferences, and environmental contexts like the time of the day or the opening times of attractions to create personalised tours [14].

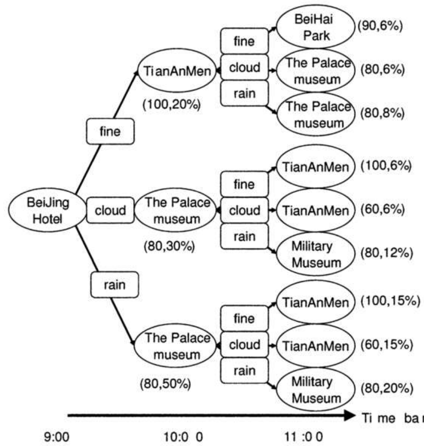


Figure 2.6.: A schedule tree instantiating a tour schedule adaptive to weather change [9]

2.3.4. Duration of Stay

A sub-problem of TTDP is to find the duration of stay at each of the destinations to be recommended to the tourists.

For user modelling and personalisation in case of different applications like systems recommending composite activities, one of the important tasks can be analysing the duration usually spent by people on various types of activities. Melià-Seguí et al. analysed the check-in patterns of users in LBSNs and derived the duration spent by them at different point of interests for various activities [44]. This way they tried to improve the recommended duration for the different user activities.

In CTRSs, the sum of duration of all regions recommended in the composite trip must not exceed the maximum possible duration of the trip. Herzog and Wörndl [36] formally formulated it as in Equation 2.1.

$$\sum_{i=1}^{n} d_i \cdot x_i \leq D \tag{2.1}$$

where n is the number of regions, d_i is the duration of stay at each region, x_i having 0 or 1 as possible values denotes whether each region is chosen to be added to the sequence to form the composite trip, and D is the maximum duration for the trip.

To determine the optimal duration of stay per region, the CTRS to recommend multiple countries in a trip as introduced by Herzog and Wörndl [36] assumed that the travellers stayed longer in places with higher significant values. Otherwise, choosing maximum duration for lesser number of regions would decrease the diversity. This value for each region was also lowered with every additional week of staying at that region, where the lowering factor depended on the maximum duration of the whole trip specified by the user. In this way, they ensured diversity also for shorter trips.

CTRSs for POIs can recommend composite POIs with [16, 62] or without [68] constructing timed paths. Those with the timed paths suggest the time of arrival and time to leave the POI along with the sequence of POIs. This determines the duration of stay at each POI. Figure 2.7 and Figure 2.8 show exemplar sequences of POIs to be visited without timed paths and with timed paths, respectively.

Figure 2.7.: An exemplar sequence of recommended POIs without timed paths [68]

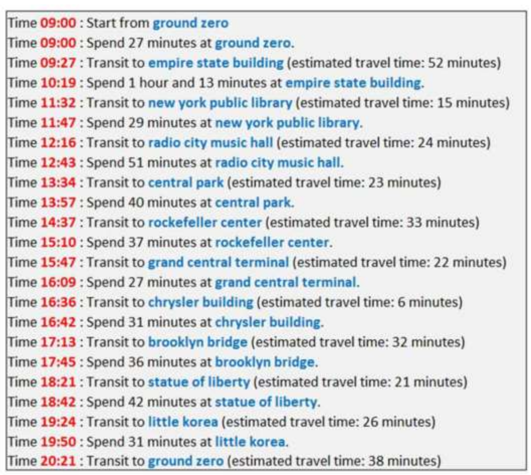

Time 09:00 : Start from ground zero
Time 09:00 : Spend 27 minutes at ground zero.
Time 09:27 : Transit to empire state building (estimated travel time: 52 minutes)
Time 10:19 : Spend 1 hour and 13 minutes at empire state building.
Time 11:32 : Transit to new york public library (estimated travel time: 15 minutes)
Time 11:47 : Spend 29 minutes at new york public library.
Time 12:16 : Transit to radio city music hall (estimated travel time: 24 minutes)
Time 12:43 : Spend 51 minutes at radio city music hall.
Time 13:34 : Transit to central park (estimated travel time: 23 minutes)
Time 13:57 : Spend 40 minutes at central park.
Time 14:37 : Transit to rockefeller center (estimated travel time: 33 minutes)
Time 15:10 : Spend 37 minutes at rockefeller center.
Time 15:47 : Transit to grand central terminal (estimated travel time: 22 minutes)
Time 16:09 : Spend 27 minutes at grand central terminal.
Time 16:36 : Transit to chrysler building (estimated travel time: 6 minutes)
Time 16:42 : Spend 31 minutes at chrysler building.
Time 17:13 : Transit to brooklyn bridge (estimated travel time: 32 minutes)
Time 17:45 : Spend 36 minutes at brooklyn bridge.
Time 18:21 : Transit to statue of liberty (estimated travel time: 21 minutes)
Time 18:42 : Spend 42 minutes at statue of liberty.
Time 19:24 : Transit to little korea (estimated travel time: 26 minutes)
Time 19:50 : Spend 31 minutes at little korea.
Time 20:21 : Transit to ground zero (estimated travel time: 38 minutes)

Figure 2.8.: An exemplar sequence of recommended POIs with timed paths [16]

Sometimes, the user of a travel itinerary recommender system application is given the option of stating the minimal period of staying at each designated spot as part of her preference [62]. This constraint of minimal staying period is considered by the system, when the final itinerary is recommended. Otherwise, in the basic case of assigning the number of days to stay at each recommended region for a TTDP, the mean [18] or the median [26] duration of stay considering all previous users could be used directly. Herzog et al. contended that if the distribution of the duration of stay showed a high variance, the recommended number of days for staying at a region should be personalised [35]. If the traveller type was determined by either self-assessment or after the user provided her check-in history, the duration recommended could then be modified depending on the habits of travellers with similar habits [18].

While building a trip itinerary, Choudhury et al. aggregated the actions of each traveller at each POI involved [16]. From the multiple visits of a user at a POI, the duration of the longest visit was chosen to be the visit time for that user. The visit time at each POI in the itinarary was then set to be the visit time closest to the 75th percentile among all users at that POI.

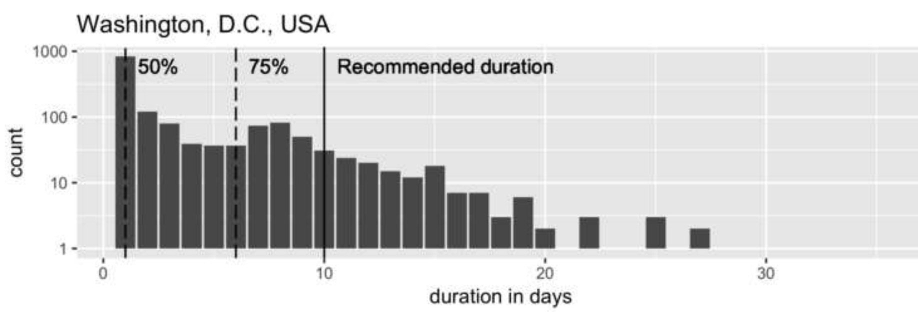


Figure 2.9.: Selection of duration of stay to be recommended at a city [23]

The paper by Dietz and Wörndl [23] specifically focused on the problem of determining the personalised amount of item consumption to be recommended, in the domain of destination recommendation. To find out the number of days a tourist should stay in a city, both the usual number of days of stay for the tourist in different cities and the typical time spent by tourists in the concerned city were considered. The quantiles of the previously visited cities were calculated to find the mean value. Next, the distribution of duration of stay in the concerned city by all the available tourists was determined. Finally, the number of days suggested to the current tourist for staying at a city was equaled to be the number of days corresponded by the mean percentile of the user when applied on the distribution of duration for all users in that city. The authors described an example scenario, where a duration of 10 days of stay at Washington, D.C. was recommended to the current traveller, where 10 days corresponded to her mean percentile of duration of stay in the previously visited cities at 91% (Figure 2.9).

For the development of our first prototype web application that recommends composite city trips to users, we calculate the mean duration of stay at a city by similar types of travellers.

2.4. Destination Characterisation

Research on quantifying the attractiveness of destinations to tourists is popular, leading to the importance of destination characterisation. Collecting information from survey data or from hotels, where tourists check in requires significant efforts and cost. Thus, publicly available photographs were used by Paldino et al. [49] to study HMPs in different cities across the world. The number of photographs captured by residents, domestic tourists or foreign tourists was used to find the global attractiveness of cities in different parts of the world (refer to Figure 2.10). Sobolevsky et al., on the other hand, analysed economic behaviours to find city attractiveness to foreign visitors [60]. After the characterisation, suitable destinations were recommended to different tourists.

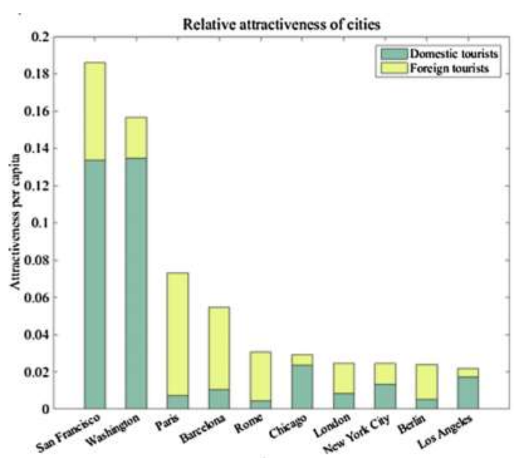
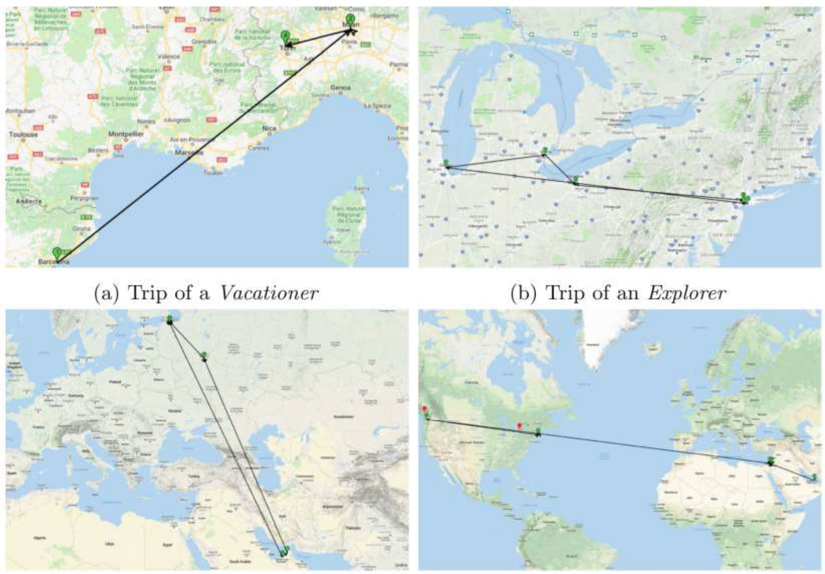


Figure 2.10.: Attractiveness of 10 cities to domestic and foreign tourists [49]

2.4.1. Clustering or Classification in Recommender System Applications

Clustering or classification result in easier understanding of patterns, rules, causes, and consequences for events involving similar elements. Designing a DRS requires us to satisfy different types of travellers understanding their preferences. This marks the need to cluster tourist types, trip types, destination types, user interests, etc into new clusters or classify them into predefined groups.



(a) Trip of a *Vacationer* (b) Trip of an *Explorer*

(c) Trip of a *Voyager* (d) Trip of a *Globetrotter*

Figure 2.11.: Visualising exemplary trips by different traveller types [19]

Kramer et al. calculated the similarities between different tourist profiles and clustered them to find diverse groups [38], showing the importance of personalised recommendations for tourists. On the other hand, Massimo and Ricci [43] learned the behaviours of the tourists via inverse reinforcement learning and clustered the similarly behaving ones to get different tourist types. This was followed by characterising each type to generate similar next POI recommendations for users belonging to the same cluster [43]. In our previous paper [19], we also discovered different tourist types by determining the various types of trips they prefer to follow (Figure 2.11). For this, trips were clustered after they were extracted from raw check-in data from LBSNs and the TMPs were analysed to characterise them with unique features. The purpose of trips were classified into different categories such as home, work, or restaurant by Griffen et al. using vehicle GPS data [32]. After the automatic collection of data with latitude, longitude and time of day information, trip stops were found, and Dbscan was used to identify POIs and aggregate them in each cluster or POI type. Wu et al. rather focussed on activities humans get involved in for observing the underlying HMPs. They used check-in data from social media to find transition probabilities between activities varying with time, and studied the different trip patterns so found [69]. They found three trip patterns, depending on if both the predecessor and successor activities are associated with static locations, both are associated with stochastic locations, or hybrid. Dietz and Weimert presented a data model structure for the user-submitted trips defining information like blocks or number of days for the trip and classified the travel items based on wOndary's categorisation scheme [22]. This was used for user modelling, implicitly finding the preferences using user's past trips, and could be used for recommending personalised trips anywhere in the world.

Travel regions were clustered by Sen and Dietz using geo-tagged tweets [56]. Spatial clustering was applied on the mined touristic regions, that resulted in four clusters on the highest level. Based on similar temporal behaviours in terms of tourism, Hawelka et al. clustered countries into three groups comparing the traveller mobility patterns in each of them [34]. Sparks et al. also clustered regions to reflect the temporal similarities between them [63]. However, the regions were more granular in the city levels, same as the granularity chosen in also other works [20, 60]. Going even deeper, functional regions of a city were also clustered [15, 53, 72]. Yuan et al. [72], as an example, used taxi-cab-generated GPS trajectory datasets and POI datasets to identify and cluster functional regions such as educational, entertainment, or historic interest areas in a city.

SAMAP, the tool to help tourists plan POI visits in a city was user-oriented [13]. For favouring personalisation, users' interests were classified into six categories — gastronomy, museums, architecture, art style, open spaces and leisure — each having further sub-categories.

Other RS applications involved clustering or classification algorithms for different other purposes. To illustrate, in the web-based travel itinerary planning application [52] developed by Rani et al., POIs chosen by the user to be visited were clustered into groups so that the POIs belonging to the same group could be explored on the same day. Sun and Lee's route recommender system [64] clustered road segments based on attraction values and the geographical closeness of the POIs in the road segments, so that the most attracctive road segments were included in the recommended route. Kramer et al. classified the tourist attractions into classes, which could then be shown to the tourists in three different organised ways — hierarchical tree structure, inspirational images, or main categories [38]. The tourists could then express their interests choosing the classes and the belonging attractions from either of the ways they opted to go with. The DRS by Cao et al. [12] clustered geo-tagged web photos and found the representative images of each of the groups. Later, a user's preferred destination images could be characterised and compared with the representative images of the clusters, so that destinations whose images belong to the same cluster could be recommended to her.

We cluster the trips after mining them from geo-located tweets, and then define traveller types following the corresponding trips. The user of our RS web application is later provided with the option of choosing one of the types she resemble the most.

2.4.2. Data Sources and Features for Travel Region Characterisation

Characterising travel regions is required to know about the places better, so that the features of the places can be compared with the requirements or preferences of the tourists to recommend them with the optimal destinations to visit.

Mobile phone data could be used not only for HMP analysis [30, 57], but also for deriving structure of cities. Sevtsuk et al. [57] used mobile data and considered demographic, economic, and environment factors for characterising the different regularity patterns at different cities. Different dynamic properties like average distance between the inhabitants in the city throughout a day were calculated from the mobile data traffic and were used for the characterisation of cities by Louail et al. [42]. Grauwin et al. also analysed mobile traffic data to reveal locations of cities where people are most actively found during different points of a day or different days of a week [31]. The weekly patterns so found were then used to characterise dynamics at a city and different city signatures were compared to find the influencing factors, highlighting the nature of the areas such as commercial or recreational or residential.

Tourists leave digital footprints when they visit cities [29]. Passive footprints can be tracked through their use of mobile phone for making calls or sending messages, whereas active ones can be found when the users themselves expose their location-based data through publicly sharing photos or even sensor measurements. Many research papers [15, 45] used social network data to characterise and compare the HMPs in different cities. Cranshaw et al. [15] utilised social media contents generated by the residents of a city to study the dynamics, structure, and character of a city on a large scale by analysing the patterns of people's movements through the city. Silva et al. also used social media contents to characterise and compare countries, cities or regions of a city, but they focussed specifically on users' food and drink check-in [58].

To quantify attractiveness of cities in Spain to foreign visitors, datasets formed from different activities or sources like bank card transactions, geotagged photographs, and tweets were used by Sobolevsky et al. [59]. For all the datasets and for different ways of defining cities, the attractiveness of a city turned out to be super-linearly scaled with its population size. Louail et al. also showed that the number of activity centres in cities scaled sublinearly with the corresponding population sizes [42]. However, Bettencourt et al. showed that the parameter population size acted as proxy to diverse socio-economic mechanisms for scaling different performance parameters of cities and discussed systematic ways of forming science-based matrics to rank and access features of cities like indices for creation, violence or wealth [8]. Unlike most other papers earlier, that calculated the generic per-capita values as performance measures of the cities [5], they also quantified the measures locally.

Travel regions were characterised with different features like mobility rate, radius of gyration, destination diversity, and inflow and outflow of travellers by Hawelka et al. [34]. Shifting the discussion more on the city level, Dietz et al. characterised cities based on availability of venues and other travel features like cost of travel, temperature, precipitation and possible activities in a data-driven approach [20]. Arribas et al. also quantified cities providing scores and world-wide ranks based on their performances across multiple dimensions, viz., "Economy", "Research & Development", "Cultural Interaction", "Livability", "Ecology & Natural Environment", and "Accessibility", using unsupervised computational neural networks called Self-Organising Maps (SOMs) [4]. Region characterisation studies [58] identified users' individual preferences like the taste and temporal habits, which were analysed to find the cultural distances and draw cultural boudaries between different places around the world. The temporal patterns of 135 cities around the world were also studied by Sparks et al., but temporal behaviours more specially meant the operational hours of restaurants and retail. The clustering of the cities by them were based on the unique social and geophysical factors associated with the cultural regions they belonged to [63]. In case of INTRIGUE [2], the destinations were characterised into categories with multiple properties like historical value, scientific value, and eye catchiness. Information of different other types like weather, temperature, and time of day and also POI categories were used by Massimo and Ricci to characterise the POI trajectories used by the RS designed by them for next POI recommendation [43].

In this thesis, we characterise 138 cities considered by us with different features — geographical coordinates (latitudes and longitudes), physical properties (average temperature and average precipitation), and the frequency of different types of POIs (arts & entertainment, food, nightlife, and outdoors & recreation) present in the cities.

Multiple scores in the aspects of natural beauty, historical interest, cultural event, and business opportunities were assigned to individual cities by Geem et al. [28], unlike single score for each in case of OP. Total score for a path differed based on the goal. In our work, we also assign multiple scores to the same set of destination cities. They are then selectively recommended to different types of travellers based on the different scores assigned to them.

2.4.3. Functional Regions

Close regions tend to be visited by the tourists in one trip. This was also drawn as a conclusion in one of the papers by Hawelka et al., where they proved that people preferred to visit neighbouring countries more than those far away [34]. They built a worldwide country-to-country directional network to estimate flows from one country to another and found out which countries were visited together and in what order. Dietz et al. also listed the countries visited together most frequently [18]. However, tourists do not always constraint themselves to visit regions in terms of different countries. Hu et al. extracted the co-occurance of cities by reading news articles to find the semantic relatedness between them [37]. They also analysed the impacts of geographical distances between the cities on the semantic relatedness and observed varied distance decay effects in the extracted semantic relatedness between cities. Sen and Dietz contended that political boundaries have a strong influence on travelling behaviours of people [56]. Therefore, the functional regions, which can be considered to be geographical units connected socially, politically and by means of transportation, are important to be considered.

The distinct geographical areas of a city were discovered by the clustering algorithm of Cranshaw et al. [15], which they called *Livehoods*. Depending on the extent of intersection between the *Livehoods* found and the municipal borders, they were termed as *split*, *spilled* and *corresponding*. The partitioning algorithm by Ratti et al. was tested on the areas within Great Britain [53]. Other than the geographically cohesive regions that corresponded well with the administrative regions, they also discovered some spatial structures that were previously only speculated in the literature. Communities detected by Hawelka et al. using twitter data [34] corresponded well with the results from some other papers [53, 61] where mobile phone data was used.

The smallest unit of travel regions considered in our research is in the level of a city. We divide the world into 10 sub-regions and map each city in our dataset to one of these regions. A user chooses one of these regions as her destination, and our DRS recommends composite cities within the selected region.

2.5. Summary

In this chapter, we discussed related work on human mobility and its more specific counterpart, traveller mobility. They were analysed by researchers in the past using wide variety of data sources, amongst which, LBSNs are recently more popular. Patterns drawn from the analyses are basis to different other studies including designing of trip recommender systems.

Next, we discussed one of the earliest area of research in the tourism domain, electronic tourist guide. This helps tourists travel in and around a monument or even a whole city more easily using the services provided by it. We also reviewed papers recommending different tourist packages suitable for different types tourists. Varied tourist recommender systems recommending destinations on different levels were also explored.

Furthermore, we talked about composite tourist recommender systems that recommend a sequence of trips, with or without modelling the problems using TTDP. Focusing more on TTDP, we discussed how they were solved by different researchers with various mathematical approaches including TSP, OP or KP. Different algorithms used for recommending routes from an origin to a destination were referred. Systems that help in planning and scheduling trips by recommending trip itineraries were discussed along with the different functionalities offered by them. We also talked about the importance of personalised recommendations, elicitation of preferences by users, and the importance of context factors in RSs. Finally, we explored the literature on the duration of stay at each destination recommended to a tourist.

Additionally, we pointed on the importance of destination characterisation for developing DRSs. Varied range of data sources and features utilised by researchers for characterising travel regions were discussed. We also explored the need of clustering on various sub-domain levels associated with the development of tourist recommender systems. Finally, we focused the discussion on discovering regions functioning together, so as to find an optimal granularity level for destinations to be recommended to tourists.

3. Data Engineering for Content-based Recommendation

In this chapter, we collect and analyse the data to be used for the content-based recommendation, to be discussed in later chapters. It mainly includes the trips and the cities. We elaborate on their initial characteristics and adapt it with further modifications. We also divide the world into travel regions, as discussed in this chapter. The cities in the database are mapped to these world regions.

3.1. Dataset I — Cities

We consider 138 cities to be recommended to travellers of different types. We characterise each of the cities with different features, to establish similarity scores between them.

3.1.1. Characterisation of Cities

The cities are attributed with different features. Those involving the frequencies of venues, with different types of touristic values located in the cities, are called *arts & entertainment*, *food*, *nightlife*, and *outdoors & recreation*. The types of these venues are based on four of the Foursquare venue categories[1]. We divide each of the frequency values by the total venue counts of all four types considered in the respective cities. This is done to avoid bias due to the varied range of venue counts in different cities and check the prevalence of the different types in each of the cities. Finally, for each city, stemming from a number derived from Numbeo[2], the *cost index* values are normalized between 0 and 100. The following provides short descriptions about the *cost index* and these frequency-based features, which will further be used up later:

- *Cost index* — It is the average price a typical tourist need to pay for buying goods or services at a certain place, i.e., a certain city, for our case.

[1]https://developer.foursquare.com/docs/resources/categories
[2]https://www.numbeo.com/cost-of-living/

- *Arts & entertainment* — It indicates the frequency of places in a city offering services with arts and/or entertainment values. An art gallery, a circus, a comedy club, a salsa club, a concert hall, a science museum, and a multiplex are some examples for the same.
- *Food* — It is the count for the restaurants, cafés, or rather, any food or drink shop in a city.
- *Nightlife* — This is the frequency of places in a city involving pubs or nightclubs.
- *Outdoors & recreation* — This sums up the number of places in a city that includes activities for recreation in a natural setting. Soccer fields, forests, national parks, ski or skydiving areas, beaches, gardens, and scenic lookouts are some exemplary locations that fall under this category of places.

The original dataset also includes the average precipitation and average temperature at each of the cities [20]. However, we decide not to consider them further because they are the annual mean values that can vary a lot at different seasons. Furthermore, we discard the *venue count* and the *venue area*. The venue count is not chosen because we have already accounted for the frequency-based features, which are also counts specific to different types of venues. On the other hand, for the prototype composite-city-trip recommender system described in this thesis, considering the size of the cities are out of scope. So, we also eliminate the feature venue area, which accounts for the same.

Besides these, the dataset also have the center geographic coordinates (latitude and longitude) for each of the cities. We use the *Haversine formula* [11] for calculating the great-circle distances between each pair of city locations.

Additionally, we map each of the cities to corresponding countries and continents they belong to. They are also mapped to some customised regions we divide the world into. We discuss these regions next.

3.1.2. World Regions

We divide six continents of the world (except Antarctica) to 10 global regions, viz., North America (NAm), South America (SAm), North Europe (NE), Southwest Europe (SWE), Southeast Europe (SEE), North Africa (NAf), South Africa (SAf), West Asia (WAs), East Asia (EAs) and Oceania (O). Figure 3.1 displays these regions on the world map. Table 3.1 shows the 62 countries that comprise the 138 cities in our database.

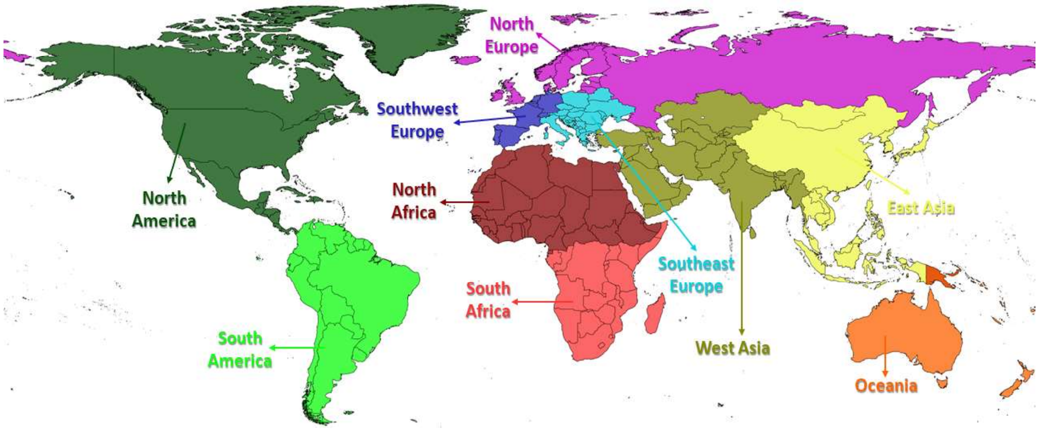


Figure 3.1.: World map annotated with our customised world regions

Table 3.1.: Countries with the cities in our database

Regions	Countries
North America	Canada, Mexico, United States
South America	Argentina, Brazil, Chile, Colombia, Ecuador, Paraguay, Peru, Uruguay, Venezuela
North Europe	Denmark, England, Estonia, Finland, Iceland, Ireland, Latvia, Lithuania, Norway, Russia, Scotland, Sweden, Wales
Southwest Europe	Belgium, France, Germany, Luxembourg, Netherlands, Portugal, Spain, Switzerland
Southeast Europe	Austria, Czech Republic, Greece, Hungary, Italy, Poland, Romania, Slovenia
North Africa	Egypt, Ghana, Morocco, Nigeria
South Africa	Kenya, Uganda
West Asia	India, Israel, United Arab Emirates
East Asia	Cambodia, China, Hong Kong, Indonesia, Japan, Philippines, Singapore, South Korea, Taiwan, Thailand, Vietnam
Oceania	Australia

3.2. Dataset II — Trips

We utilise the check-ins from LBSNs to learn about travel mobility patterns of people with a motive to improve DRSs. Trips are identified out of the check-ins from Twitter using a data-mining approach [18]. After the characterisation of these trips, some of them are removed based on their poor qualities. This is followed by splitting up of the remaining trips into 10 subsets based on the home regions of their corresponding travellers. We elaborate the characterisation and elimination of trips in Subsection 3.2.1 and Subsection 3.2.2, respectively.

3.2.1. Characterisation of Trips

A trip is annotated with different characteristic features broadly categorised on the basis of the trip's mobility, the traveller for the trip, and the blocks visited during the trip. They are elaborately described below with the sub-categories:

(a) Mobility-based features — the features that help in analysing mobility patterns of travelling in the respective trips

- *Travel duration* — It is the number of travel days given by the difference in dates of the first check-in and last check-in of the identified trip.
- *Mean displacement* — This measures the distance between the traveller's home location and the mean position of the places visited by her in the trip. Mean displacement provides how far away from home the trip is.
- *Radius of gyration* — Radius of gyration gives us how far the movement within the trip is. It is calculated as the mean distance between the mean location of the trip and all the other check-in locations of the same.
- *Number of blocks* — A block here can roughly be identified as a city. So this parameter gives us the number of distinct cities visited during the trip.
- *Number of countries* — This provides us with the number of distinct countries visited during the whole trip.

(b) Traveller characteristics — the features that include information about the travellers of the identified trips

- *Traveller home location* — Each trip has the home location of the traveller in terms of its geographical coordinates. We map it to a city, country, continent, and finally to one of our 10 customised home regions.

- *Traveller home ratio* — This provides the ratio of the number of check-ins of the traveller at her home country to that at a location outside the home country. Higher home ratio signifies that the user has stayed more at her home than she has travelled somewhere, which is expected to be the case, if we have correctly extracted the trips.

(c) Block characteristics — A trip involves visiting one or multiple places. The places are locations in the same or different cities, countries or regions. We consider a city to be a block. Therefore, we also take into account the five previously discussed features of each city visited during a trip to characterise it. They are *cost index*, *arts & entertainment*, *food*, *nightlife*, and *outdoors & recreation*. These features, which we can call as the city-based features from now on, signify the kind of places visited during the trip. Since there can be multiple blocks in a trip, we calculate the average values for each of the city-based features within the trip. The average value F_i of feature F for trip i is calculated using Equation 3.1.

$$F_i = \frac{\sum_{j=1}^{n} \left(n_{b_j} * F_{b_j} \right)}{\sum_{j=1}^{n} n_{b_j}}, \tag{3.1}$$

where n is the distinct number of blocks within the trip, n_{bj} denotes the number of times block b_j is visited within the trip, F_{bj} denotes value of the feature F for block b_j, and F designates AE (*arts & entertainment*), FD (*food*), NL (*nightlife*), OR (*outdoors & recreation*), or CI (*cost index*).

3.2.2. Elimination of Outlier Trips

After characterising the trips, we assign threshold values to some of the features in the dataset. Based on that values, we remove outlier trips in order to improve the quality of our data. Here, we discuss which features we use as bases for the quality assurance of the dataset and how they benefit.

Ensuring non-existence of typical business trips

To eliminate the typical business trips, we consider only those having at least seven days of travel duration. We continue with $102,440$ trips after this step.

Ensuring correct home location of traveller

We assume that the calculated home ratio of a traveller should be high enough for us to say that we correctly identified the home location of the traveller. This is because, yearly, the user is usually supposed to stay at home more often than she travels.

We consider the threshold value for this parameter to be 0.3, and thus remove all the trips with its traveller's home ratio lesser than that leaving us with 95,290 trips to proceed further with.

Ensuring no huge shift of mean values of features due to infrequent existence

We consider the mobility-based features and delete the outlier trips existing exceptionally with very infrequent characteristic values. We visualise the distribution of each feature by plotting a histogram for it, remove the trips with the least frequent values, which are all those greater than chosen independent threshold values, and repeat the same steps with the remaining trips for the rest of the features. We can notice from the histograms that the height of the bars representing the frequencies keep on decreasing until it merges with the x-axes. We chose the threshold for each feature to be the value, beyond which we can barely notice the bars, after visualising the corresponding histogram. We keep a fixed bin size of 30 for all the histograms.

We start with the feature *number of blocks*, and plot the histogram considering 95,290 trips. Looking at the plot, we decide to eliminate those trips having *number of blocks* greater than 25 and are left with 95,085 trips after doing that. Next, we proceed with the same steps for the other features, viz., *number of countries*, *radius of gyration*, and finally *travel duration* as well. Figure 3.2 shows the histograms of the features after each round of elimination. Table 3.2 tabulates the threshold values and the number of remaining trips after each step. We carry on with the 93,955 trips for the future operations.

Table 3.2.: Thresholds of mobility-based features for elimination of trips

Mobility-based feature	Threshold value	Number of trips after elimination
Number of distinct blocks	25	95,085
Number of distinct countries	8	94,931
Travel duration	200	94,337
Radius of gyration	10,500	94,015
Mean displacement	17,500	93,955

3.3. Summary

In this chapter, we explored the datasets we used for analysing TMPs, that followed the development of our RS application. We had two main datasets — *trips* and *cities*.

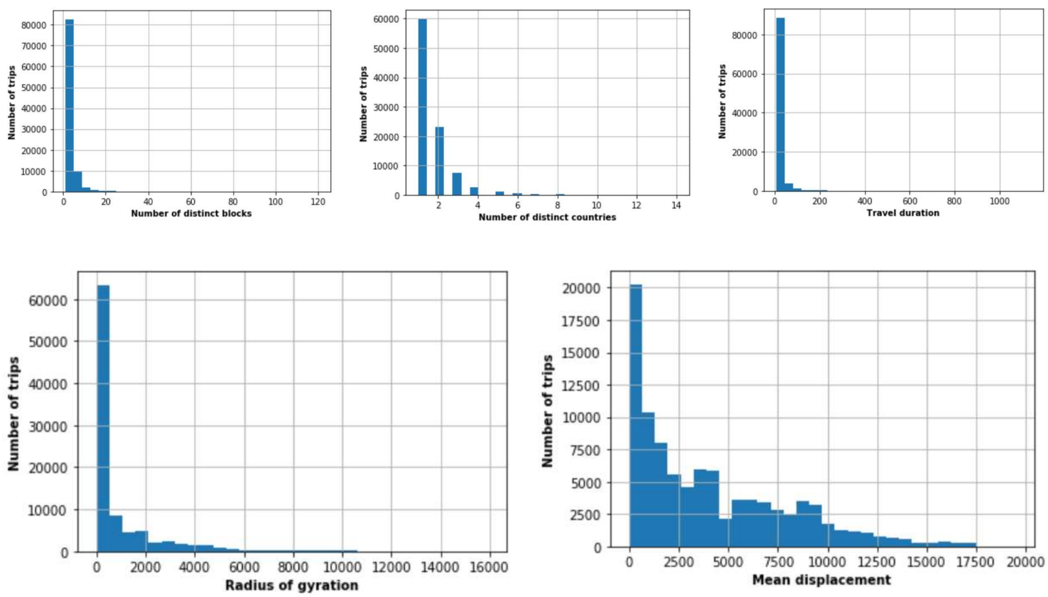


Figure 3.2.: Histograms for mobility-based features

Cities were mainly characterised with their geographical coordinates, the cost indices, and the respective frequency of different types of venues located in the cities. The geographical coordinates of all the cities were used to calculate the physical distances between each other. Next, we divided the world into 10 regions and countries were assigned to each of them. We mapped each city in our database to one of our customised world regions, based on the country they belong to.

Each trip was attributed to varied characteristics — the generic ones, those related to its traveller information, and the block-related features, also called as the city-based features. The generic characteristics were further categorised into the check-in-based and the mobility-based features. Few trips were then eliminated to assure a better quality of the dataset.

In the next two chapters, we discuss how we furthermore utilise the data engineered in this chapter. In the upcoming chapter, we talk about how we cluster trips to characterise their corresponding travellers.

4. Identification of Regional Traveller Types

This section discusses the methodologies used for clustering travellers from different parts of the world based on the trips undertaken by them. We define the characteristics of the cluster prototypes and determine their prevalence in different regions.

4.1. Defining Cluster Prototypes

The mobility-based trip features are divided into four divisions — very small ($--$), small ($-$), large ($+$) and very large ($++$). Table 4.1 shows the ranges of the corresponding feature values for the four divisions.

Table 4.1.: Division ranges for mobility-based features of the trips

Division	Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration
$--$	<8 [less than 1 week]	<1000	1	1	<1000
$-$	8-21 [1-3 weeks]	1000-2500	1-5	1-3	1000-2000
$+$	21-42 [3-6 weeks]	2500-7500	5-10	3-5	2000-6000
$++$	>42 [more than 6 weeks]	>7500	>10	>5	>6000

The $95,290$ trips left after the trip elimination step discussed in the previous chapter are divided into 10 subsets depending on the home regions of their respective travellers. For each subset, we compare the mean values of the mobility-based features to the corresponding value ranges from Table 4.1. After analysis, we come up with 10 types of travellers based on the characteristics of the trips by them.

Figure 4.1 shows the characteristic features delved down to identify different types of travellers. We provide short descriptions highlighting the main characteristics of the prototype travellers below:

(i) **Vacationers** — travel nearby (low mean displacement), visiting few cities and countries (very small number of blocks and countries) for short travel duration.

(ii) **Explorers** — similar characteristics as vacationers, but travel far from home (high mean displacement).

(iii) **Voyagers** — very distant travellers (high to very high mean displacement) for short travel duration.

(iv) **Globetrotters** — covers many cities and countries in a trip (high number of blocks and countries).

(v) **Eurotrotters** — travellers from Europe, travelling many nearby cities and countries (low mean displacement but high number of blocks and countries).

(vi) **Stopping Voyagers** — those travelling far from home (high mean displacement) for a short travel duration, but travelling far within trip (large radius of gyration).

(vii) **Long Stopping Voyagers** — Stopping Voyagers going for long duration trips.

(viii) **Hop-trotters** — those travelling many cities and countries, far from home (high mean displacement) for a short travel duration, also not travelling far within trip (small radius of gyration).

(ix) **Long Hop-trotters** — Hop-trotters going for long duration trips.

(x) **Long Explorers** — Explorers going for long duration trips.

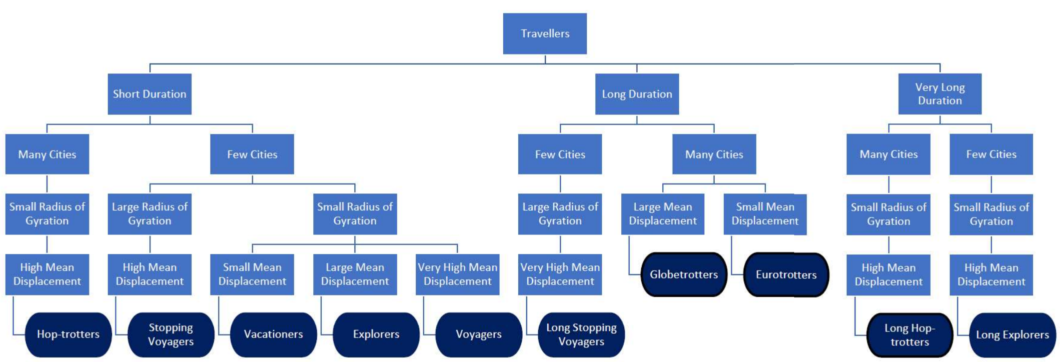


Figure 4.1.: Analysis of mobility-based features for traveller types

Table 4.2 tabulates the characteristic value ranges for the mobility-based trip features.

Table 4.2.: Traveller cluster prototypes

Traveller types	Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration
Vacationers	−	−− to −	−	−	−−
Explorers	−	+	−	−	−−
Voyagers	−	++	−	−	−− to −
Stopping Voyagers	−	+ to ++	−	−	+ to ++
Hop-trotters	−	+	+	+	−
Eurotrotters	+	−	+	+	−−
Globetrotters	+	+ to ++	+	+	− to +
Long Stopping Voyagers	+	++	−	−	++
Long Hop-trotters	++	++	++	+	−
Long Explorers	++	+	−	−	−−

In our previous papers for clustering travellers [19, 21], we did not segregate trips by travellers from different home regions before identifying the travel mobility patterns. Moreover, in the first paper [19], we considered trips obtained by analysing check-in data only from Foursquare, unlike this thesis, where we consider Twitter data. In the former case [19], we obtained four types of travellers, viz., *vacationers*, *explorers*, *voyagers*, and *globetrotters*. These types correspond well with the prototype traveller types obtained in this thesis work, designated by the same names. In the second paper [21], the three clusters of travellers obtained by mining trips from Twitter, viz., *domestic*, *globetrotters*, and *distant vacationers* had the similar characteristics as *vacationers*, *globetrotters*, and *voyagers*, respectively obtained in the current work. However, this time, we have obtained more prominent clusters in addition to those discovered in our earlier projects.

4.2. Analysis of Clusters from Each Region

In this section, we discuss clustering, considering subsets of trips by travellers from different home regions. At first, we normalise the mobility-based feature values between 0 and 1 using min-max scaling, for each subset. This is done in order to avoid clustering bias due to varied ranges of the different features. Next, we use these features and apply the familiar k-means clustering algorithm to get different trip clusters within each subset. The suitable number of clusters is chosen using silhouette index. After clustering, the mean values for the different features of the clusters are compared with the feature ranges mentioned in Table 4.1. The prototype clusters are then identified within each trip subset from Table 4.2. This is how we detect which types of travellers are present in different regions.

We now specifically look into the clusters within each of the 10 world regions. We show the plots of the average silhouette scores obtained by clustering trips into two to nine clusters for each trip subset (Figure A.1–Figure A.10). We choose the number of clusters by finding after which number of clusters we obtain the highest drop in scores. As an example, in the silhouette plot of region *NAm* (Figure A.1), we can see the highest difference in average silhouette scores between five and six clusters. Therefore, we choose to cluster the trips by the travellers with home at North America into five groups. Table 4.3 shows the total number of trips that are there to be clustered for travellers from each region and the number of clusters decided to be chosen for clustering based on the corresponding silhouette plots.

Table 4.3.: Frequency of trips and number of clusters for different regions

Regions	Number of trips	Number of clusters
North America	43580	5
South America	7392	5
North Europe	14490	5
Southwest Europe	10468	4
Southeast Europe	3677	4
North Africa	372	6
South Africa	216	6
West Asia	1776	3
East Asia	10127	4
Oceania	1857	5

The mean values of the mobility-based features of clusters within each trip subset can be seen in Table B.1–Table B.10. The traveller types identified after comparing them with the prototype clusters are listed in Table C.1–Table C.10.

4.3. Synopsis of Clusters for All Regions

In this section, we summarize the traveller types obtained in different regions. To exemplify the prevalence of these types throughout the world, we analyse the frequencies of the prototype traveller types in the different regions as shown in Table 4.4.

Table 4.4.: Frequency of traveller types in the different regions

Traveller types	Regions	Frequency
Vacationers	All except *SAf* and *WAs*	8
Explorers	*NAf*, *SAf*, *NE*, *WAs*, *O*	5
Voyagers	All	10
Stopping Voyagers	All except *WAs* and *O*	8
Hop-trotters	*SAf*	1
Eurotrotters	*NE*, *SWE*, *SEE*	3
Globetrotters	All except *NE*, *SWE*, *SEE*	7
Long Stopping Voyagers	*O*	1
Long Hop-trotters	*SAf*	1
Long Explorers	*NAm*, *SAm*, *NAf*	3

We note some key points about the existence of different traveller types in all regions below:

(a) Largest number of travellers from most of the regions seem to be *vacationers*, with the travellers from South Africa, North Africa, West Asia, and Oceania being the exceptions.

(b) From South Africa, North Africa, and West Asia, the travellers from the dataset mostly tend to be *explorers*. Moreover, the travellers from the other two regions where *explorers* are found in, viz., North Europe and Oceania, are the second highest in number.

(c) From Oceania, the maximum number of travellers are *voyagers*. *Voyagers* is the only type that is noticeable amongst the travellers from all the 10 regions considered. With the exception of North Europe, it is also the second most prevalent cluster.

(d) *Stopping voyagers* are also seen in all the regions except West Asia and Oceania.

(e) *Globetrotters*, covering many cities and countries travelling far from home for around a month, are interestingly there in all except the European regions, viz., Southeast Europe, Southwest Europe and North Europe. On the other hand, we got the cluster *eurotrotters*, the European equivalent of the *globetrotters*, visiting many cities and countries also travelling for around a month, but not far from home.

(f) The types *hop-trotters* and *long hop-trotters* happen to be seen only in South Africa, and the cluster *long stopping voyagers* is seen only in Oceania.

(g) The types *vacationers*, *voyagers* and *globetrotters* are found in both the Asian regions — East Asia and West Asia — with East Asia having one more type, *stopping voyagers*.

(h) The types *vacationers*, *voyagers*, *eurotrotters* and *stopping voyagers* are found in all the European regions — Southeast Europe, Southwest Europe and North Europe — with North Europe having one more type, *explorers*.

(i) South America and North America have the same five types of travellers, viz., *vacationers*, *voyagers*, *globetrotters*, *long explorers* and *stopping voyagers*.

(j) Both the African regions, South Africa and North Africa, have only four common type of travellers, viz., *explorers*, *voyagers*, *globetrotters*, and *stopping voyagers*.

4.4. Summary

In this chapter, we discussed the methodologies and the results of clustering subsets of trips, by travellers from each of our customised world regions. The clustering of the trips also grouped the corresponding travellers of the trips into respective clusters.

We discovered 10 prototype clusters for the types of travellers around the world, after characterising the trips followed by them. Next, we clustered the trip subsets and thus the travellers using k-means clustering into suitable number of groups. This was followed by the identification of the traveller types found in different regions. Finally, we checked the prevalence of the different prototype traveller types in different regions and analysed them.

So, we got a sum total of 47 traveller types from different regions of the world. These will act as the possible options for traveller types to be chosen from by a user of our final RS application. In the next chapter, we talk about the algorithm explaining how the application recommends composite list of cities based on user inputs.

5. Algorithmic Aspects of Recommending Composite City Trips

In this chapter, we determine the duration of stays at each city for different traveller types. This is followed by presenting an algorithm that obtains a composite list of cities adding up to form the trip to be recommended to a user. We discuss the intermediate steps in detail using other constituting algorithms, noting the inputs and outputs for each of them.

5.1. Calculation of Duration of Stays

The travellers around the world are divided into 47 clusters in two steps, first by their home region, and then by the characteristics of the trips taken. Our next task is to find personalised duration of stays at the different cities in our database for different traveller types.

At first, we calculate the initial duration of stay at a city for a traveller of certain type and from a particular home region. To do so, we compute the mean duration of stay at that city considering the trips by all the travellers of the same type having their home location in the same region. We do this for all the cities, for the different traveller types belonging to each of the 10 regions. Altogether, we obtain 47 different values for duration of stay at each city depending on the 47 traveller types. Refer to algorithm *initial-duration-of-stay-calculation* for the pseudo-code to find these initial duration of stays.

In the trips we consider, not all traveller types visit all the cities. However, we can find visits to all of the 138 cities in our database, if we consider the travellers of all types. We do not intend to omit the possibility of recommendation of any of these 138 cities to any traveller type. Thus, we recalculate the duration of stays if it is initially computed to be 0 days for a particular traveller type. We update the duration of stay at a city with the average stay by the outer group of travellers, if it is found to be zero by the current group. If it is still zero, it means that no traveller from any type belonging to the

Algorithm 1: initial-duration-of-stay-calculation

Input: City dataset (*city_dataset*),
47 trip clusters for 47 different types of travellers (*cluster_wise_trips*),
list of 10 regions (*region_list*),
traveller types in the 10 regions (*traveller_types_list*)
Output: 47 lists of cities with initial duration of stay at each city for each of the 47 traveller clusters (*cluster_wise_cities*)

```
foreach region in region_list do
    traveller_types ← find traveller types for region from traveller_types_list
    set cluster_wise_cities to empty list
    foreach type in traveller_types do
        let trips be the trips by travellers of type type, having home at region region
        cities ← city_dataset ▷ copy all cities and their features from the database
        foreach city in cities do
            duration ← 0
            count ← 0
            foreach trip in trips do
                foreach block in blocks_trip ▷ blocks_trip contains information about all
                  blocks visited in trip
                 do
                    if cityname_city = cityname_block ▷ cityname_city is name of current city,
                      and cityname_block is name of the city in current block
                     then
                        duration ← duration + duration_block ▷ duration_block is the
                          duration of stay at the current block of the current trip
                        count ← count + 1
                    end
                end
            end
            if count > 0 then
                duration ← duration ÷ count ▷ average duration of stay at the city
                  for the current cluster of travellers
            end
            duration_city,type,region ← duration ▷ duration_city,type,region is the duration of
              stay at the particular city by the particular type of travellers having
              home at the particular region
        end
    end
    store into cluster_wise_cities the cities with duration of stay at each for the
      cluster
end
return cluster_wise_cities
```

particular home region visits the city in the trips we considered. In that case, we update the duration with the mean duration of stay in the city by the travellers of the outermost group. This is done for all cities, for all the 47 types. Figure 5.1 shows the set relation between current, outer and the outermost groups of travellers. The pseudo-code of the algorithm we use to calculate the final duration of stay at a city by the travellers from different regions is as shown in algorithm *final-duration-of-stay-calculation*.

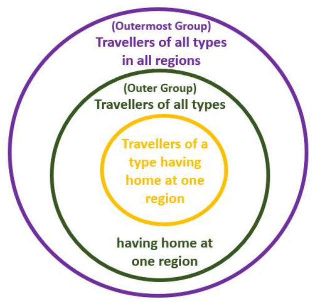


Figure 5.1.: Set representation of traveller groups

5.2. Recommendation for a User

In the previous section, we discussed how we determine the duration of stays at each city in our database that is recommended to different types of travellers. In this section, we find the recommendation for one user of a particular traveller type. In the first sub-section, we put forward the algorithm that determines the composite list of cities to be recommended to a user, depending on her constraints and preferences. It utilises other intermediate algorithms, that are introduced in the further sub-sections.

5.2.1. Finding Composite List of Cities for Recommendation

After calculating the final duration of stay at different cities for different traveller types, and after having the inputs from one user, we can utilise the algorithm *composite-city-recommendation* to find a recommendation for her. It determines the list of cities to be recommended to the user in a specific order, along with the duration of stay at each of the cities.

The intermediate algorithms used for finding the composite list are discussed in the next subsections.

Algorithm 2: final-duration-of-stay-calculation

Input: City dataset (*city_dataset*),
47 list of cities with initial duration of stays (*cluster_wise_cities*),
list of 10 regions (*region_list*),
traveller types in the 10 regions (*traveller_types_list*)

Output: 47 lists of cities with final duration of stay at each city for each of the 47 clusters of travellers (*cluster_wise_cities*)

```
foreach region in region_list do
    foreach city in city_dataset do
        calculate and store average duration of stay at the city considering all
          traveller types in the region ▷ for outer group travellers of each region
    end
end
foreach city in city_dataset do
    calculate and store average duration of stay at the city considering all traveller
      types in all regions ▷ for outermost group travellers of all regions
end
foreach region in region_list do
    traveller_types ← find traveller types for region from traveller_types_list
    foreach type in traveller_types do
        foreach city in city_dataset do
            duration_{city,type,region} ← find duration of stay at the city by travellers of
              type type having home at region from cluster_wise_cities
            if duration_{city,type,region} = 0 ▷ checking if initially calculated duration of
              stay at city is 0 days for the current traveller group
             then
                duration_{city,type,region} ← average duration of stay at the city for all
                  traveller types in the region ▷ overwrite by the average duration of
                  stay of the outer group
                if duration_{city,type,region} = 0 then
                    duration_{city,type,region} ← average duration of stay at the city for all
                      traveller types in all regions ▷ overwrite by the average duration
                      of stay of the outermost group
                end
            end
            duration_{city,type,region} ← ⌈duration_{city,type,region}⌉ ▷ taking the upper bound
              value for final duration to be recommended
        end
    end
end
return cluster_wise_cities
```

Algorithm 3: composite-city-recommendation

Input: 47 clusters of cities with duration of stays at each ($cluster\_wise\_cities$),
Distances between each pair of cities in the database ($geo\_distances$),
home region of a user ($home\_region$),
traveller type of the user *(traveller_type)*,
the region the user wants to go to *(destination_region)*,
maximum travel duration provided by the user *(max_duration_of_stay)*,
preference vector of user's preference values for city-based features
$P = [P_{CI}, P_{AE}, P_F, P_N, P_{OR}]$

Output: Ordered list of cities and its corresponding parameters, i.e., countries, geographical coordinates, and the recommended duration of stay at each city, for the user ($ordered\_city\_list$)

$city\_subset \leftarrow$ filter cities from $cluster\_wise\_cities$ based on
$home\_region$, $traveller\_type$, and $destination\_region$
▷ `refer to Subsection 5.2.2`

$city\_subset \leftarrow$ **city-score-assignment** ($city\_subset$, preference vector P)

$n \leftarrow$ number of items in city_subset $- 1$

$selected\_city\_list, total\_duration \leftarrow$ **greedy-city-selection**
($city\_subset, max\_duration\_of\_stay, n1 \leftarrow 0, n2 \leftarrow n, total\_duration \leftarrow 0,$
$selected\_city\_list \leftarrow$ empty list, $initial \leftarrow True$) ▷ `initial city selection`

$distance\_matrix \leftarrow$ calculate distance matrix for initial $selected\_city\_list$ using
$geo\_distances$ ▷ `refer to Subsection 5.2.5`

$selected\_city\_list \leftarrow$ **unfit-city-removal** ($distance\_matrix, selected\_city\_list$)

if *there **exists** at least one item in the selected_city_list* **then**

  $idx \leftarrow$ **get** last item from $selected\_city\_list$ and **find** its index in $city\_subset$

  foreach *city **in** city_selection_list* **do**

    $total\_duration \leftarrow total\_duration + duration_{city}$

  end

  $selected\_city\_list, total\_duration \leftarrow$ **greedy-city-selection**
  ($city\_subset, max\_duration\_of\_stay, n1 \leftarrow idx + 1, n2 \leftarrow n,$
  $total\_duration, selected\_city\_list, initial \leftarrow False$) ▷ `final city selection`

end

$distance\_matrix \leftarrow$ calculate distance matrix for final $selected\_city\_list$ using
$geo\_distances$ ▷ `refer to Subsection 5.2.5`

$ordered\_city\_list \leftarrow$ **selected-city-ordering** ($distance\_matrix, selected\_city\_list$)

return $ordered\_city\_list$

5.2.2. Filtering Cities According to Destination Region

From the 47 lists of cities with different duration of stays for different traveller types, we select the list based on the current user's home region (*home_region*) and her travelling type (*traveller_type*). From that list of 138 cities, we remove the ones which do not belong to the region chosen by the user as her destination region (*destination_region*). As a result, we are left with a subset of cities considered further for recommendation (*city_subset*).

5.2.3. Assigning Scores to Cities

We assign scores to the selected subset of cities comparing their city-based feature values with the user preference values ($P = [P_{CI}, P_{AE}, P_F, P_N, P_{OR}]$) as shown in algorithm *city-score-assignment*. CI, AE, F, N, and OR represents cost index, arts & entertainment, food, nightlife, and outdoors & recreation, respectively.

Algorithm 4: city-score-assignment

Input: List of cities considered further for recommendation *(city_subset)*, preference vector of user's preference values for city-based features ($P = [P_{CI}, P_{AE}, P_F, P_N, P_{OR}]$)

Output: Subset of cities sorted based on assigned scores *(city_subset)*

foreach *city* ***in*** *city_subset* **do**

$taste\_difference \leftarrow$
$$\sqrt{(CI_{\text{city}} - P_{AE})^2 + (AE_{\text{city}} - P_{AE})^2 + (F_{\text{city}} - P_F)^2 + (N_{\text{city}} - P_N)^2 + (OR_{\text{city}} - P_{OR})^2}$$
▷ CI_{city} is the cost index value for the particular $city$, and so on

$score \leftarrow \frac{1}{taste\_difference+1}$ ▷ $score$ is inversely proportional to the $taste\_difference$

$city\_score_{city} \leftarrow score$ ▷ $score$ assigned to the $city$

end

sort cities of *city_subset* **in descending order** of corresponding *city_score*

return *city_subset*

5.2.4. Finding Initial and Final Selected City Lists

We consider the subset of cities (*city_subset*) sorted based on the scores assigned to them, and greedily keep selecting the highly scored cities. This is done till the total duration of stay at the selected cities does not exceed the maximum travel duration (*max_duration_of_stay*) input of the user. This constitutes the initial list of selected cities. After the removal of some "unfit" cities as discussed in subsection 5.2.6, we add some more cities, if possible, to the selection list forming the finally selected cities. To avoid the city removal step again, we consider adding the next cities during the final selection only if duration of stay there is more than just a single day. Based on the indicator of whether the city selection process is being done initially or finally, it is decided to consider the cities with a single day stay or not. We utilise algorithm *greedy-city-selection* for both the initial and the final selection of cities.

Algorithm 5: greedy-city-selection

Input: Cities scored and sorted, considered for recommendation *(city_subset)*,
maximum travel duration provided by the user *(max_duration_of_stay)*,
lower bound *(n1)* and upper bound *(n2)* of *city_subset* within which we perform the selection from,
total duration of stay at currently selected cities *(total_duration)*,
current selected city list *(selected_city_list)*,
indicator of whether the function call is for initial selection or not *(initial)*

Output: Cities selected from city_subset matching constraints *(selected_city_list)*,
total duration of stay at selected cities *(total_duration)*

$i \leftarrow n1$ ▷ `initialising a variable` i
while $i \leq n2$ ***and*** $(total\_duration + duration_{city\_subset[i]} \leq max\_duration\_of\_stay)$
▷ $duration_{city\_subset[i]}$ `is the duration of stay at the` i^{th} `city of` *city_subset*
do
  if $(initial = True)$ ***or*** $(initial = False$ ***and*** $duration_{city\_subset[i]} > 1)$ **then**
    add *city_subset[i]* **to** *selected_city_list* ▷ `selecting the` i^{th} `city from` *city_subset*
    $total\_duration \leftarrow total\_duration + duration_{city\_subset[i]}$ ▷ `adding the duration of stay at the` i^{th} `city of` *city_subset* `to the total duration till now`
    $i \leftarrow i + 1$ ▷ `incrementing` i `by 1`
  end
end
return *selected_city_list and total_duration*

5.2.5. Forming Distance Matrix for the Distances Between Selected Cities

Distance matrix contains the physical distances between each pair of selected cities. Considering to have the distances between each pair of cities in the database, we form a list of lists as the data structure for storing the distances only between the selected cities. Each of the inner lists consists of the sorted distances from a fixed city to other cities along with their names, and the outer list contains these distance lists along with the names of the fixed cities. Refer to Figure 5.2 for visualising the distance matrix for an exemplar set of selected cities $C1$, $C2$, and $C3$ having distances a, b, and c between each other as shown in the figure.

We calculate the distance matrix (*distance_matrix*) for the initial and final lists of selected cities. This matrix is used for removal of "unfit" cities from the initially selected cities (refer to Subsection 5.2.6) and the ordering of the finally selected cities (refer to Subsection 5.2.7).

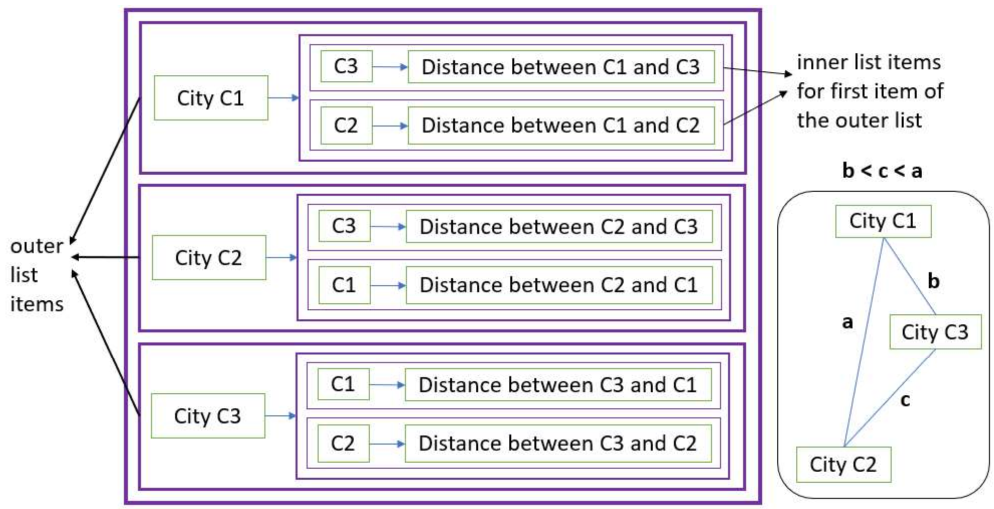


Figure 5.2.: Exemplary representation of a distance matrix

5.2.6. Removing "Unfit" Cities from Initially Selected City List

It might typically not be worthy for tourists to go to a far-away city while visiting other near ones, if the duration spent there is just one day. We call these cities to be "unfit" and consider not to include them in the list of our selected cities to be recommended to a user. Formally, we designate a city in the selection list to be "unfit" if —

(a) it has only a single day as duration of stay, and

(b) there is no other city in the selection list, which is $\leq 500km$ away from it.

We remove "unfit" cities from the initial selection as in algorithm *unfit-city-removal*.

Algorithm 6: unfit-city-removal

Input: *distance_matrix*,
current selected city list *(selected_city_list)*
Output: The list of selected cities without the "unfit" ones *(selected_city_list)*

```
foreach city in selected_city_list do
    if duration_city = 1 ▷ condition (a) for unfit city
    then
        foreach outer_item in distance_matrix ▷ looping inside the outer list
        do
            if cityname_outer_item = cityname_city then
                ▷ if outer city name matches, will loop in the inner list
                if no other city in the inner list exists with distance between that city and
                  cityname_city ≤ 500 ▷ condition (b) for unfit city
                then
                    remove city from selected_city_list
                end
            end
        end
    end
end
return selected_city_list
```

5.2.7. Ordering the Selected Cities

Starting with each of the finally selected cities as source, we greedily choose the nearest next cities from the list to find different city orders. For each case, we keep adding the geographical distances to be covered while moving to the next cities, finding the total distances covered. Finally, we choose the order for the selected cities based on the shortest total distance to be covered. Refer to algorithm *selected-city-ordering* for the corresponding pseudo-code.

5.3. Summary

In this chapter, we described the pseudo codes for different algorithms for finding the cities to be recommended to a user.

We presented algorithms that calculate the number of days different tourists should be recommended to stay at different cities. This was based on the earlier trips by other

Algorithm 7: selected-city-ordering

Input: *distance_matrix*,
selected city list *(selected_city_list)*

Output: The ordered list of the selected cities and its corresponding other parameters

```
if there exists at least one item in the selected_city_list then
    set distance_covered_list to empty list ▷ this is to store the total distances to be
      covered, starting the trip with each city in selected_city_list
    set city_orders_list to empty list ▷ list to store the orders of cities covered one
      after the other, starting from each source, going to the nearest one next
    foreach city in selected_city_list ▷ loop to consider each city as source
     do
        set temp_city_list to empty list ▷ list for storing ordered cities each time
        last_city = cityname_city ▷ last_city is a variable to track the last city added
          in the current order and is same as the source city initially
        add last_city to temp_city_list
        total_distance ← 0 ▷ initialising total distance to cover for current order
        for count ← 1 to (number of items in selected_city_list − 1) ▷ counter to add
          rest of the selected cities in the current order, after the source is added
         do
            foreach outer_item in distance_matrix do
                if cityname_outer_item = cityname_city then
                    find first inner item city within outer_item that is not yet added to
                      temp_city_list ▷ first city within inner list is the nearest one
                    add this city to temp_city_list ▷ nearest city is added as the next
                      city in the current order
                    total_distance ← total_distance + distance between last_city and
                      this city ▷ update total distance to be covered
                    set last_city to this city ▷ last city added to the current order
                end
            end
        end
        add temp_city_list to city_orders_list ▷ order with current source is added
        add total_distance to distance_covered_list ▷ total distance for current order
    end
    find minimum from distance_covered_list as minimum distance to be covered
    find the corresponding item from city_orders_list as the final order of cities
    rearrange other parameters of a city according to this order of cities
    overwrite selected_city_list by cities and other parameters in the chosen order
end
return selected_city_list
```

tourists of similar type. If no visit to a city is found by any tourist type, the trips by other tourist types were also considered for finding the duration of stay in that city.

The other algorithms introduced assume to have all the required inputs provided by a user while finding a trip. They find the cities with the duration of stay calculated for the user's type of travelling. Next, they filter the cities based on destination region, and score them based on the user preferences for different city-based features. After that, they select some of the cities considering the constraint of maximum travel duration provided by the user. Some of the cities, that are far away from others in the list and have only a single duration of stay for the current user, are removed from the list. After the removal, if the constraints permit, more cities are considered to be added to form the final list of selected cities.

The algorithm to order the selected list of cities initially finds different orders starting from each city in the list as source and moving to the nearest one next. It calculates the total distances to be covered for visiting the cities in the different orders, and picks up the specific order with the shortest distance to be covered.

We described a composite algorithm that utilises all the other ones discussed for finding the list of cities to be recommended to a particular user based on her preferences. The final recommendation consists of an ordered list of cities along with other parameters including duration of stay at each of them.

6. TripRec — an Application for Recommending Composite City Trips

TripRec[1] is a data-driven prototype application to recommend composite city trips to different types of travellers using the content-based recommendation strategy. The user inputs her preferences to the application and the system recommends her a trip to one or multiple cities. In this chapter, we discuss the development of this application — the front-end and the back-end, as well as the technical requirements for developing the same.

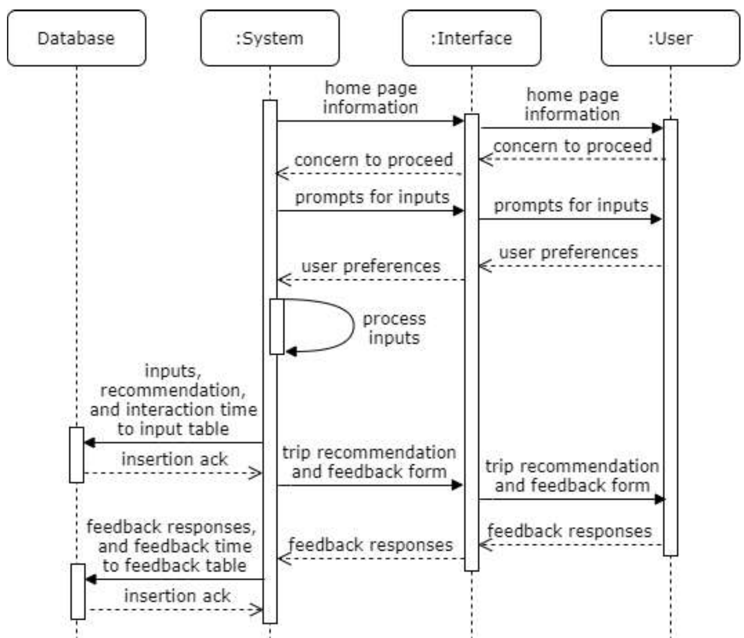


Figure 6.1.: Interaction between user and system

[1]http://triprec.cm.in.tum.de/

6.1. Development of User Interface

A user interface (UI) facilitates the interaction between a user and the system. The system and a user do the following functions using the UI for TripRec:

(a) The system shows basic information about the application to the user.

(b) The system prompts a user to provide different inputs for eliciting her preferences.

(c) The user provides her preferences to the system.

(d) The system shows trip recommendations for the user based on the inputs.

(e) The system requests the user to provide a feedback.

(f) The user submits feedback to the system about her experiences using the application.

Figure 6.1 shows the interaction flows between a user and the system through the UI. Now, we discuss the inputs, the recommendation and the feedback in detail.

6.1.1. User Inputs

The basic way a system interacts with a user is by showing some information, asking her to respond. The user responds by providing inputs, allowing the system to proceed accordingly. We now discuss information by the system and inputs by the user.

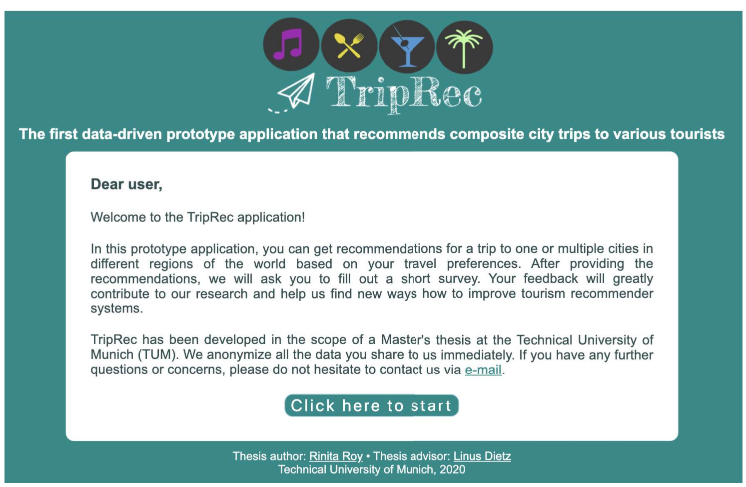


Figure 6.2.: Landing page of *TripRec*

The landing page of TripRec

The landing page (Figure 6.2) of *TripRec* application introduces it with a few words and asks the user to start the recommendation process. Clicking on the start button leads to begin the step-by-step process in which the application prompts the user to provide different inputs, which are processed to generate recommendations for the user.

Home region of the user

After displaying our customised regions in a world map, *TripRec* asks the user to input the region of her home location. As shown in Figure 6.3, we show the 10 regions using 10 different colors on the map. The user is then asked to identify the region of her home location in the map and select the corresponding button with the same color.

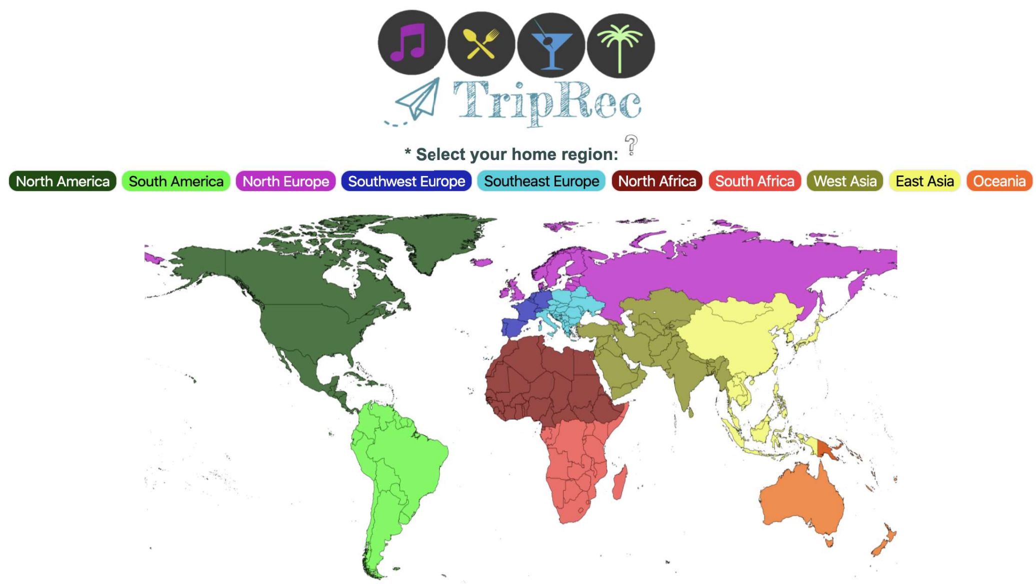


Figure 6.3.: Home region selection

Traveller type of the user

Based on the home region of the user, the application shows the different traveller types we obtained earlier for all those travellers having their home at the corresponding region. It also provides a textual description for each of the types shown, so as to help users choose the traveller role that matches the best for them. We provided short

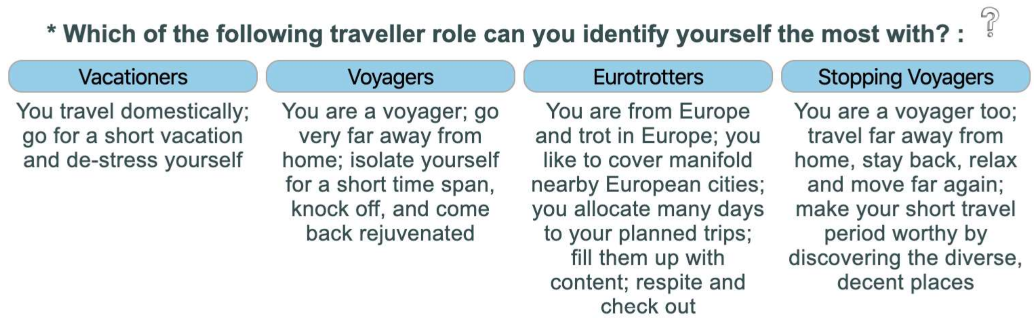


Figure 6.4.: Traveller type self-identification

descriptions for 10 types of travellers across different home regions in Section 4.1, describing their characteristic features. However, in this case, the textual descriptions are more informal and non-technical, as enlisted below:

(i) **Vacationers** — *You travel domestically; go for a short vacation and de-stress yourself.*

(ii) **Explorers** — *You like exploration trips; go touring to somewhere distant, for a relaxing short time period.*

(iii) **Voyagers** — *You are a voyager; go very far away from home; isolate yourself for a short time span, knock off, and come back rejuvenated.*

(iv) **Globetrotters** — *You try to cover as many cities as possible in a trip; travel far from home; you allocate many days to your planned trips; delighten the period with a thrilling roller coaster ride to manifold cities.*

(v) **Eurotrotters** — *You are from Europe and trot in Europe; you like to cover manifold nearby European cities; you allocate many days to your planned trips; fill them up with content; respite and check out.*

(vi) **Stopping Voyagers** — *You are a voyager too; travel far away from home, stay back, relax and move far again; make your short travel period worthy by discovering the diverse, decent places.*

(vii) **Long Stopping Voyagers** — *You are a voyager too; take your time out, move very far from home, away from all stress; you have wide travel span but you prefer to visit few cities; tranquilize at a place, move far again and discover more.*

(viii) **Hop-trotters** — *You are a voyager too, travel far from home; cover many cities as well, however, within a short time span.*

(ix) **Long Hop-trotters** — *You try to cover as many cities as possible; travel very far away from home; you enjoy all the places and get involved into the culture staying everywhere back for long time.*

(x) **Long Explorers** — *You are an explorer; travel to few cities, however, completely relish the places you go to; utilize your long travel period to mix up with the local cultures and unwind yourself.*

Figure 6.4 shows a screenshot for how the traveller types and their corresponding descriptions are shown using the UI. This is additionally displayed to the user after the home region *Southwest Europe* is selected by the user.

Destination region for the trip

Following the user's selection of a traveller type, she is asked to provide her destination region for the desired trip. This is accompanied by displaying the flags of the countries in our database corresponding to each region, as shown in Figure 6.5. They are added below the traveller types displayed earlier. The inputs selected by the user in the previous steps also keep on getting appended. As an example, the traveller type *Eurotrotters* selected by the user can also be seen in the screenshot.

Figure 6.5.: Destination region selection

Maximum travel duration constraint

The application aggregates all trips in our database by travellers that match the current user, in terms of her selected home region and traveller type. From the distribution of travel duration for these trips, the absolute values of the $50th$ and the $85th$ percentiles are evaluated to be, say, *med_dur* and *max_dur* days, respectively. Setting the maximum travel duration to the median value, *med_dur* days as a default, user is given an option to change it to a value between 7 and *max_dur* days. The median duration is set to be the default duration initially as it is the most frequent value in the distribution. The maximum is set to the $85th$ percentile to avoid the infrequent high outliers. The slider shown to the user for changing the maximum duration value can be adjusted using the handle within or with the $-$ or $+$ buttons at both sides (Figure 6.6).

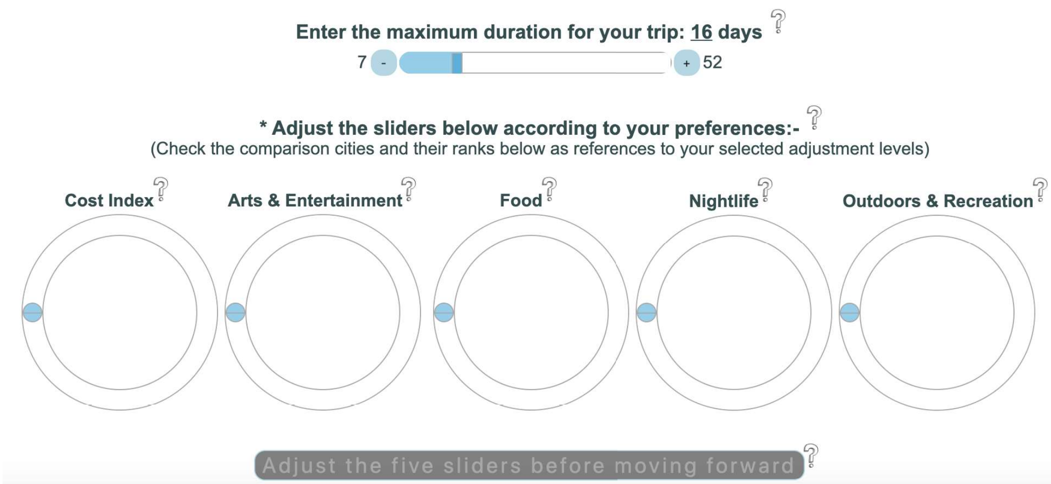


Figure 6.6.: Selection of maximum travel duration and initialisation of round sliders

User preferences for city-based features

In the same step when the maximum travel duration slider is shown, the UI simultaneously displays five round sliders asking the user to provide preferences for each of the city-based features, for the cities she wants to visit. The 138 cities in our database are sorted on the basis of the values for individual features. Each time the user selects a level for a feature, she is shown the city closest in terms of the corresponding feature value and its rank amongst the 138 cities for aiding the comparison.

A user is asked to adjust the round sliders, which can have values valid between 0.00 and 1.00 with a precision of 0.01. Four different colors are provided to the slider tracks based on the range within which the value is selected, emphasising different levels. The ranges are $0.00 - 0.25$ (slider range I), $0.26 - 0.50$ (slider range II), $0.51 - 0.75$ (slider range III), and $0.76 - 1.00$ (slider range IV). The values for the round sliders are initially set to -1, and the user must adjust all of them before moving forward, as shown in Figure 6.6. We do not show any other default values because users might be negligent to change them if some values are already selected, making the results biased.

In Figure 6.7, selected values for *cost index*, *arts & entertainment*, *food*, *nightlife*, and *outdoors & recreation* are in slider ranges III, II, IV, I, and III, respectively, displaying four different colors of the tracks. After all the five sliders are adjusted, the submit button, which was earlier disabled, becomes enabled, clicking on which shows the recommendation as results. Providing all the inputs to the system, the user can move forward and request a trip recommendation.

Figure 6.7.: City-based features' preference level selection

6.1.2. Trip Recommendation

Receiving the recommendation request, the system starts processing the inputs further. It executes algorithm *composite-city-recommendation* discussed in the previous chapter. The algorithm finds the ordered list of cities along with other parameters to be recommended to the current user. The user interface displays the cities, its corresponding countries, and the duration of stay at each city in the order returned by the algorithm. This is also accompanied by presenting the order of visits to the recommended cities using Google Maps[2]. We get the recommendation as shown in Figure 6.8 for the exemplary inputs in the earlier screenshots.

If there is no result found for the current user based on the input data provided, we show a message to the user asking her to adjust inputs and try again (Figure 6.9). Moreover, the user can also change the inputs on her own and request for different recommendations multiple times.

[2]https://www.google.com/maps

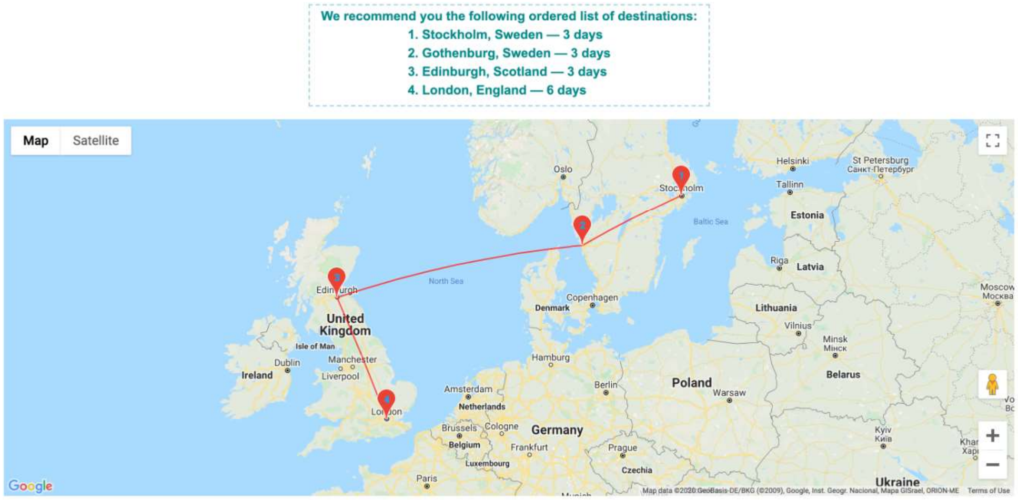


Figure 6.8.: Exemplary recommendation

No results found! Please increase your maximum travel duration or change your destination region, and try again.

Figure 6.9.: Message when no results are found

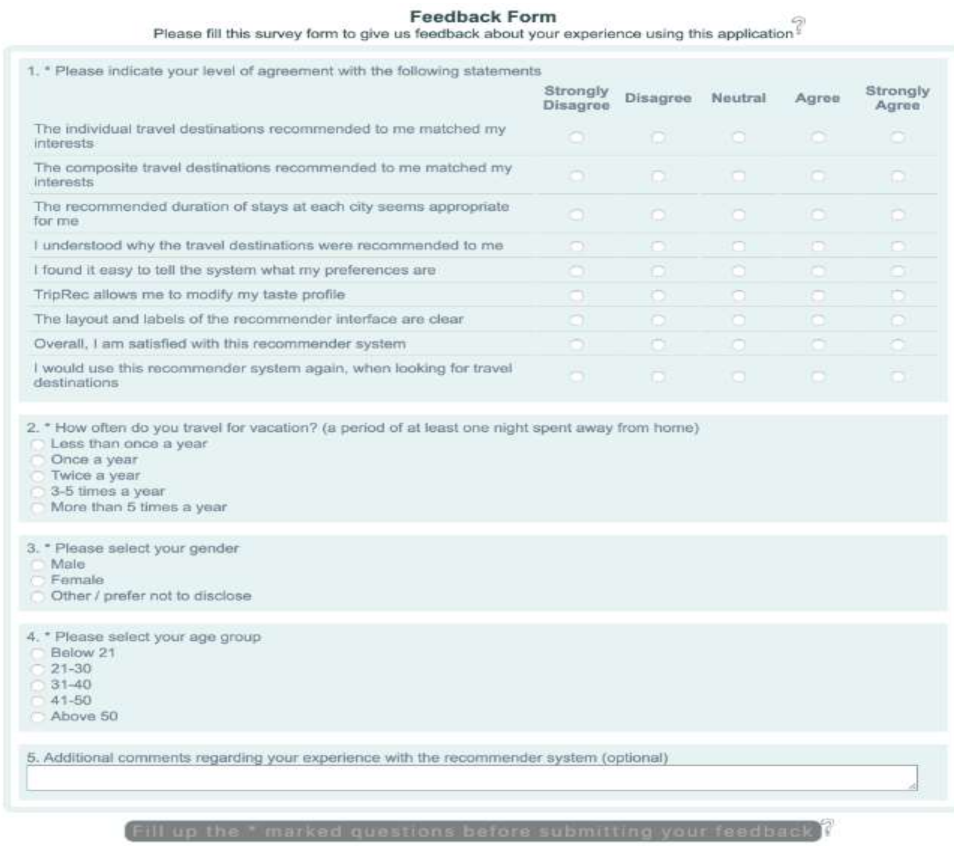
6.1.3. Feedback Form

Below the recommendation list and the map, the interface also shows a feedback form (Figure 6.10) to users, asking them to share their experiences using the *TripRec* application.

Before submitting the feedback form, there are total 12 questions for the user mandatory to be filled and one optional additional comment to be added. In the first nine questions, user needs to specify their level of agreement to nine different statements on a five-point likert scale [50] — *strongly disagree*, *disagree*, *neutral*, *agree*, and *strongly agree*. The responses check the recommender system on the basis of quality of the recommended items, transparency, ease of preference elicitation and revision, interface adequacy and attitudes of the user. Following are the statements under each category:

Quality of Recommended Items

1. The individual travel destinations recommended to me matched my interests (q1a)



Feedback Form

Please fill this survey form to give us feedback about your experience using this application

1. * Please indicate your level of agreement with the following statements

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The individual travel destinations recommended to me matched my interests	○	○	○	○	○
The composite travel destinations recommended to me matched my interests	○	○	○	○	○
The recommended duration of stays at each city seems appropriate for me	○	○	○	○	○
I understood why the travel destinations were recommended to me	○	○	○	○	○
I found it easy to tell the system what my preferences are	○	○	○	○	○
TripRec allows me to modify my taste profile	○	○	○	○	○
The layout and labels of the recommender interface are clear	○	○	○	○	○
Overall, I am satisfied with this recommender system	○	○	○	○	○
I would use this recommender system again, when looking for travel destinations	○	○	○	○	○

2. * How often do you travel for vacation? (a period of at least one night spent away from home)
- Less than once a year
- Once a year
- Twice a year
- 3-5 times a year
- More than 5 times a year

3. * Please select your gender
- Male
- Female
- Other / prefer not to disclose

4. * Please select your age group
- Below 21
- 21-30
- 31-40
- 41-50
- Above 50

5. Additional comments regarding your experience with the recommender system (optional)

Fill up the * marked questions before submitting your feedback

Figure 6.10.: Feedback form

2. The composite travel destinations recommended to me matched my interests (q1b)

3. The recommended duration of stays at each city seems appropriate for me (q1c)

Transparency

4. I understood why the travel destinations were recommended to me (q1d)

Ease of Preference Elicitation and Revision

5. I found it easy to tell the system what my preferences are (q1e)

6. TripRec allows me to modify my taste profile (q1f)

Interface Adequacy

7. The layout and labels of the recommender interface are clear (q1g)

Attitudes

8. Overall, I am satisfied with this recommender system (q1h)
9. I would use this recommender system again, when looking for travel destinations (q1i)

The remaining three questions ask for personal information about the user with one of the given options to select from. Following are the questions and the corresponding options:

10. How often do you travel for vacation? (a period of at least one night spent away from home)
    (a) Less than once a year
    (b) Once a year
    (c) Twice a year
    (d) 3-5 times a year
    (e) More than 5 times a year
11. Please select your gender
    (a) Male
    (b) Female
    (c) Other / prefer not to disclose
12. Please select your age group
    (a) Below 21
    (b) 21-30
    (c) 31-40
    (d) 41-50
    (e) Above 50

The feedback form is shown throughout below the recommendations until it is submitted. During a session, once feedback is provided by a user, a dialogue box is displayed as shown in Figure 6.11. The form disappears from the web-page after clicking on the *OK* or the × (close) button. However, the user can still continue using the application for getting further recommendations.

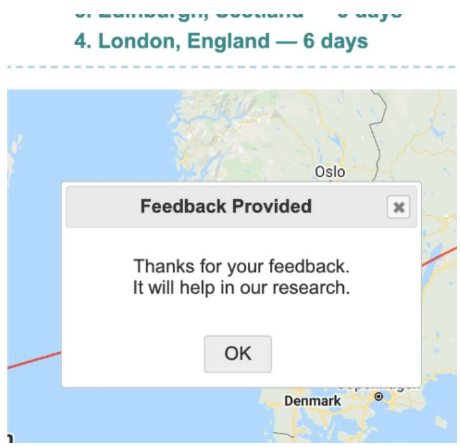


Figure 6.11.: Dialogue shown after feedback is provided

6.1.4. Tooltips for User Assistance

Throughout the interaction between a user and the system, the user is supported with helpful information about the reason for collection of data and how to proceed further in each step. "?" (.gif) icons are displayed throughout the UI to gain user attention. When a user hovers over them, we show the corresponding tooltip texts during different steps as follows:

- Home region selection: *We consider the travellers from the same home region as yours and consult their previous trips before recommending you something similar. Identify your home country from the world map shown below, check the color provided to the region it belongs to, and click on the button with the corresponding color.*
- Traveller type self-identification: *We account for the tours of the previous travellers having same type as selected by you here. Check the descriptions provided below each type and click the one that match the best with you.*
- Destination region selection: *We recommend you the destination cities belonging to the region you choose here. Go over the flags below for identifying the countries we currently have in our database under each region. Click on one of the corresponding buttons below.*

We also display the names of the countries when hovered over corresponding flags using tooltip texts.

- Maximum travel duration selection: *You provide us with an upper limit for your travel duration with this input. We set a default maximum duration of travel for your traveller type. Move the blue handle right or left for increasing or decreasing this value. You can also click on the + or - buttons for adjustments.*

- City-based features' preference level selection: *Your adjustment of the five sliders below help us develop your taste profile with the type of venues you might be more interested in. Slide the blue circular handles below clockwise or anti-clockwise to increase or decrease your individual preference levels for five different parameters of a city. You can compare your selected level with a city and its corresponding rank for each parameter.*

  Tooltip texts shown above the round sliders corresponding to each of the city-based features are as follows:

  * Cost index — *Higher the rank for Cost Index, more costly is the city*
  * Arts & entertainment — *Higher the rank for Arts Entertainment, higher is the frequency of venues like art galleries, circuses, comedy clubs, salsa clubs, concert halls, science museums, multiplexes found in the city*
  * Food — *Higher the rank for Food, more is the availability of restaurants and cafés in the city*
  * Nightlife — *Higher the rank for Nightlife, higher is the presence of pubs or nightclubs in the city*
  * Outdoors & recreation — *Higher the rank for Outdoors Recreation, more is the prevalence of venues like forests, national parks, ski or skydiving areas, beaches, gardens, and scenic lookouts in the city*

- Recommendation and feedback:

  * Near the submit button before showing results — *You can still modify your preferences above to get different recommendations. Please also scroll down for a feedback form.*
  * Near the submit button, after showing results and before feedback is submitted — *You can still modify your preferences above to get different recommendations. Please also scroll down for a feedback form.*
  * Near the submit button, after showing results and after feedback is submitted — *You can still modify your preferences above to get different recommendations.*

* Near the submit feedback button when it is disabled — *Please check that you have selected an option for each of the * marked questions.*

* Near the submit feedback button when it is enabled — *Please submit your valuable feedback.*

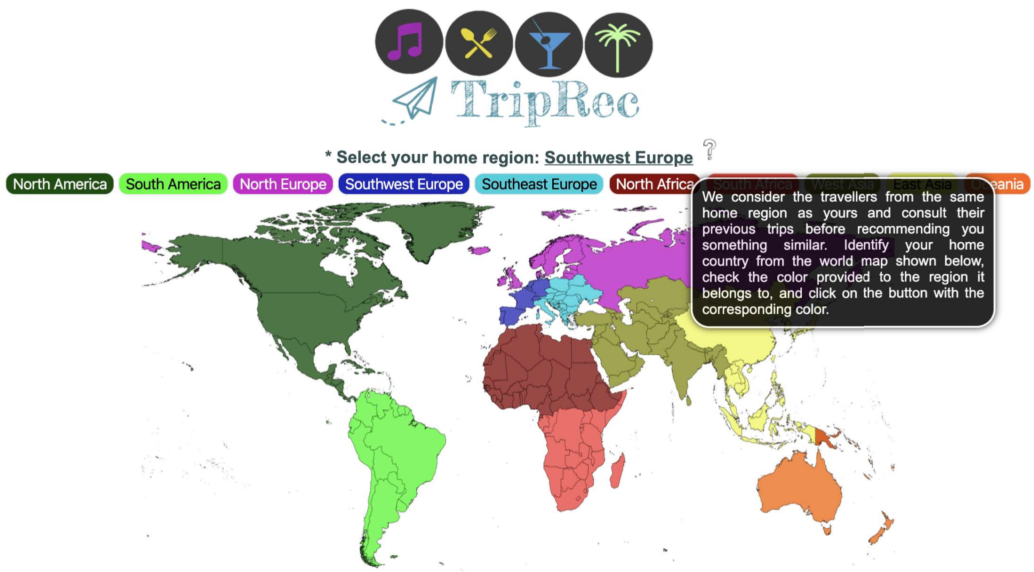


Figure 6.12.: Tooltips to help users

Figure 6.12 shows an exemplary tooltip from our application during the home region selection step.

6.2. Development of Back-end

We use flat files — comma-seperated values (*.csv) files and Microsoft Excel (*.xlsx) files — for storing all processed data like trips, cities, and traveller types. During the recommendation process, these files are fetched run-time into the main memory only once and are used throughout for further processing. However, we create a relational database for keeping on storing user data while people use the application. The database has two tables:

(i) **input table** — for storing all the user inputs, the corresponding recommendation provided to the users, and the interaction time.

Interaction time is the time spent by the user interacting with the application before getting a recommendation. The timer to measure it runs from showing the prompt for providing the first input (home region), till the user requests for the recommendation.

Each time a user clicks on the submit button to get a recommendation, there is a new entry in the input table. The session id stored for each row inserted into the table remains the same for one user during a session. This might be later used up to check if the user requests for multiple recommendations during a session.

(ii) **feedback table** — this table contains the feedback from the users. It stores the answers provided by the users against each question in the feedback form, sharing their experiences, and the feedback time.

Feedback time is the time elapsed since showing a recommendation to the user until she submits a feedback responding to the provided questions.

There can be one or no row inserted into the feedback table for a user in one session, based on if she submits a feedback or not, respectively.

6.3. Technical Specifications

In this section, we specify the tools and technologies (Figure 6.13) used for the development of our application. We list the server side and the client side technologies dealing with the back-end and the front-end concerns, respectively. We also mention the full-stack framework used for supporting both, followed by citing other tools.

- **Server Side:**
  1. Programming language
     * Python — Python 3.8[3] is used for computation, clustering, analysis, and plotting of data.
  2. Structured query language
     * PostgreSQL — *PostgreSQL 12.2*[4] is used as the database management system to store the user input and feedback data to be evaluated later.
- **Client Side:**

[3]https://docs.python.org/3.8/
[4]https://www.postgresql.org/docs/12/index.html

3. Markup language
   * Hypertext Markup Language (HTML) — *HTML5*[5] is used to design the web-pages or interfaces.
4. Style sheet language
   * Cascading Style Sheets (CSS) — *CSS3*[6] is used to make the web-pages more presentable.
5. Scripting language and libraries
   * JavaScript (JS)[7] is used to make the web pages dynamic and facilitate interaction between the user and the web pages.
   * jQuery[8], a JavaScript library, is also used to do some common interface interaction tasks in simpler ways, and provide specific animations to the web pages.
   * jQuery UI[9], a library built on top of jQuery, is used for widgets like dialog, and for providing themes to the tooltips intended to assist users.

- **Application Framework and other tools:**

6. Micro-framework Flask and its dependencies
   * Flask — *Flask 1.1.1*[10], a web micro-framework for Python, is used to build our web application, manage the http requests, mapping them to Python functions and rendering respective pages.
   * Jinja — we use this flask dependency as the template language that renders the web pages of our application.
7. Integrated development environments (IDEs)
   * Visual Studio Code 1.44.2[11]
   * pgAdmin 4[12]

[5]https://html.spec.whatwg.org/multipage/
[6]https://www.w3.org/TR/css-writing-modes-3/
[7]https://developer.mozilla.org/en-US/docs/Archive/Web/JavaScript/New_in_JavaScript
[8]https://jquery.com/
[9]https://jqueryui.com/
[10]https://flask.palletsprojects.com/en/1.1.x/
[11]https://code.visualstudio.com/docs
[12]https://www.pgadmin.org/docs/pgadmin4/4.11/index.html

Figure 6.13.: Technologies used in the development of *TripRec*

6.4. Summary

In this chapter, we presented *TripRec*, the prototype application we developed to test the functionalities of the first version of our composite city trip recommender system. We discussed the development of its user interface, the back-end and also specified the technical requirements.

The user interface consists of a landing page and the options to provide the inputs to the system as prompted by it step by step. This is followed by the recommendation delivered to the user after she requests the system to do so. A list of cities and their corresponding countries are displayed to the user in a recommended order of visits, along with the duration of stay in each city. The user is also asked to fill up and submit a feedback form shown to her sharing her experiences using our application. She is guided with helpful information in every step while she uses the application.

We also talked about the database used to store the user inputs and the feedback information provided by them, followed by the technical specifications. We analyse the data provided by the users while using *TripRec* in the next chapter.

7. User-centric Evaluation of TripRec

We conducted a user study for *TripRec* to examine the behaviours of its users, their opinions about the system, and some characteristics of the services provided to them. Within a span of two weeks, we accumulated 217 recommendation requests and 75 feedback from users for evaluation.

One user can request for recommendations multiple times. The application received 217 recommendation requests. However, they were only 113 unique users using the application. Table 7.1 and Table 7.2 show the statistics of the requests from these users, and their distribution, respectively.

Table 7.1.: Request statistics

User statistics	No. of requests
count	113
mean	1.92
std. deviation	2.88
min	1
25%	1
median	1
75%	2
max	26

Table 7.2.: Request distribution

No. of requests	No. of users
1	78
2	17
3	7
4	5
5	2
6	2
16	1
26	1

We can notice that most of the users requested for only one recommendation each ($median = 1$). Few users changed their preferences and requested for one more recommendation ($75\% = 2$). However, the number of requests from even fewer of them were more than two, with the maximum number of requests being 26. Thus, to avoid duplicates, we consider only the last request from each user for further analysis in the upcoming sections.

7.1. Different Users and their Behaviours

We consider recommendation requests by various users to learn about their behaviours, in terms of their traveller types, travel destinations, and preferences for the city-based features.

7.1.1. Flows from Home Region to Destination Regions

The 113 user requests are segregated with respect to home regions and destination regions in Table 7.3 and Table 7.4, respectively. The world regions considered to be home or destination regions, as discussed earlier, are North America (NAm), South America (SAm), North Europe (NE), Southwest Europe (SWE), Southeast Europe (SEE), North Africa (NAf), South Africa (SAf), West Asia (WAs), East Asia (EAs) and Oceania (O).

Table 7.3.: Home region-wise user requests

Home	Requests
WAs	37
SWE	27
SEE	20
EAs	8
NE	5
NAm	5
SAm	4
NAf	4
SAf	3
O	0

Table 7.4.: Destination region-wise user requests

Destination	Requests
SWE	28
WAs	18
SEE	16
NE	15
NAm	14
EAs	10
SAm	6
NAf	3
O	2
SAf	1

The application received the maximum number of requests by people from *West Asia*, followed by those from *Southwest Europe* and *Southeast Europe*, whereas there were no user from *Oceania* using our system. Other than *West Asia*, users wanted to go to the European regions, viz., *Southwest Europe*, *Southeast Europe*, and *North Europe* the most. To understand who wants to go where, we observe the transition of users from one region to the another. We visualise the flow proportions from home regions to different destination regions using parallel sets as shown Figure 7.1.

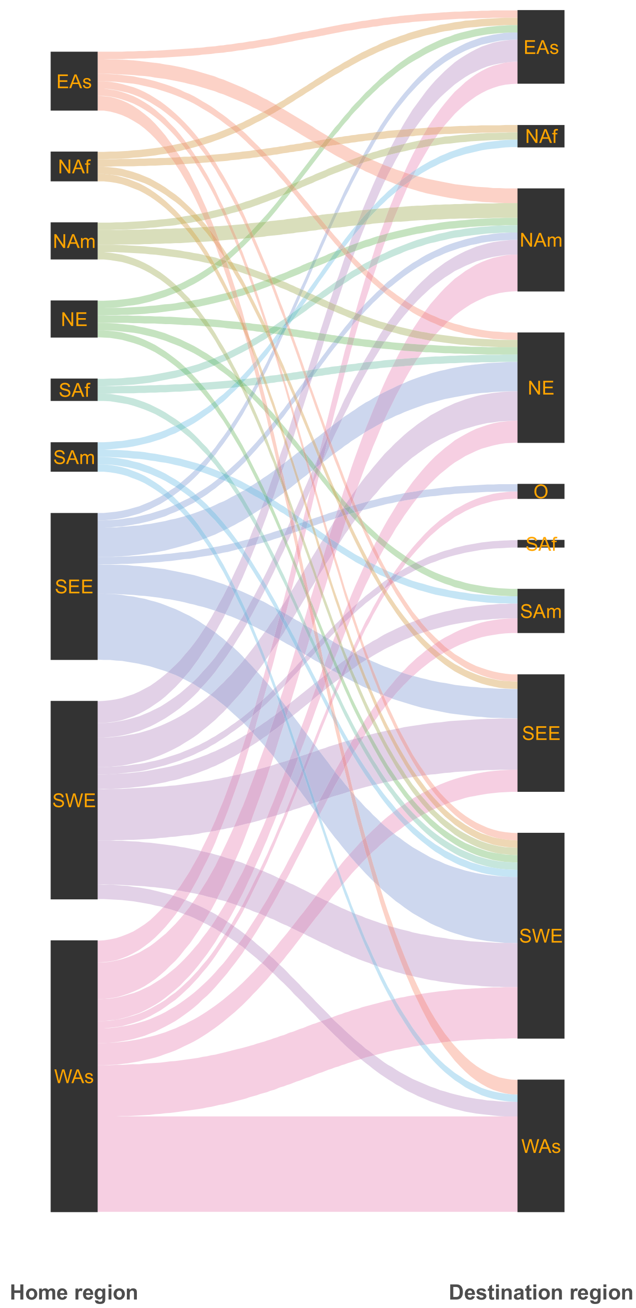


Figure 7.1.: Flow of user requests from home regions to destination regions

Following are some observations from the diagram:

(a) Other than being the destination with the maximum number of upcoming tourists, *Southwest Europe* is also the most popular travel destination, based on this user study. We can say this as it is the only destination region, receiving recommendation requests from users of all other home regions.

(b) *North America* seems to be the second popular destination region amongst tourists from different regions.

(c) Users from *Southwest Europe* wanted to visit *Southeast Europe* the most, and vice versa, followed by destinations within their own regions and then *North Europe*.

(d) Maximum users were from the home regions *West Asia* and *Southwest Europe*. The regions also had the most diverse users in terms of different destination regions they prefer to go to.

(e) *North Africa* is the only common region, that received no request from either of the regions with most diverse users, viz., *WAs* and *SWE*.

7.1.2. Popularity of the Traveller Types

We enlisted the prevalence of the different traveller types across all home regions earlier in Table 4.4. Now, we detect the popularity of these traveller types from our user data.

Table 7.5.: User-data-wise frequency of traveller types in the different regions

Traveller types	Home regions	Frequency (User data/dataset)
Vacationers	*NAm*, *SAm*, *NAf*, *SWE*, *SEE*, *EAs*	6/8
Explorers	*NAf*, *SAf*, *NE*, *WAs*	4/5
Voyagers	*NAm*, *SAm*, *SWE*, *SEE*, *WAs*, *EAs*	6/10
Stopping Voyagers	*SAm*, *NAf*, *NE*, *SWE*, *EAs*	5/8
Hop-trotters	*SAf*	1/1
Eurotrotters	*NE*, *SWE*, *SEE*	3/3
Globetrotters	*NAm*, *NAf*, *WAs*	3/7
Long Stopping Voyagers	-	0/1
Long Hop-trotters	-	0/1
Long Explorers	*NAf*	1/3

For each of the 10 traveller types, Table 7.5 enlists the home regions, users from which chose that traveller type to follow, for their requested trips. The frequency column displays the count of those home regions for each traveller type, against the number of regions, in which we identified the corresponding traveller type in our dataset. As an example, users from six home regions chose to be *vacationers* for their requested trip. However, as shown in Table 4.4, we identified travellers of type *vacationers* in eight home regions in our original dataset.

7.1.3. User Preferences for City-based Features

A user has to provide her preferences to the system to get a trip recommendation. In Figure 7.2, we radially arrange nine axes representing the home regions of the users, and plot the home region-wise median preference values for the five city-based features on each axis. This spider plot helps us compare (a) the user preferences for the different city-based features within one home region, and (b) the preferences for each city-based feature by the users from different home regions. We choose the mean value of the features by the users from each home region for plotting.

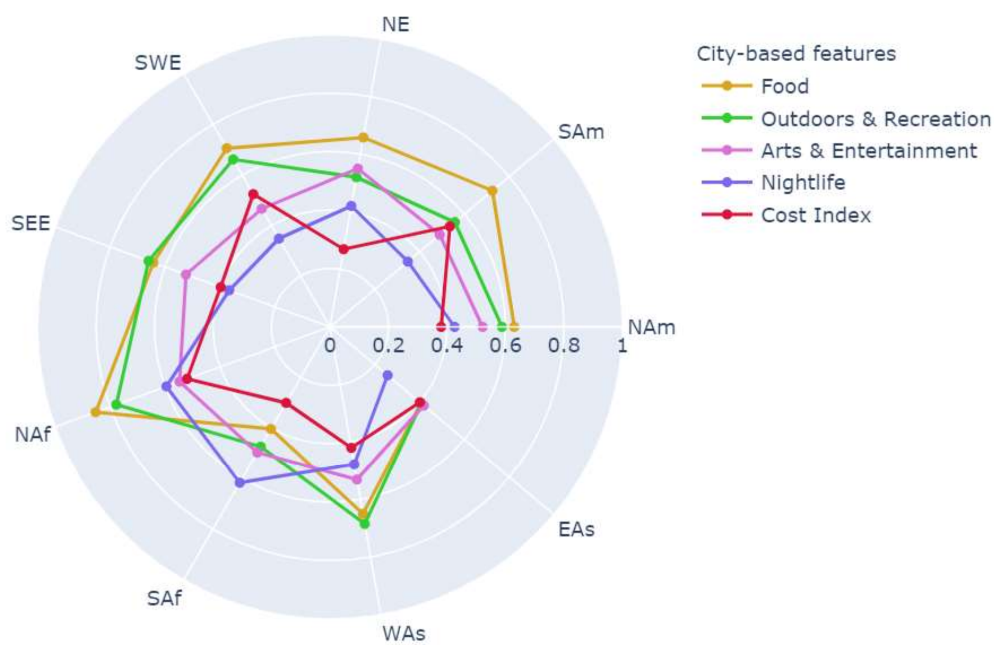


Figure 7.2.: Home-region-wise user preferences for city-based features

Following are some of the observations of our user data from Figure 7.2:

(a) The user preferences for *food* tend to be typically higher, except for users from *South Africa*.

(b) The user preferences for *outdoors & recreation* is slightly higher than preferences for *food* for users from *West Asia* and *Southeast Europe*. For others also, it is typically between medium and high, except for users from *South Africa* and *East Asia*.

(c) Users from all regions tend to visit cities with medium *cost index* and *arts & entertainment* values.

(d) The preference for *nightlife* is comparatively more in case of users from *South Africa* and *North Africa*, whereas for people of *East Asia*, this preference value is the smallest.

7.1.4. Modification of Default Travel Duration

In the UI for our application, we show a default value for the maximum travel duration as discussed in Section 6.1. With our user data, we now check how many users actually modified these values before requesting recommendations. We subtract the default travel duration shown to the user from her maximum travel duration input. The result can be a positive integer, a negative integer or zero, depending on whether the user increased her preferred maximum travel duration more than the default value, decreased it from default, or did not change it at all, respectively. Figure 7.3 shows a histogram for these modification values.

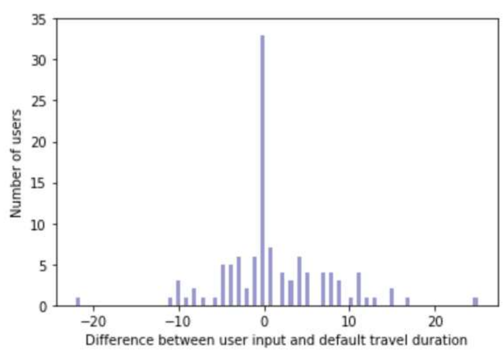


Figure 7.3.: User modification of default travel duration

We can notice that most of the users opted not to modify the default travel duration shown to them. This leads us to think if we should make the modification by a user mandatory before she is able to move forward, to avoid any bias in the recommendation results.

7.2. Analysing Recommendations Based on User Data

In this section, we analyse the recommendations provided to users by *TripRec* based on their preferences. This is done in terms of the cities recommended to the users most frequently and the typical duration of stay at each city for different users. In both the cases, we take all the 217 recommendation results into consideration, as these cases examine the recommendations by the system, not the user behaviours.

7.2.1. Frequently Recommended Cities Per Destination Region

We determine which cities are recommended the most by calculating the recommendation ratio (RR) of the cities within each destination region. Recommendation ratio of a city is the number of times it is recommended divided by the total number of recommendations within the corresponding destination region. Refer to Appendix D for the destination region-wise RRs of the cities in our database, calculated using the collected user data. Considering the cities within the same destination region, those with higher RR has been recommended more often than those having lower values.

As can be seen in Table D.1, from *Southwest Europe*, *Leipzig* was recommended the most to the users, whereas *Bilbao* was the least recommended city. We can also see from Table D.2 that both the cities *Nairobi* and *Kampala* were recommended every time users asked to visit *South Africa*. This lack of diversity in the recommendation is because these two are the only cities under *SAf* in our database. Thus, we can see more variation of cities in the recommendation results when there are more cities under a region.

7.2.2. Proportionate Recommended Duration Distribution across Cities

Different duration of stays are recommended to travellers of different types, that depend on their home regions. Using our user study data, we find out, on an average, what proportion of the total travel duration is the recommended duration of stay at each city, for travellers from different regions. The recommended average duration of stay at a city divided by the average duration of a trip is called as the mean proportionate duration of stay (MPDS) at a city. We use stacked bar plots for showing the results in Figure 7.4 and Figure 7.5. The usage of two figures aids for clear visualisation.

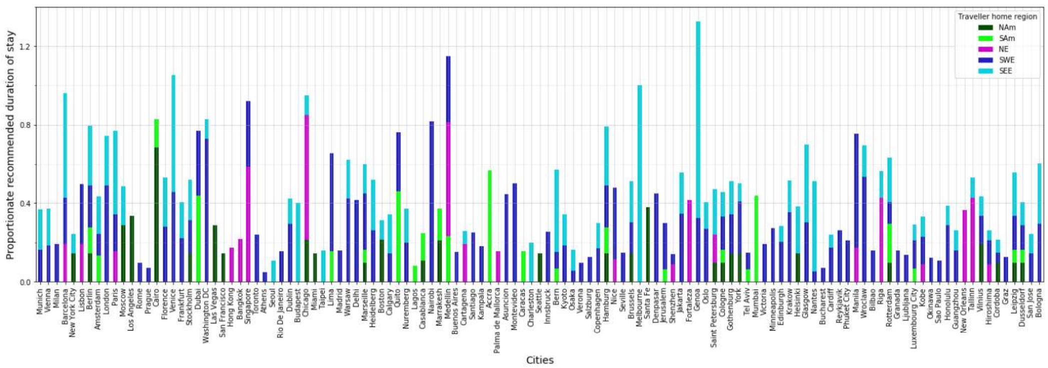


Figure 7.4.: Proportionate recommended duration of stay for travellers across cities (I)

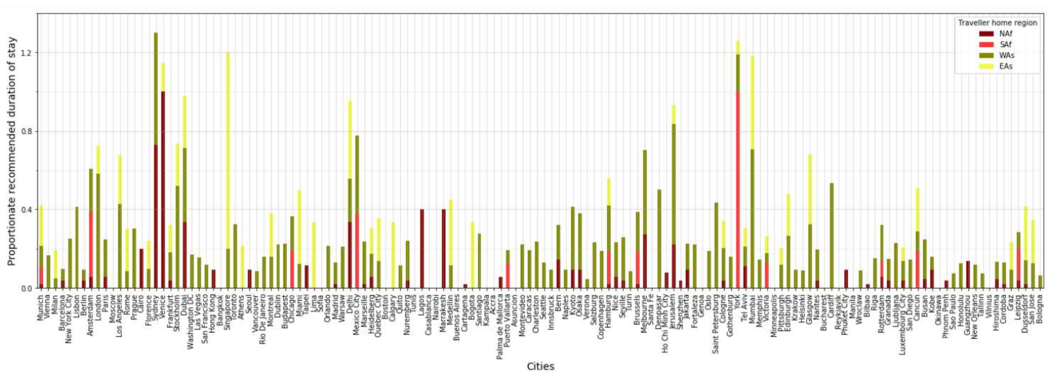


Figure 7.5.: Proportionate recommended duration of stay for travellers across cities (II)

Following are some exemplary observations:

(a) The MPDSs at *Munich* recommended to travellers from *SWE*, *SEE*, and *EAs* were quite balanced in terms of the whole trip duration. The values were comparatively smaller for those from *WAs* and *SAf*, and very small for people from *NAf*. *Munich* was not recommended to users from *NAm*, *SAm*, and *NE* corresponding to their elicited preferences during the user study, explaining the absence of the representing bars for the city.

(b) People from *NAf* were typically recommended to stay at *Venice* during their whole trip at *SEE*. Similar is the case for users from *SAf*, who were recommended to stay in the city *York* for their whole trip when they wanted to go to *NE* for a particular duration. In both these cases, the MPDS at the cities for the corresponding travellers is 1.0.

The addition of more cities in the database with diverse duration of stays can balance the MPDS at different cities.

7.3. User Preferences & Experiences Based on their Feedback

23 female and 52 male participants constituted the 75 users who provided their feedback after using our system. We inspect them both qualitatively and quantitatively.

7.3.1. Quantitative Feedback Analysis

Here, we examine user preferences and experiences based on their feedback using summarising tables and plots. Firstly, we discover the preferable trip duration of these users w.r.t their age groups (Table 7.6) and vacation frequency (Table 7.7).

Table 7.6.: Age-group-wise mean travel duration

Age group (years)	Travel dur (days)	Std
Below 21	20.0	0.0
21–30	16.3	8.9
31–40	11.8	3.7
41–50	16.8	3.9

Table 7.7.: Vacation-frequency-wise mean travel duration

Vacation frequency (per year)	Travel dur (days)	Std
Less than once	16.3	5.6
Once	15.4	7.3
Twice	13.8	6.2
3-5 times	16.6	6.8
More than 5 times	24.8	23.1

Next, we find out how satisfied the users were with our system based on their age groups (Table 7.8). Most of the users belonged to the age range between 21 and 30 years. We also noticed that the users aged over 40 years tended to get more satisfied with the system, probably because of lower expectations.

Now, we compare the users' agreement levels to the various feedback questions ($q1a$–$q1i$) mentioned in subsection *Feedback Form* by plotting a stacked bar chart (Figure 7.6). Users seem to have mostly agreed to the provided questions in favour of *TripRec*. However, looking at the responses closely, we note the following points:

Table 7.8.: Overall user satisfaction w.r.t. age group

Age group (years)	Satisfaction level	Number of users	Percentage of users (%)
Below 21	Neutral	1	100
21–30	Very unsatisfied	3	4.9
21–30	Unsatisfied	5	8.2
21–30	Neutral	9	14.8
21–30	Satisfied	36	59.0
21–30	Very satisfied	8	13.1
31–40	Unsatisfied	1	12.5
31–40	Neutral	3	37.5
31–40	Satisfied	2	25.0
31–40	Very satisfied	2	25.0
41–50	Satisfied	3	60.0
41–50	Very satisfied	2	40.0

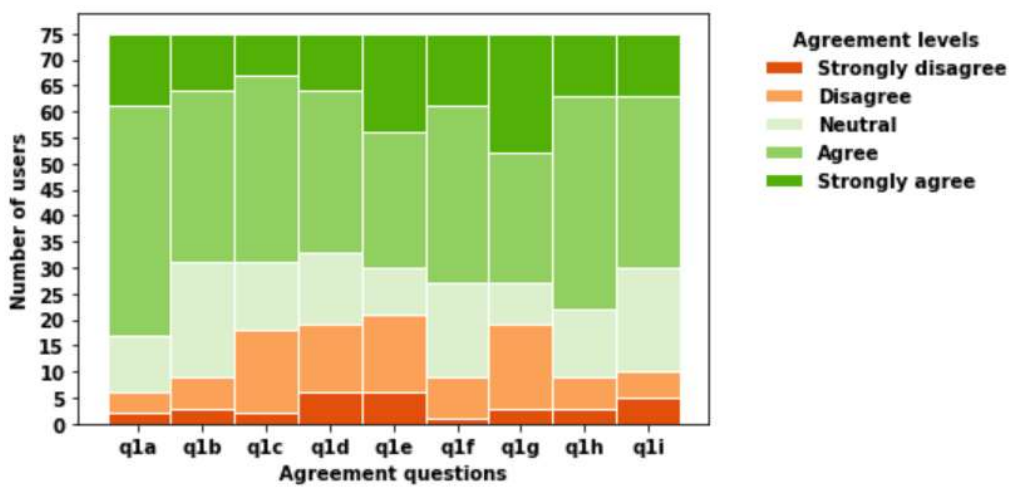


Figure 7.6.: Users’ agreement levels for feedback questions

(a) Most of the users were satisfied with the individual recommendations ($q1a$). However, the number of users satisfied with the composite recommendations ($q1b$) was comparatively lesser.

(b) Users dissatisfied with the recommended duration of stays ($q1c$) were comparatively more than those dissatisfied with the recommended cities ($q1a$, $q1b$).

(c) More people found it easier overall to modify their taste profiles ($q1f$) and get different results, than those agreeing to easily tell the system about their specific preferences ($q1e$).

(d) Maximum number of users have strongly agreed to having clear layouts and labels for the interface ($q1g$), followed by those strongly agreeing to being able to specify their preferences to the system ($q1e$).

(e) A lot of users have agreed to have overall liked *TripRec* ($q1h$). However, comparatively lesser people agreed to use the system again ($q1i$). People who were neutral about whether they liked the system overall ($q1h$) were lesser than those neutral about using it again for planning their trips ($q1i$).

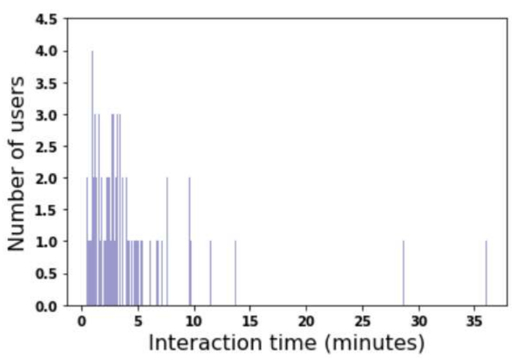


Figure 7.7.: Interaction time histogram

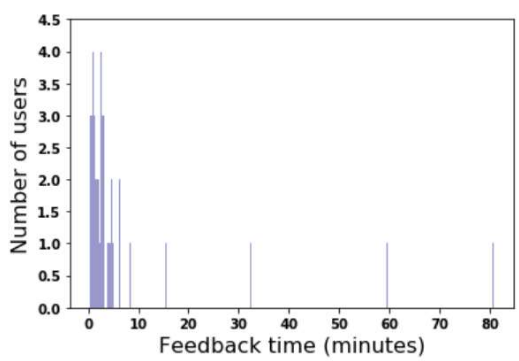


Figure 7.8.: Feedback time histogram

Finally, we determine how long the users interact with the system in terms of interaction time and feedback time. We consider only the final interaction time by each user. One user might have stayed inactive for too long before submitting her final request to get a trip recommendation. We eliminate that outlier record with interaction time of about *10 hours*. Then, we plot the histogram for interaction time as shown in Figure 7.7, the mean interaction time being *4 minutes and 40 seconds* with *6 minutes* standard deviation. Similarly, Figure 7.8 shows the histogram for feedback time, the average being *5 minutes and 45 seconds* with standard deviation *12 minutes and 25 seconds*.

7.3.2. Qualitative Feedback Analysis

Some users gave additional comments expressing their concerns or providing suggestions to improve our system. Few notable ones are summarised below with our remarks on top of that:

(a) Exclude already visited countries from the recommendation list of a user — this was out of scope, but can be considered later.

(b) Round sliders were difficult to handle in some mobile devices — our prototype system was not yet designed for the specific needs of all types of devices. Bootstrap[1] or other technologies can be used later to make the system more responsive.

(c) Nearby cities should be recommended — the prototype database had only 138 cities to check the functionalities of the system. With more cities added, the system is supposed to perform better.

7.4. Summary

In this chapter, we analysed the user data collected by us during the user study that ran for about two weeks. It was done mainly for the following — (a) to study the behaviour of the users, (b) to inspect the results provided by the system according to users' preferences, and (c) to scrutinise the feedback provided by users to the system.

The user behaviours were noticed in terms of where they want to go to, following which traveller role, and having what preferences for the city-based features. The recommendations provided to the users were analysed with regard to the cities that were recommended more frequently than others. The recommended duration of stays at the cities w.r.t. the total trip duration were also visualised to compare them amongst each other and with that recommended to travellers from different home regions.

The feedback provided by the users were discussed quantitatively and qualitatively. We enlisted the mean travel duration of users based on their age groups and their preferable frequency of vacations over a year. We discussed the user responses for the agreement-based feedback questions, and checked the overall user satisfaction based on their age groups. We also inspected the time users typically interacted with our system for, to provide preferences to get recommendations and to provide feedback. Finally, we discussed few additional feedback comments provided by the users as well.

[1] https://getbootstrap.com/docs/4.4/getting-started/introduction/

8. Conclusion & Future Work

In this master thesis, we designed and developed the first destination recommender system for computing personalised, composite city trips for any user, after analysing mobility data from location based social networks.

The related work available around designing of data-driven DRSs was assembled, presenting a detailed overview about the topic. It included discussions on travel mobility patterns, tourist recommender systems recommending individual and composite destinations, and duration of stay at different destinations. The world was divided into 10 regions and each city in our database was mapped with one of them. Trips extracted from Twitter check-ins were characterised into features based on travellers' home locations, mobility metrics and the types of cities visited. The study of travel mobility patterns helped model 10 prototype traveller types with characteristic features. The travellers from the 10 different regions of the world were then clustered, conforming to the modelled traveller types. A prototype web application, *TripRec* was developed in the thesis, that utilises travel preferences of a user to determine personalised city trips for them. The user inputs include the home region, destination region, traveller type, maximum travel duration and preferences for different types of venues in a city. On the other hand, the recommended trip consists of an ordered list of cities with duration of stay at each. The implementation of this recommender system marked the feasibility of utilising check-in data from LBSNs to generate composite city trips to different users. The evaluation of our application using the collected user data helped in better understanding of the limitations and getting ideas about the future scope for our RS.

Our recommendation algorithm compares the typical cost indices of cities with users' preferences during the selection of cities to be recommended. However, we do not include and display any information about the overall budget of a trip recommended to a user. Also, other information about how to move from one city to another is also not incorporated. These were out of scope for our present research and the prototype application. We can nevertheless incorporate these features — the budget information and the flight booking options — for the upcoming versions of TripRec. Point of interests specific to user preferences can be shown to the users, when demanded, for each of the recommended cities. This might enhance user satisfaction and ensure recommendation transparency.

The recommendations were limited by the availability of only 138 cities for our prototype application. Thus, the recommended ratios of the cities were found to be quite high in some destination regions. We can add more cities to our database to examine the functioning of the algorithms better. The world divisions can then also be made more granular, each region having enough cities.

We identified prominent traveller types in different world regions that also included those discovered in our earlier papers [19, 21]. The cities recommended to similar travellers based on user preferences worked fine. Users were also found to have overall liked *TripRec*. However, we can do more research to make the recommended trips more personalised. Construction of a composite city list for a trip can involve selection of individual cities based on (a) comparison of the cities with user preferences for the city-based features, and (b) distance between the cities. Our current version of the recommender system accounted for the first point, where the user preferences made the city selection customised as well. However, we considered distances between cities only during the possible second-round selection of cities after the removal of "unfit" cities. For future work, with more options of cities in the database, we can take account of the distances between them from the beginning during their selection. Moreover, different threshold distance values can be considered that are determined based on the radius of gyration of trips by different traveller types.

Further improvement can be in terms of the duration of stays recommended to different users at the selected cities. Rather than recommending identical duration of stay at a city for same type of travellers, we can adjust it proportionate to the maximum travel duration entered by a user, to make it more personalised.

Finally, we can also make our next prototype more responsive, so that users find it easier to specify their preferences to the system, while loading the web application from any device. More options can be provided to the users to elicit their interests, priorities, and posteriorities, for greater user satisfaction. Our recommendation algorithms can be adapted to accommodate the additional user inputs.

A. Regional Silhouette Plots

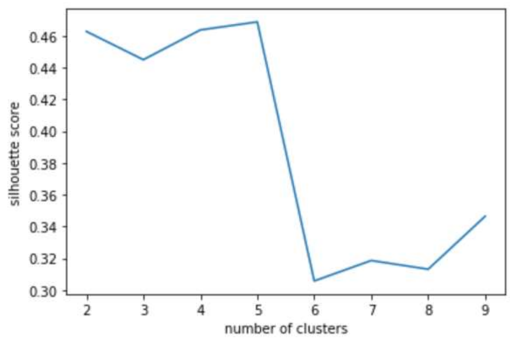


Figure A.1.: Silhouette plot for trips by travellers from North America

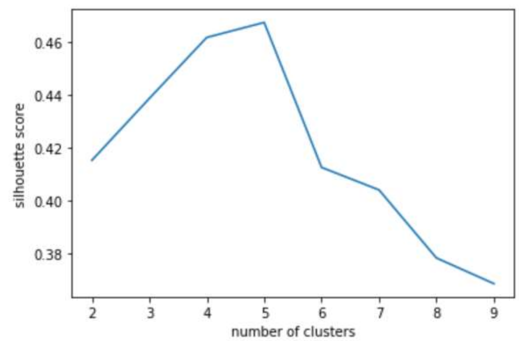


Figure A.2.: Silhouette plot for trips by travellers from South America

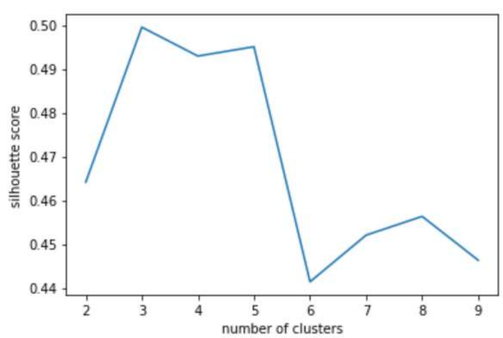


Figure A.3.: Silhouette plot for trips by travellers from North Europe

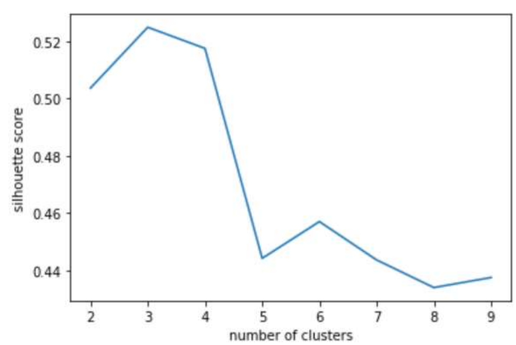


Figure A.4.: Silhouette plot for trips by travellers from Southwest Europe

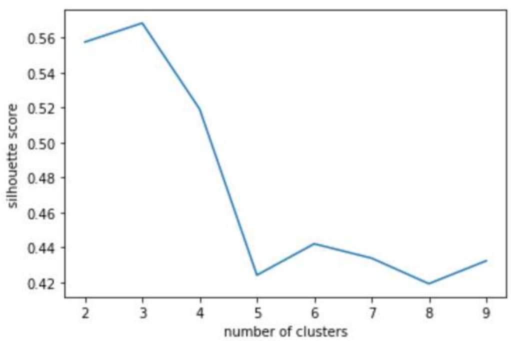


Figure A.5.: Silhouette plot for trips by travellers from Southeast Europe

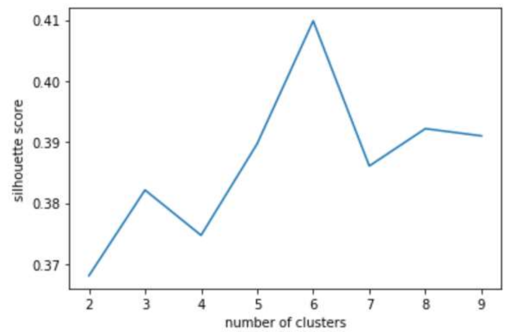


Figure A.6.: Silhouette plot for trips by travellers from North Africa

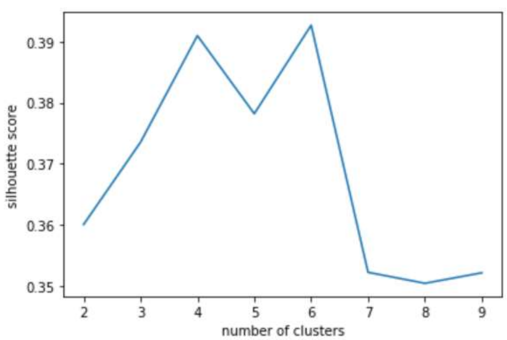


Figure A.7.: Silhouette plot for trips by travellers from South Africa

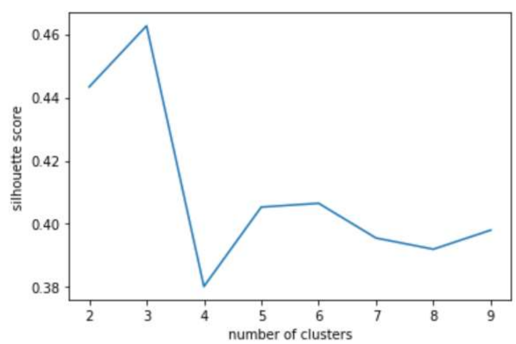


Figure A.8.: Silhouette plot for trips by travellers from West Asia

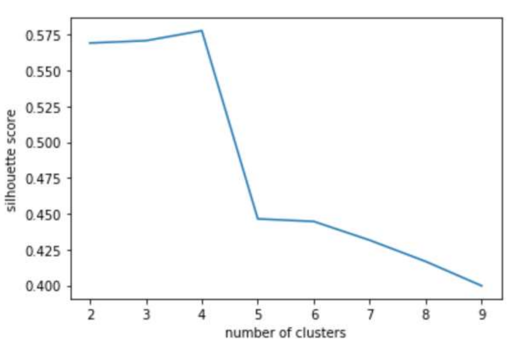


Figure A.9.: Silhouette plot for trips by travellers from East Asia

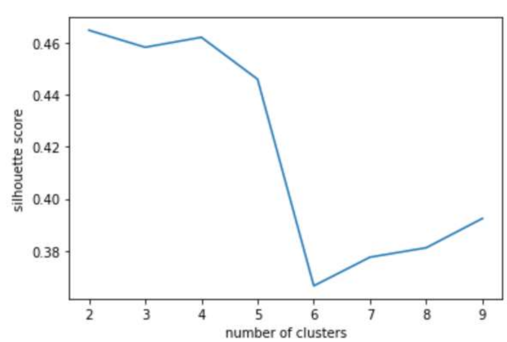


Figure A.10.: Silhouette plot for trips by travellers from Oceania

B. Feature Means of Regional Trip Clusters

Table B.1.: Cluster means for trips by travellers from North America

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Number of trips
13.56	2196.15	1.88	1.09	393.58	27308
17.20	5989.74	3.71	2.41	4601.28	4339
95.85	2876.72	4.35	1.37	709.65	1662
34.58	7738.70	8.13	4.64	2154.25	1927
13.84	8610.69	2.49	1.70	466.98	8344

Table B.2.: Cluster means for trips by travellers from South America

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Number of trips
14.61	8783.79	2.26	1.32	373.93	2764
24.65	9935.71	6.89	3.92	1222.94	794
14.34	1742.83	1.41	1.17	217.52	2933
17.13	6929.10	3.62	2.48	3986.68	738
117.82	5806.80	4.84	1.80	901.47	163

Table B.3.: Cluster means for trips by travellers from North Europe

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Number of trips
19.05	5878.35	3.60	2.45	3947.54	1647
13.79	927.48	1.66	1.33	160.87	7573
20.84	14047.73	2.51	1.79	1541.06	529
14.98	7455.60	1.97	1.16	371.72	3448
31.35	1450.49	5.59	3.51	883.33	1293

Table B.4.: Cluster means for trips by travellers from Southwest Europe

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Number of trips
13.96	730.93	1.80	1.31	149.25	6250
19.56	6146.70	3.77	2.54	3894.09	1357
17.74	8233.69	2.09	1.26	466.93	1715
29.62	1139.09	5.86	3.34	795.66	1146

Table B.5.: Cluster means for trips by travellers from Southeast Europe

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Number of trips
14.67	650.33	2.02	1.29	159.63	2478
24.75	1014.77	6.12	3.27	836.35	373
18.51	8984.69	2.38	1.33	677.37	402
20.95	5903.47	4.25	2.60	3787.94	424

Table B.6.: Cluster means for trips by travellers from North Africa

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Number of trips
16.36	7939.45	3.66	2.36	3591.66	47
15.53	5002.28	1.72	1.35	219.42	137
39.35	4316.29	7.15	3.92	2121.26	26
13.95	755.01	1.51	1.28	354.14	74
19.63	9836.79	2.05	1.16	538.19	75
93.46	5267.52	1.69	1.15	351.41	13

Table B.7.: Cluster means for trips by travellers from South Africa

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Number of trips
13.27	5195.46	1.47	1.32	170.92	92
101.40	8170.47	10.80	3.40	1560.26	5
18.76	8835.24	2.88	2.29	3581.49	34
17.00	6141.69	5.00	3.56	1266.98	25
15.39	12964.03	2.39	1.27	814.55	49
33.64	7844.02	6.91	4.91	5691.22	11

Table B.8.: Cluster means for trips by travellers from West Asia

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Number of trips
26.80	5934.54	6.07	3.48	2788.64	296
14.67	4574.85	1.88	1.37	252.21	1175
20.58	11951.29	3.11	1.47	1416.53	305

Table B.9.: Cluster means for trips by travellers from East Asia

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Number of trips
14.02	1269.37	1.88	1.18	186.27	6777
17.06	10583.85	2.71	1.43	518.36	2015
34.94	8807.73	8.89	4.08	1794.30	547
18.86	7670.74	3.98	2.63	5655.79	788

Table B.10.: Cluster means for trips by travellers from Oceania

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Number of trips
16.34	755.07	1.07	1.04	100.65	300
36.91	15535.24	7.40	4.19	2188.86	189
18.02	14728.87	2.66	1.43	758.42	727
23.38	11181.77	4.01	2.70	6025.26	188
15.39	6579.89	2.01	1.42	529.47	453

C. Regional Traveller Types Detected

Table C.1.: Traveller types detected in North America

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Traveller types
−	−	−	−	−−	Vacationers
−	+	−	−	+	Stopping Voyagers
++	+	−	−	−−	Long Explorers
+	++	+	+	+	Globetrotters
−	++	−	−	−−	Voyagers

Table C.2.: Traveller types detected in South America

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Traveller types
−	++	−	−	−−	Voyagers
+	++	+	+	−	Globetrotters
−	−	−	−	−−	Vacationers
−	+	−	−	+	Stopping Voyagers
++	+	−	−	−−	Long Explorers

Table C.3.: Traveller types detected in North Europe

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Traveller types
−	+	−	−	+	Stopping Voyagers
−	−−	−	−	−−	Vacationers
−	++	−	−	−	Voyagers
−	+	−	−	−−	Explorers
+	−	+	+	−−	Eurotrotters

Table C.4.: Traveller types detected in Southwest Europe

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Traveller types
−	−−	−	−	−−	Vacationers
−	+	−	−	+	Stopping Voyagers
−	++	−	−	−−	Voyagers
+	−	+	+	−−	Eurotrotters

Table C.5.: Traveller types detected in Southeast Europe

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Traveller types
−	−−	−	−	−−	Vacationers
+	−	+	+	−−	Eurotrotters
−	++	−	−	−−	Voyagers
−	+	−	−	+	Stopping Voyagers

Table C.6.: Traveller types detected in North Africa

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Traveller types
−	++	−	−	+	Stopping Voyagers
−	+	−	−	−−	Explorers
+	+	+	+	+	Globetrotters
−	−−	−	−	−−	Vacationers
−	++	−	−	−−	Voyagers
++	+	−	−	−−	Long Explorers

Table C.7.: Traveller types detected in South Africa

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Traveller types
−	+	−	−	−−	Explorers
++	++	++	+	−	Long Hop-trotters
−	++	−	−	+	Stopping Voyagers
−	+	+	+	−	Hop-trotters
−	++	−	−	−−	Voyagers
+	++	+	+	+	Globetrotters

Table C.8.: Traveller types detected in West Asia

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Traveller types
+	+	+	+	+	Globetrotters
−	+	−	−	−−	Explorers
−	++	−	−	−−	Voyagers

Table C.9.: Traveller types detected in East Asia

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Traveller types
−	−	−	−	−−	Vacationers
−	++	−	−	−−	Voyagers
+	++	+	+	−	Globetrotters
−	++	−	−	+	Stopping Voyagers

Table C.10.: Traveller types detected in Oceania

Travel duration	Mean displacement	Number of blocks	Number of countries	Radius of gyration	Traveller types
−	−−	−	−	−−	Vacationers
+	++	+	+	+	Globetrotters
−	++	−	−	−−	Voyagers
+	++	−	−	++	Long Stopping Voyagers
−	+	−	−	−−	Explorers

D. Recommendation Ratios of Cities (from User Study)

Table D.1.: Recommendation ratio of cities (Part I)

City (NE)	RR	City (SWE)	RR	City (SEE)	RR
Edinburgh	0.5185	Leipzig	0.5192	Bologna	0.5263
Stockholm	0.4815	Hamburg	0.4038	Venice	0.4474
Gothenburg	0.4074	Amsterdam	0.3846	Genoa	0.3158
London	0.3333	Rotterdam	0.3846	Innsbruck	0.2895
Dublin	0.3333	Dusseldorf	0.3462	Wroclaw	0.2632
Helsinki	0.2963	Frankfurt	0.3462	Graz	0.2369
Glasgow	0.2593	Nuremberg	0.3269	Florence	0.2369
York	0.2593	Brussels	0.3077	Warsaw	0.2105
Copenhagen	0.2593	Berlin	0.2885	Vienna	0.2105
Tallinn	0.2222	Luxembourg City	0.2885	Verona	0.1579
Saint Petersburg	0.1852	Cologne	0.2885	Salzburg	0.1579
Oslo	0.1852	Munich	0.2692	Krakow	0.1579
Cardiff	0.1852	Heidelberg	0.2692	Ljubljana	0.1579
Riga	0.1852	Marseille	0.2692	Budapest	0.1316
Vilnius	0.1481	Nantes	0.1923	Milan	0.1316
Moscow	0.0741	Paris	0.1731	Rome	0.1052
Reykjavik	0.0741	Barcelona	0.1346	Prague	0.0789
		Bern	0.1346	Naples	0.0789
		Nice	0.1154	Turin	0.0526
		Cordoba	0.0962	Athens	0.0526
		Seville	0.0769	Bucharest	0.0263
		Lisbon	0.0769		
		Cartagena	0.0577		
		Madrid	0.0577		
		Granada	0.0577		
		Palma de Mallorca	0.0385		
		Bilbao	0.0385		

Table D.2.: Recommendation ratio of cities (Part II)

City (NAm)	RR	City (SAm)	RR	City (WAs)	RR
Montreal	0.3125	Rio De Janeiro	0.5556	Jerusalem	0.8333
Victoria	0.3125	Medellin	0.5556	Tel Aviv	0.7778
Quebec City	0.3125	Sao Paulo	0.5556	Dubai	0.7500
Miami	0.3125	Montevideo	0.4444	Delhi	0.5278
Chicago	0.3125	Fortaleza	0.4444	Mumbai	0.1944
Calgary	0.2500	Lima	0.3333	**City (EAs)**	**RR**
Cancun	0.2500	Quito	0.3333	Jakarta	0.5625
San Jose	0.2500	Santiago	0.3333	Kyoto	0.5625
Washington DC	0.2500	Caracas	0.2222	Hiroshima	0.5000
Las Vegas	0.2500	Buenos Aires	0.1111	Kobe	0.5000
New York City	0.1875	Bogota	0.1111	Denpasar	0.4375
Seattle	0.1875	Asuncion	0.1111	Osaka	0.3750
Honolulu	0.1875	**City (NAf)**	**RR**	Singapore	0.3125
Los Angeles	0.1875	Marrakesh	1.0000	Guangzhou	0.2500
Toronto	0.1875	Cairo	0.8000	Busan	0.1875
New Orleans	0.1250	Lagos	0.8000	Phuket City	0.1875
Pittsburgh	0.1250	Casablanca	0.6000	Hong Kong	0.1875
Vancouver	0.1250	Accra	0.6000	Seoul	0.1875
Charleston	0.1250	**City (SAf)**	**RR**	Shenzhen	0.1875
Puerto Vallarta	0.1250	Nairobi	1.0000	Manila	0.1250
Boston	0.1250	Kampala	1.0000	Taipei	0.1250
San Francisco	0.1250	**City (O)**	**RR**	Okinawa	0.1250
Mexico City	0.1250	Melbourne	1.0000	Bangkok	0.0625
Memphis	0.0625	Sydney	0.7500	Ho Chi Minh City	0.0625
Minneapolis	0.0625			Phnom Penh	0.0625
San Diego	0.0625				
Orlando	0.0625				
Santa Fe	0.0625				

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] G. Adomavicius and A. Tuzhilin. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." In: *IEEE Trans. on Knowl. and Data Eng.* 17.6 (June 2005), pp. 734–749. ISSN: 1041-4347. DOI: 10.1109/TKDE.2005.99.

[2] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. "Intrigue: Personalized Recommendation of Tourist Attractions for Desktop and Hand Held Devices." In: *Applied Artificial Intelligence* 17.8-9 (2003), pp. 687–714. DOI: 10.1080/713827254.

[3] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. "Ubiquitous User Assistance in a Tourist Information Server." In: *Adaptive Hypermedia and Adaptive Web-Based Systems, Second International Conference, AH 2002, Malaga, Spain, May 29-31, 2002, Proceedings*. 2002, pp. 14–23. DOI: 10.1007/3-540-47952-X_4.

[4] D. Arribas-Bel, K. Kourtit, and P. Nijkamp. "Benchmarking of world cities through Self-Organizing Maps." In: *Cities* 31 (Apr. 2013), pp. 248–257. DOI: https://doi.org/10.1016/j.cities.2012.06.019.

[5] C. Ash, B. R. Jasny, L. Roberts, R. Stone, and A. M. Sugden. "Reimagining Cities." In: *Science* 319 (5864 Feb. 2008), p. 739. DOI: 10.1126/science.319.5864.739.

[6] J. Bao, Y. Zheng, and M. F. Mokbel. "Location-based and Preference-aware Recommendation Using Sparse Geo-social Networking Data." In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. SIGSPATIAL '12. Redondo Beach, California: ACM, 2012, pp. 199–208. ISBN: 978-1-4503-1691-0. DOI: 10.1145/2424321.2424348.

[7] H. Bapierre. "Context Specific Next Location Prediction." PhD thesis. Technical University of Munich, 2014.

[8] L. M. A. Bettencourt, J. Lobo, D. Strumsky, and G. B. West. "Urban Scaling and Its Deviations: Revealing the Structure of Wealth, Innovation and Crime across Cities." In: *PLOS ONE* 5.11 (Nov. 2010), pp. 1–9. DOI: 10.1371/journal.pone.0013541.

[9] Bing Wu, Y. Murata, N. Shibata, K. Yasumoto, and M. Ito. "A method for composing tour schedules adaptive to weather change." In: *2009 IEEE Intelligent Vehicles Symposium*. June 2009, pp. 1407–1412. DOI: 10.1109/IVS.2009.5164491.

[10] D. Buhalis. "Information Technologies in tourism: Implications for the tourism curriculum." In: *Information and Communication Technologies in Tourism 1998*. Ed. by D. Buhalis, A. M. Tjoa, and J. Jafari. Vienna: Springer Vienna, 1998, pp. 289–297. ISBN: 978-3-7091-7504-0.

[11] *Calculate distance, bearing and more between Latitude/Longitude points*. URL: https://www.movable-type.co.uk/scripts/latlong.html.

[12] L. Cao, J. Luo, A. Gallagher, X. Jin, J. Han, and T. Huang. "A worldwide tourism recommendation system based on geotagged web photos." English (US). In: *2010 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010 - Proceedings*. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. Nov. 2010, pp. 2274–2277. ISBN: 9781424442966. DOI: 10.1109/ICASSP.2010.5495905.

[13] L. A. Castillo, E. Armengol, E. Onaindia, L. Sebastia, J. González-Boticario, A. Rodrıguez, S. Fernández, J. D. Arias, and D. Borrajo. "samap: An user-oriented adaptive system for planning tourist visits." In: *Expert Syst. Appl.* 34.2 (2008), pp. 1318–1332. DOI: 10.1016/j.eswa.2006.12.029.

[14] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou. "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '00. The Hague, The Netherlands: ACM, 2000, pp. 17–24. ISBN: 1-58113-216-6. DOI: 10.1145/332040.332047.

[15] J. Cranshaw, R. Schwartz, J. I. Hong, and N. M. Sadeh. "The Livehoods Project: Utilizing Social Media to Understand the Dynamics of a City." In: *Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012*. 2012.

[16] M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. "Automatic Construction of Travel Itineraries Using Social Breadcrumbs." In: *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*. HT '10. Toronto, Ontario, Canada: Association for Computing Machinery, 2010, pp. 35–44. ISBN: 9781450300414. DOI: 10.1145/1810617.1810626.

[17] L. W. Dietz. "Data-Driven Destination Recommender Systems." In: *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*. UMAP '18. Singapore, Singapore: Association for Computing Machinery, 2018, pp. 257–260. ISBN: 9781450355896. DOI: 10.1145/3209219.3213591.

[18] L. W. Dietz, D. Herzog, and W. Wörndl. "Deriving Tourist Mobility Patterns from Check-in Data." In: *Proceedings of the WSDM 2018 Workshop on Learning from User Interactions*. Los Angeles, CA, USA, 2018.

[19] L. W. Dietz, R. Roy, and W. Wörndl. "Characterisation of Traveller Types Using Check-In Data from Location-Based Social Networks." In: *Information and Communication Technologies in Tourism 2019, ENTER 2019, Proceedings of the International Conference in Nicosia, Cyprus, January 30-February 1, 2019*. 2019, pp. 15–26. doi: 10.1007/978-3-030-05940-8_2.

[20] L. W. Dietz, M. Saadi, and W. Wörndl. "Designing a Conversational Travel Recommender System Based on Data-Driven Destination Characterization." In: *ACM RecSys Workshop on Recommenders in Tourism*. RecTour 2019. Copenhagen, Denmark, 2019, pp. 17–21.

[21] L. W. Dietz, A. Sen, R. Roy, and W. Wörndl. "Mining trips from location-based social networks for clustering travelers and destinations." In: *Information Technology & Tourism* (2020). issn: 1943-4294. doi: 10.1007/s40558-020-00170-6.

[22] L. W. Dietz and A. Weimert. "Recommending Crowdsourced Travels on wOndary." In: *Proceedings of the Workshop on Recommenders in Tourism, RecTour 2018, co-located with the 12th ACM Conference on Recommender Systems (RecSys 2018), Vancouver, Canada, October 7, 2018*. 2018, pp. 13–17.

[23] L. W. Dietz and W. Wörndl. "How Long to Stay Where? On the Amount of Item Consumption in Travel Recommendation." In: *Proceedings of ACM RecSys 2019 Late-Breaking Results co-located with the 13th ACM Conference on Recommender Systems, RecSys 2019 Late-Breaking Results, Copenhagen, Denmark, September 16-20, 2019*. 2019, pp. 31–35.

[24] E. W. Dijkstra. "A Note on Two Problems in Connexion with Graphs." In: *Numer. Math.* 1.1 (Dec. 1959), pp. 269–271. issn: 0029-599X. doi: 10.1007/BF01386390.

[25] *Direct contribution of travel and tourism to GDP in leading countries worldwide in 2019*. 2020. url: https://www.statista.com/statistics/292461/contribution-of-travel-and-tourism-to-gdp-in-select-countries/ (visited on 02/25/2020).

[26] Z. Friggstad, S. Gollapudi, K. Kollias, T. Sarlos, C. Swamy, and A. Tomkins. "Orienteering Algorithms for Generating Travel Itineraries." In: *International Conference on Web Search and Data Mining (WSDM)*. 2018.

[27] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. E. Pantziou. "A survey on algorithmic approaches for solving tourist trip design problems." In: *J. Heuristics* 20.3 (2014), pp. 291–328. doi: 10.1007/s10732-014-9242-5.

[28] Z. W. Geem, C. Tseng, and Y. Park. "Harmony Search for Generalized Orienteering Problem: Best Touring in China." In: *Advances in Natural Computation, First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part III*. 2005, pp. 741–750. DOI: 10.1007/11539902_91.

[29] F. Girardin, F. Calabrese, F. D. Fiore, C. Ratti, and J. Blat. "Digital Footprinting: Uncovering Tourists with User-Generated Content." In: *IEEE Pervasive Computing* 7.4 (2008), pp. 36–43. DOI: 10.1109/MPRV.2008.71.

[30] M. C. González, C. A. Hidalgo, and A. L. Barabási. "Understanding individual human mobility patterns." In: *Nature* 453 (2008), pp. 779–782.

[31] S. Grauwin, S. Sobolevsky, S. Moritz, I. Gódor, and C. Ratti. "Towards a comparative science of cities: using mobile traffic records in New York, London and Hong Kong." In: *CoRR* abs/1406.4400 (2014). arXiv: 1406.4400.

[32] T. Griffin, Y. Huang, and R. Halverson. "Computerized trip classification of GPS data." In: *Proceedings of 3rd international conference on cybernetics and information technologies, systems and applications (CITSA 2006)*. 2006, pp. 22–30.

[33] S. Hanson and O. J. Huff. "Systematic variability in repetitious travel." In: *Transportation* 15.1 (1988), pp. 111–135. ISSN: 1572-9435. DOI: 10.1007/BF00167983.

[34] B. Hawelka, I. Sitko, E. Beinat, S. Sobolevsky, P. Kazakopoulos, and C. Ratti. "Geo-located Twitter as proxy for global mobility patterns." In: *Cartography and Geographic Information Science* 41.3 (2014). PMID: 27019645, pp. 260–271. DOI: 10.1080/15230406.2014.890072. eprint: https://doi.org/10.1080/15230406.2014.890072.

[35] D. Herzog, L. W. Dietz, and W. Wörndl. "Tourist Trip Recommendations – Foundations, State of the Art and Challenges." In: *Personalized Human-Computer Interaction*. Ed. by M. Augstein, E. Herder, and W. Wolfgang. Berlin, Germany: De Gruyter Oldenbourg, 2019, pp. 159–182. ISBN: 978-3-11-055247-8.

[36] D. Herzog and W. Wörndl. "A Travel Recommender System for Combining Multiple Travel Regions to a Composite Trip." In: *Proceedings of the 1st Workshop on New Trends in Content-based Recommender Systems co-located with the 8th ACM Conference on Recommender Systems, CBRecSys@RecSys 2014, Foster City, Silicon Valley, California, USA, October 6, 2014*. 2014, pp. 42–48.

[37] Y. Hu, X. Ye, and S. Shaw. "Extracting and analyzing semantic relatedness between cities using news articles." In: *International Journal of Geographical Information Science* 31.12 (2017), pp. 2427–2451. DOI: 10.1080/13658816.2017.1367797.

[38] R. Kramer, M. Modsching, and K. T. Hagen. "A City Guide Agent Creating and Adapting Individual Sightseeing Tours Based on Field Trial Results." In: *International Journal of Computational Intelligence Research* 2.2 (2006), pp. 191–206. ISSN: 0973-1873.

[39] Q. Liu, E. Chen, H. Xiong, Y. Ge, Z. Li, and X. Wu. "A Cocktail Approach for Travel Package Recommendation." In: *IEEE Trans. on Knowl. and Data Eng.* 26.2 (Feb. 2014), pp. 278–293. ISSN: 1041-4347. DOI: `10.1109/TKDE.2012.233`.

[40] Q. Liu, Y. Ge, Z. Li, E. Chen, and H. Xiong. "Personalized Travel Package Recommendation." In: *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*. ICDM '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 407–416. ISBN: 978-0-7695-4408-3. DOI: `10.1109/ICDM.2011.118`.

[41] S. Long, R. Kooper, G. D. Abowd, and C. G. Atkeson. "Rapid Prototyping of Mobile Context-aware Applications: The Cyberguide Case Study." In: *Proceedings of the 2Nd Annual International Conference on Mobile Computing and Networking*. MobiCom '96. Rye, New York, USA: ACM, 1996, pp. 97–107. ISBN: 0-89791-872-X. DOI: `10.1145/236387.236412`.

[42] T. Louail, M. Lenormand, O. G. Cantu Ros, M. Picornell, R. Herranz, E. Frias-Martinez, J. J. Ramasco, and M. Barthelemy. "From mobile phone data to the spatial structure of cities." In: *Scientific Reports* 4.1 (2014), p. 5276. ISSN: 2045-2322. DOI: `10.1038/srep05276`.

[43] D. Massimo and F. Ricci. "Clustering Users' POIs Visit Trajectories for Next-POI Recommendation." In: *Information and Communication Technologies in Tourism 2019*. Ed. by J. Pesonen and J. Neidhardt. Cham: Springer International Publishing, 2019, pp. 3–14. ISBN: 978-3-030-05940-8.

[44] J. Melià-Seguí, R. Zhang, E. Bart, B. Price, and O. Brdiczka. "Activity duration analysis for context-aware services using foursquare check-ins." In: *Self-IoT '12*. 2012.

[45] A. Noulas, S. Scellato, R. Lambiotte, M. Pontil, and C. Mascolo. "A Tale of Many Cities: Universal Patterns in Human Urban Mobility." In: *PloS one*. 2012.

[46] G. M. P. O'Hare and M. J. O'Grady. "Gulliver's Genie: A Multi-agent System for Ubiquitous and Intelligent Content Delivery." In: *Comput. Commun.* 26.11 (July 2003), pp. 1177–1187. ISSN: 0140-3664. DOI: `10.1016/S0140-3664(02)00252-9`.

[47] R. Olson. *The optimal U. S. national parks centennial road trip*. `http://www.randalolson.com/2016/07/30/the-optimal-u-s-national-parks-centennial-road-trip/`. 2016.

[48] R. Oppermann and M. Specht. "A Nomadic Information System for Adaptive Exhibition Guidance." In: *Archives and Museum Informatics* 13.2 (1999), pp. 127–138. ISSN: 1573-7500. DOI: 10.1023/A:1016619506241.

[49] S. Paldino, I. Bojic, S. Sobolevsky, C. Ratti, and M. C. González. "Urban magnetism through the lens of geo-tagged photography." In: *EPJ Data Sci.* 4.1 (2015), p. 5. DOI: 10.1140/epjds/s13688-015-0043-3.

[50] V. R. Preedy and R. R. Watson. "5-Point Likert Scale." In: *Handbook of Disease Burdens and Quality of Life Measures*. New York, NY: Springer New York, 2010, pp. 4288–4288. ISBN: 978-0-387-78665-0. DOI: 10.1007/978-0-387-78665-0_6363.

[51] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft. "Recommending Social Events from Mobile Phone Location Data." In: *Proceedings of the 2010 IEEE International Conference on Data Mining*. ICDM '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 971–976. ISBN: 978-0-7695-4256-0. DOI: 10.1109/ICDM.2010.152.

[52] S. Rani, K. N. Kholidah, and S. N. Huda. "A Development of Travel Itinerary Planning Application using Traveling Salesman Problem and K-Means Clustering Approach." In: *Proceedings of the 7th International Conference on Software and Computer Applications, ICSCA 2018, Kuantan, Malaysia, February 08-10, 2018*. 2018, pp. 327–331. DOI: 10.1145/3185089.3185142.

[53] C. Ratti, S. Sobolevsky, F. Calabrese, C. Andris, J. Reades, M. Martino, R. Claxton, and S. H. Strogatz. "Redrawing the Map of Great Britain from a Network of Human Interactions." In: *PLOS ONE* 5.12 (Dec. 2010), pp. 1–6. DOI: 10.1371/journal.pone.0014248.

[54] H. Ritchie and M. Roser. *Urbanization*. URL: https://ourworldindata.org/urbanization.

[55] R. Roy and L. W. Dietz. "Modeling Physiological Conditions for Proactive Tourist Recommendations." In: *Proceedings of the 23rd International Workshop on Personalization and Recommendation on the Web and Beyond*. ABIS '19. Hof, Germany: ACM, 2019, pp. 25–27. ISBN: 978-1-4503-6896-4. DOI: 10.1145/3345002.3349289.

[56] A. Sen and L. W. Dietz. "Identifying Travel Regions Using Location-Based Social Network Check-in Data." In: *Frontiers in Big Data* 2 (2019), p. 12. ISSN: 2624-909X. DOI: 10.3389/fdata.2019.00012.

[57] A. Sevtsuk and C. Ratti. "Does Urban Mobility Have a Daily Routine? Learning from the Aggregate Data of Mobile Networks." In: *Journal of Urban Technology* 17.1 (2010), pp. 41–60. DOI: 10.1080/10630731003597322. eprint: https://doi.org/10.1080/10630731003597322.

[58] T. H. Silva, P. O. S. V. de Melo, J. M. Almeida, M. Musolesi, and A. A. F. Loureiro. "You Are What You Eat (and Drink): Identifying Cultural Boundaries by Analyzing Food and Drink Habits in Foursquare." In: *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014*. 2014.

[59] S. Sobolevsky, I. Bojic, A. Belyi, I. Sitko, B. Hawelka, J. M. Arias, and C. Ratti. "Scaling of City Attractiveness for Foreign Visitors through Big Data of Human Economical and Social Media Activity." In: *2015 IEEE International Congress on Big Data, New York City, NY, USA, June 27 - July 2, 2015*. 2015, pp. 600–607. DOI: `10.1109/BigDataCongress.2015.92`.

[60] S. Sobolevsky, I. Sitko, R. T. des Combes, B. Hawelka, J. M. Arias, and C. Ratti. "Cities through the Prism of People's Spending Behavior." In: *CoRR* abs/1505.03854 (2015). arXiv: `1505.03854`.

[61] S. Sobolevsky, M. Szell, R. Campari, T. Couronné, Z. Smoreda, and C. Ratti. "Delineating geographical regions with networks of human interactions in an extensive set of countries." In: *CoRR* abs/1310.1829 (2013). arXiv: `1310.1829`.

[62] V.-W. Soo and S.-H. Liang. "Recommending a Trip Plan by Negotiation with a Software Travel Agent." In: *Cooperative Information Agents V*. Ed. by M. Klusch and F. Zambonelli. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 32–37. ISBN: 978-3-540-44799-3.

[63] K. Sparks, G. Thakur, A. Pasarkar, and M. Urban. "A global analysis of cities' geosocial temporal signatures for points of interest hours of operation." In: *International Journal of Geographical Information Science* 0.0 (2019), pp. 1–18. DOI: `10.1080/13658816.2019.1615069`. eprint: `https://doi.org/10.1080/13658816.2019.1615069`.

[64] Y. Sun and L. Lee. "AGENT-BASED PERSONALIZED TOURIST ROUTE ADVICE SYSTEM." In: *SPRS Congress Istanbul 2004, Proceedings of Commission II*. 2004, pp. 319–324.

[65] P. Vansteenwegen and W. Souffriau. "Trip Planning Functionalities: State of the Art and Future." In: *J. of IT & Tourism* 12.4 (2010), pp. 305–315. DOI: `10.3727/109830511X13049763021853`.

[66] P. Vansteenwegen and D. Van Oudheusden. "The Mobile Tourist Guide: An OR Opportunity." In: *OR Insight* 20.3 (2007), pp. 21–27. ISSN: 1759-0477. DOI: `10.1057/ori.2007.17`.

[67] W. Wörndl. "A Web-based Application for Recommending Travel Regions." In: *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. UMAP '17. Bratislava, Slovakia: ACM, 2017, pp. 105–106. ISBN: 978-1-4503-5067-9. DOI: 10.1145/3099023.3099031.

[68] W. Wörndl, A. Hefele, and D. Herzog. "Recommending a sequence of interesting places for tourist trips." In: *J. of IT & Tourism* 17.1 (2017), pp. 31–54. DOI: 10.1007/s40558-017-0076-5.

[69] L. Wu, Y. Zhi, Z. Sui, and Y. Liu. "Intra-Urban Human Mobility and Activity Transition: Evidence from Social Media Check-In Data." In: *PLOS ONE* 9.5 (May 2014), pp. 1–13. DOI: 10.1371/journal.pone.0097010.

[70] M. Xie, L. V. S. Lakshmanan, and P. T. Wood. "Composite recommendations: from items to packages." In: *Frontiers Comput. Sci.* 6.3 (2012), pp. 264–277. DOI: 10.1007/s11704-012-2014-1.

[71] C.-C. Yu and H.-p. Chang. "Personalized Location-Based Recommendation Services for Tour Planning in Mobile Tourism Applications." In: *E-Commerce and Web Technologies*. Ed. by T. Di Noia and F. Buccafurri. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 38–49. ISBN: 978-3-642-03964-5.

[72] J. Yuan, Y. Zheng, and X. Xie. "Discovering Regions of Different Functions in a City Using Human Mobility and POIs." In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '12. Beijing, China: ACM, 2012, pp. 186–194. ISBN: 978-1-4503-1462-6. DOI: 10.1145/2339530.2339561.

[73] Y. Zheng and X. Xie. "Learning travel recommendations from user-generated GPS traces." In: *ACM TIST* 2.1 (2011), 2:1–2:29. DOI: 10.1145/1889681.1889683.