

Article

Combining Cluster-Based Profiling Based on Social Media Features and Association Rule Mining for Personalised Recommendations of Touristic Activities

Jonathan Ayebakuro Orama ^{1,*} , Joan Borràs ¹  and Antonio Moreno ² 

¹ Eurecat, Centre Tecnològic de Catalunya, Unit of Tourism Innovation, C/ Joanot Martorell, 15, 43480 Vila-seca, Spain; joan.borras@eurecat.org

² Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA) Research Group, Computer Science and Mathematics Department, Av. Països Catalans, 26, 43007 Tarragona, Spain; antonio.moreno@urv.cat

* Correspondence: jonathan.orama@eurecat.org; Tel.: +34-674-885-859

Abstract: Tourists who visit a city for the first time may find it difficult to decide on places to visit, as the amount of information in the Web about cultural and leisure activities may be large. Recommender systems address this problem by suggesting the points of interest that fit better with the user's preferences. This paper presents a novel recommender system that leverages tweets to build user profiles, taking into account not only their personal preferences but also their travel habits. Association rules, which are mined from the previous visits of users documented on Twitter, are used to make the final recommendations of places to visit. The system has been applied to data of the city of Barcelona, and the results show that the use of the social media-based clustering procedure increases its performance according to several relevant metrics.

Keywords: recommender systems; user profiling; social media; cluster analysis; association rule mining



Citation: Orama, J.A.; Borràs, J.; Moreno, A. Combining Cluster-Based Profiling Based on Social Media Features and Association Rule Mining for Personalised Recommendations of Touristic Activities. *Appl. Sci.* **2021**, *11*, 6512. <https://doi.org/10.3390/app11146512>

Academic Editor: Pasi Fränti

Received: 14 June 2021

Accepted: 13 July 2021

Published: 15 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The tourism industry has seen steady growth in the couple of decades. There was a 6% average increase per year in international tourist arrivals within the period 2010–2019, and there has been 117% overall growth since the year 2000 [1,2]. This has spurred the creation of more tourist activities, which makes it hard for visitors to choose the ones that fit better with their preferences and travel habits. Online travel guide services such as TripAdvisor, Yelp and Booking.com have become very popular among tourists seeking to reduce the information overload. However, visitors still need to sort through ratings and reviews to make decisions on attractions to visit. *Recommender systems* (RS) provide a solution to this problem, because they can suggest suitable attractions tailored to a user's preferences without human intervention. The development of accurate RS in the tourism field has been a hot topic of research in the last ten years. It has spawned different specialisations, such as personalised RS [3], context-aware RS [4], sequence-aware RS [5], group RS [6] and hybrid RS [7]. These categories focus on different kinds of situations in which recommendations are relevant [8].

In tourism research, data are typically obtained from surveys carried out by international tourism statistics and regulation bodies, such as the United National World Tourism Organization (UNWTO), or from smaller scale surveys carried out personally by the researchers themselves or by research institutes. In most cases, these surveys are undertaken using questionnaires, and they may include GPS devices handed to tourists to analyse their mobility and trajectory patterns [9]. In recent years, social media has also become a very valuable data source for tourism analysis because of its availability and the significant online presence of tourists on social networks. The main hypothesis of this work is that recommender systems can also profit from the messages shared on social media to gain insights into a user's travel preferences.

In this paper, we present a tourism recommender system that utilises features extracted from tweets of tourists in order to create user profiles which are employed to create personalised recommendations of touristic activities. The features, extracted from the activity of the users on Twitter, represent not only the users' cultural preferences, but also the context, their travel habits and the popularity of the visited *points of interest* (POIs). We propose a recommendation algorithm that combines in a novel way clustering based on social media features with association rule mining to find the preferred combinations of POIs from those visited by similar users. These combinations capture in a novel way the *relatedness* of certain activities, which may be due to their similarity, their physical proximity, the ease of travel from one to the other and even their popularity. Additionally, the system ranks the mined association rules using methods that adapt to the characteristics of the user to ensure that the recommended POIs fit the user's unique interests and his/her attraction towards popular or unpopular POIs. The performance of the system has been evaluated with the initial clustering step and without it, to study the influence of the social media-based clustering in the recommendation process. The obtained results confirm the usefulness of the clustering phase to improving the performance of the recommender in several metrics.

In summary, the main contributions of this work are the following:

1. We introduce a method for analysing data from social media to build user profiles that encapsulate their travel preferences and habits.
2. We present insights into the potential benefits of the combination of cluster analysis and association rule mining in tourism recommenders.
3. We provide the results of in-depth experiments using a comprehensive set of numerical evaluation metrics to gauge the benefits of social media-based clustering for user profiling.

The rest of the paper is organised as follows. Section 2 summarises works related to the construction of user profiles using information from social media and the use of clustering and association rule mining in recommender systems. Section 3 details the proposed system, including data sources, the pre-processing steps and the different stages of the recommendation process. Section 4 provides details of the experimentation, proposed evaluation and analysis of the results. Finally, Section 5 concludes the paper and outlines some possible lines of future work.

2. Related Work

2.1. Social Media User Profiling

Social networks have become one of the primary data sources for the analysis of tourist activities in the last decade, due to the enormous amount of heterogeneous information that they provide (messages, pictures, videos, opinions, ratings, etc.). Some works have shown the general reliability of these data for comprehensive studies, by checking the correlation between check-ins on social media and standard cellphone and survey data [10]. Research that uses social media data in fields such as recommender systems [11], spatial analysis [12–15] and human behaviour modelling [16] has become increasingly common.

In particular, recommender systems need to know what the *preferences* of users are in order to provide appropriate personalised suggestions. One possible way to discover these preferences is to analyse the information provided by users in social media. It is indeed feasible to take into account not only the textual content of the messages they send, but also additional information, such as the moment in which they are sent (day of the week and time of the day), the language in which messages are written, the exact geospatial location from which each message is sent, etc. Thus, different features may be extracted from the social media data in order to build user profiles. Having a personalised profile for each user makes it possible to apply clustering procedures to detect different types of users. For example, previous works have suggested that the text in tweets could be used to capture emotions and cluster users by their personalities, so that it would be possible to make travel recommendations to similar users [17].

Incorporating information about the *context* of the users to improve recommendations is an active research area in recommender systems; however, most projects focus only on extracting and analyzing geo-located check-ins from location-based social networks (LBSNs). A common pre-processing step is to study both the locations of the messages and the users' public information in order to distinguish local citizens from tourists in a particular destination [15,18,19]. It is also possible to infer the purpose of a trip by utilising check-ins, timestamps and point-of-interest labels from Gowalla and Foursquare [20]. Check-in information has also been used to study the trajectory of users within a city, allowing next-POI recommendations to be made to users based on historical data [20–23]. Some works also included temporal and social factors extracted from check-in information [24]. When dealing with check-ins, it is often necessary to link a geolocation to a specific POI name and category, especially in travel planning. Some LBSNs such as TripAdvisor or Gowalla provide these labels, but in other cases they must be inferred. The option that was chosen in the work reported in this paper was to make queries to the Open Street Map (OSM) server using the Overpass engine to assign POIs to tweets based on proximity and usefulness to tourists [25].

There are also works that combined different data modalities or data from different social networks when creating user profiles. For example, Farnadi et al. proposed a hybrid deep neural network that aggregates textual, visual and social relational data extracted from Facebook profiles into a user profile which is evaluated by predicting the user's age, gender and personality [26]. A platform-independent system that automatically extracts textual features, including comments, check-in labels and links, from multiple social networks to build user profiles was proposed by Orlandi et al. [27]. These profiles, which represent the interests of users, were enriched with information from DBpedia [28].

2.2. Improving Recommendations with Clustering and Association Rule Mining

Context-aware recommender systems [4] try to improve the quality of the recommendations by incorporating data about the current state of a user (position, emotions, personality, etc.) and his/her environment (e.g., weather and traffic conditions), or even the user's social media information (e.g., followers and followees in Twitter). For example, Esmaili et al. incorporated trust, reputation and user relationships in social communities into a collaborative filtering recommender to provide improved recommendations of tourism products [29].

One of the possible solutions to the *cold start* and *data sparsity* problems in recommender systems is to *cluster* users or items to gain generalised information about them. For instance, Liji et al. proposed an evolutionary algorithm to cluster user attributes before building a user–item matrix for collaborative filtering [30]. Ma et al. went a step further by combining three clustering processes (based on user trust, user similarity and item similarity) to form the user–item matrix and then using the matrix factorisation model to make predictions [31]. Following the same trend, Nguyen et al. proposed grouping users by their cognitive similarity, determined by their interest in similar items, to handle the cold start and data sparsity problems [32,33]. Along the same lines of user similarity, Fränti et al. compared the similarity of users via different factors (location visit frequencies, opinions and liked pages) to improve the recommendations [34]. The same authors also considered the sparse location histories of mobile users to find similar users even if a user's trajectory was incomplete [35].

Another artificial intelligence technique that may be useful in recommender systems is *association rule mining* (ARM). Rules provide common relationships between items (objects frequently bought together, or POIs frequently visited together), which may be helpful in the recommendation process. For example, the use of association rules has been suggested to recommend songs [36]. More concretely, song clusters are used to build a user's profile, and then rules are mined from the user's listening history to make recommendations that fit his/her preferences.

The use of ARM in tandem with clustering is a popular concept that has been successfully applied in several fields. Pandya et al. proposed this combination to battle data sparsity and cold start problems in e-commerce, using k-means for user-item matrix clustering and mining association rules from Boolean data extracted from user clusters [37]. Similarly, Jalalimanesh et al. proposed a recommender system in the inter-library domain, to recommend books by assigning users to clusters (built on users' categorical features extracted from library logs) that are attached to association rules mined using decision trees [38]. Despite the popularity of this concept, it has been scarcely applied in the field of tourism. Fenza et al. proposed a tourism recommender system based on separate fuzzy clustering of users and POIs, allowing new users and POIs to be assigned to clusters. Users are then matched with previously mined association rules that relate users and their context to POI clusters. However, this system was specifically tailored to the data collected by their application and can not be used out of the bounds of their project [39].

In the work reported in this paper, we propose a new combination of clustering and association rule mining to improve the recommendations of POIs within a city. It also takes into account the information provided by the users on Twitter.

3. A Recommender of Tourist Activities Based on Clustering and Rule Mining

3.1. System Overview

This section presents a recommender system for tourist activities, which combines in a novel way the use of clustering to aggregate users that have similar preferences when visiting a city (according to the data they provide on social media) and the use of association rules, which indicate the preferred combinations of items for classes of users (also obtained from the analysis of the sequences of POIs visited by users, registered in their social media posts).

User profiles are not built with a specific generalisation in mind (for instance, personality [17]). They are solely based on the features extracted from Twitter that embed users' interests, travel habits and degrees of interest in popular items.

Figure 1 presents the architecture of the system, which has three basic stages. First, data from a collection of users that have visited a destination is collected from Twitter, pre-processed and stored in a PostgreSQL database. Some pertinent information about geolocations is extracted from Open Street Map. Features are then extracted after filtering unwanted data. In the second stage, a clustering process is performed to uncover user profiles (groups of visitors with similar characteristics, preferences and patterns of travel). In the final stage a set of association rules is mined for each cluster. These rules describe POIs frequently visited together by the members of that cluster. Finally, these rules are employed to provide personalised recommendations to a particular user. The following subsections describe these stages in detail.

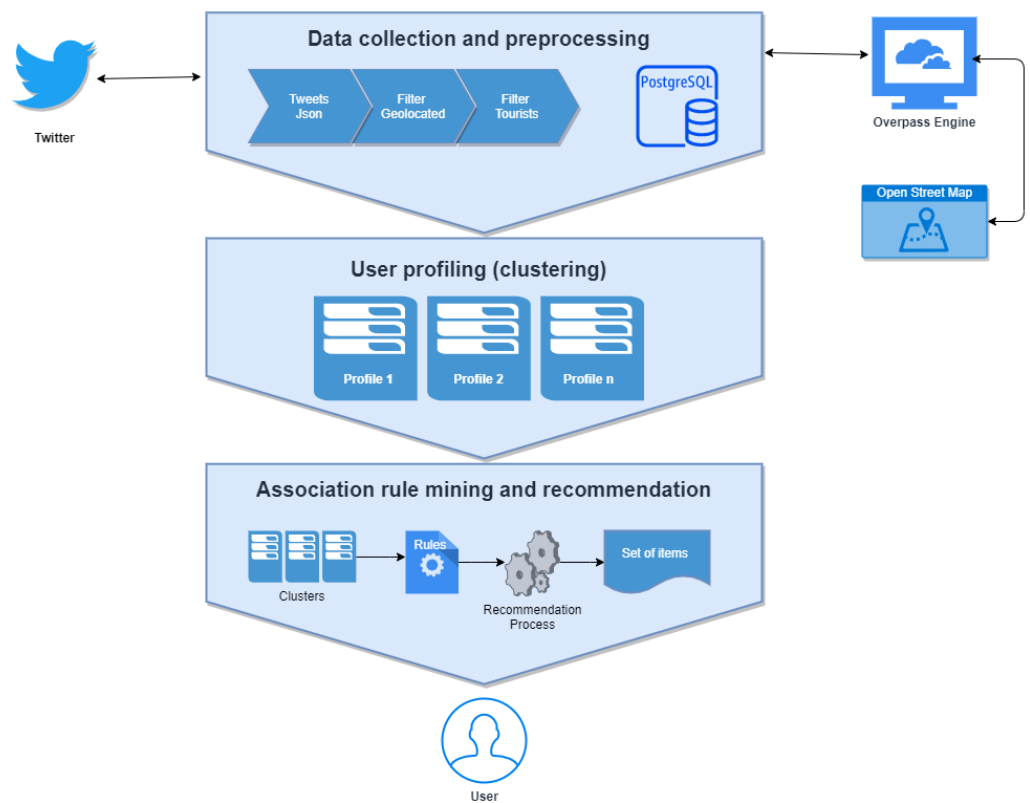


Figure 1. The architecture of the recommender system.

3.2. Data Collection and Pre-Processing

3.2.1. About Twitter

Twitter is an LBSN and a microblogging platform on which users post short pieces of text called tweets, which may contain URLs and references to other users. It was created by Jack Dorsey, Noah Glass, Biz Stone and Evan Williams in March 2006 and launched in July 2006 [40]. Twitter boasts over 300 million users who send approximately 500 million tweets per day [41], making it the fourth largest web site in the world as of January 2021 [42]. It is a popular and widely available medium for users to share their tourism experiences.

We utilised the freely available Twitter Application Programming Interface (API) [43] to download tweets posted in Catalonia. Tweets are returned as JavaScript Object Notation (JSON) files with different kinds of information. Table 1 shows the attributes of a tweet relevant to this work. A full list of tweet attributes can be found at [44].

Table 1. Tweet attributes in Twitter data.

Attribute	Data Type	Short Description
Created at	String	UTC time when the tweet was created.
Id str	String	Unique identifier of the tweet.
Text	String	Actual UTF-8 text of the tweet.
User	User object	Data dictionary containing information about a user including id, screen_name, geo_enable, etc.
Coordinates	Coordinates	Geographical location of the tweet, if shared by the user.
Place	Place object	Geographical data dictionary that indicates the place from which the tweet was sent. It can be a country, a region or even a POI.
Lang	String	BCP 47 language identifier of the machine-detected language of the text of the tweet.

Twitter was chosen as our data source because of its popularity and availability for research. It has been widely used in Tourism fields, such as destination brand communication analysis [45,46], sentiment analysis [18,19] and detection of trajectories. It also includes shared links from posts on other LBSNs, such as Instagram, which might be relevant in more complex multi-platform and multi-modal (text, images, videos) analysis.

3.2.2. Identification of Tweets from Tourists

As one of the aims of this work was to build profiles of tourists and study how they move in a particular city, it was of paramount importance to distinguish the tweets sent by visitors from the tweets sent by local citizens. Twitter does not provide any specific information in this regard.

To identify the tweets of the tourists, we extended the solution proposed in [15] to consider *locations frequently visited* by users. We define a tweet as touristic if it was sent from a location other than the user's frequently visited locations or his/her profile location (which the user may indicate by free text in his/her public profile). The frequently visited locations are defined as those cities from which the user posted tweets for at least 20 days, at least one tweet per day.

Algorithm 1 details the steps performed to determine if a tweet was sent by a tourist. The frequent locations of all users are first determined by counting the number of days a user has tweeted from each place. After that, the coordinates of each geolocated tweet are checked against the set of home locations. If they do not fall within any home location, then the tweets are considered *touristic*. We classify the tweets rather than the users, because a user might be posting as a tourist in one tweet and as a local citizen in another.

Algorithm 1 Identification of tourists' tweets (pseudocode)

```

1: Let  $T$  be the set of all geolocated tweets;
2: Let  $U : L$  be the set of all users and their profile locations  $U : L = \{u_1 : l_1, \dots, u_i : l_i\}$ ;
3: Let  $\alpha = 20$  be the threshold of days tweeting from a location;
4: for users in  $U$  do
5:   Initialize an empty set of home locations for user  $u_i (H_{u_i})$ 
6:   Add user's profile location  $l_i$  to  $H_{u_i}$ , if available
7:   Get tweets  $t_u$  of user  $u_i$  from  $T$ 
8:   Group tweets  $t_u$  by their tweet location
9:   if the count of days with tweets in a tweet location exceeds  $\alpha$  then
10:     Add tweet location to  $H_{u_i}$ 
11:   end if
12:   for tweet in  $t_u$  do
13:     if tweet location in  $H_{u_i}$  then
14:       Set tweet as local
15:     else
16:       Set tweet as touristic
17:     end if
18:   end for
19: end for

```

3.2.3. Activity Identification

In order to capture the preferences of users, it is necessary to analyse their tweets to determine the POIs and types of activities that they have taken part in. The term *point of interest (POI)* has appeared in many tourism-related articles, and was used as early as the 1930s. It usually refers to a place a tourist is expected to see or visit, also known as a "sight" [47,48]. These POIs may be defined by verified administrative bodies or from user interactions and trajectories. Waga et al. considered all these sources, including user interactions by incorporating POIs determined from photos taken by users [49]. It is also possible to predict POIs based on a user's trajectory and proximity, as proposed by Mariescu-Istodor et al. [50]. Twitter does not provide information on the POIs visited

by users, so it has to be inferred from the available data. In this work the POIs and types of activities were determined from Open Street Map and represented in a hierarchical structure. This approach is an extension of the method proposed by Bustamante et al. ([25]), but in this work we have considered more activity categories and we have added text analysis to the identification process.

The following services have been used in the identification process:

- *Open Street Map (OSM)* [51]: It is an open-source map server which includes cartographic documentation of roads, streets, water bodies, buildings, etc. It also provides geocoding and geoparsing services. As it is open source, it has become the first choice for academic research.

In the OSM database, physical features (buildings, roads, etc.) are represented by tags, which describe the geographic attributes of those features [52]. These tags provide information about an element, such as its “name” and “purpose”. For example, the tag *nature : beach* is used to identify a beach. We have used these tags to create a tree structure to categorise activities experienced by tourists. A fully comprehensive list of tags and their descriptions can be found at OSM taginfo [52,53]. It is important to note that tags may change overtime or be discontinued and replaced by OSM.

The activity tree has a root node named “Activities”. Its children are the main categories, which were inspired by an ontology we developed in previous work ([54]): Routes, Sports, Gastronomy, Leisure, Accommodation, Transportation, Nature and Culture. The tree also includes numerous subcategories that are descendants of the main categories. The leaves of the tree correspond to the OSM tags.

Figure 2 shows a sample section of the activity tree. The complete tree consists of 32 subcategories and a total of 175 OSM tags in the leaves. The complete tree is detailed in Appendix A.

```

|-- Gastronomy
|  |-- Food
|     |-- amenity_bbq
|     |-- amenity_biergarten
|     |-- amenity_cafe
|     |-- amenity_restaurant
|     |-- amenity_food_court
|     |-- amenity_fast_food
|     |-- amenity_ice_cream
|     +-- craft_bakery
+-- Enotourism
   |-- craft_winery
   |-- shop_brewing_supplies
   |-- landuse_vineyard
   |-- landuse_orchard
   +-- craft_brewery

```

Figure 2. Sample chunk of the Activity tree, showing the main category “Gastronomy”, its subcategories “Food” and “Enotourism” and the OSM tags associated with them.

- *Overpass turbo* [55]: It is a query server for requesting specific features in the OSM database. Overpass provides a query language (Overpass QL [56]), similar to Structured Query Language (SQL), to help users gain access to specific information in the OSM database. For example, physical features within a certain radius from a [Latitude, Longitude] coordinate pair can be requested and filtered by their OSM tags. We used the Overpass query language to request all POIs within a certain range around the coordinates of a touristic tweet. These POIs were OSM map features categorised as *Nodes*, *Ways*, *Relations* or *Areas*. *Nodes* are single structures, such as office buildings, which include coordinates to represent their locations. *Ways* consist of several nodes with individual coordinates, which represent structures such as roads, highways,

streets, pathways, plaza, fountains, parks and steps. *Relations* are compound structures which include several nodes and ways. For example, complex attractions comprising multiple buildings such as Sagrada Familia are relations. Finally, *Areas* are large physical features that are represented by bounding boxes. Areas contain several nodes, ways and relations. For example, the Port Aventura theme park in Spain is categorised as an area, because it contains several attractions over a large area.

This Overpass query requests all named nodes, ways and relations within a 50 m radius from the coordinates of a tweet, also including the areas if they are not cities, countries, towns or time-zone boundaries.

Figure 3 shows the code snippet written in Overpass QL. Line 1 [out:json] sets the query output as JSON and [timeout:1000] sets the wait time in seconds before the query is terminated. Line 2 'nwr' requests all nodes, ways and relations 'around' <Lat>,<Lon> within the radius of <displacement> meters, and [~"^\name(:.*)?\$"~"."] filters out unnamed map features. Line 3 requests all areas bordering the <Lat>,<Lon>. Lines 4 and 5 filter out all areas that are cities, towns, countries and time-zone boundaries. Finally, line 6 formats the results, 'out geom' gets the full geometry of results, 'tags' includes IDs and tags of the results and 'qt' sorts the results by their geometry.

```

1: [out:json] [timeout:1000];
2: (nwr(around:<displacement>,<Lat>,<Lon>)[~"^\name(:.*)?$"~"."]);
3: is_in(%s,%s);)->.a;
4: ((nwr.a; - nwr.a[type=boundary]); - area.a[place=town]););
5: ((area.a; - area.a[type=boundary]); - area.a[place=town]);););
6: out geom tags (<bounding-box>) qt;

```

Figure 3. Code snippet of an Overpass query.

- *NLTK* [57]: Natural Language Toolkit (NLTK) is an open source toolkit for Natural Language Processing written in Python. It is widely used because it includes a large number of tools for text analysis and it is very well documented. The NLTK library is used to pre-process the text of each tweet. The text is first stripped of stop words based on its language, and then separated into tokens with the NLTK tweet tokenizer. URLs and links of any form are removed, and hashtags composed of several capitalised words are split (e.g., #SagradaFamilia). Finally, numbers, icons, accents, punctuation, user mentions (i.e., user tags beginning with "@") and excess letters in words, such as "funnnn", are also eliminated. Once tweets have been processed, they can be compared with the names of the POIs returned from Overpass to find matches.

The NLTK evergram tool is used to make n-grams of the POI names returned from Overpass. In this way it is possible to detect hashtags that contain POI names which could not be split in the pre-processing step.

To associate an activity with each geolocated tweet in the dataset, POIs are requested from Overpass as described. After the resulting POIs are returned, we adopt a priority-based method, as suggested by Bustamante et al. ([25]), to determine the categories to be assigned to the tweet.

Table 2 is used to assign a category when there are conflicting OSM tags in the proximity of the tweet. We established the priority according to the importance of a category to a tourist, 1 being the highest priority and 8 the least. Additionally, the distance shown in the table is the maximum range in meters at which the priority is relevant. Thus, if a tweet matches two or more POIs with OSM tags from different categories, the category with the highest priority is chosen if the tweet is within the stipulated distance of that POI.

The activity identification steps are detailed in Algorithm 2:

- **Step 1:** The Overpass server is queried to return POIs within a 50 m radius of each tweet in the dataset.

- **Step 2:** Names of POIs returned from Overpass are analysed to find matches with the tweet text or the place name provided by Twitter if it is a POI. The tokens from the tweet are compared with the POI names returned by Overpass, and also with the n-grams made from the POI names that are between 2 and 5 words.

Table 2. Category priority table.

Category	Distance	Priority
Culture	50 m	1
Leisure	25 m	2
Accommodation	35 m	3
Gastronomy	25 m	4
Nature	15 m	5
Routes	15 m	6
Sports	15 m	7
Transportation	15 m	8

Algorithm 2 Activity identification (pseudocode)

Input: Geolocated tweets with exact coordinates

Output: category of each tweet

```

1: for all tweets do
2:   Get set of POIs within 50 m radius of tweet ( $P$ )
3:   Preprocess text in tweet or in POI name when provided by Twitter ( $t\_tokens$ )
4:   if  $P$  is empty then
5:     Return CategoryNull
6:   end if
7:   for POIs in  $P$  do
8:     Tokenize and remove stop words from POI name ( $p\_tokens$ )
9:     Find intersection between  $t\_tokens$  and  $p\_tokens$ 
10:    if size of intersection greater than two then
11:      Add POI to matched list. ( $M$ )
12:    end if
13:    if size of matching list  $M$  is equal to one then /*Case 1 : One match*/
14:      Find category/ies of POI tags using the activity tree
15:      if multiple categories found then
16:        Return category with highest priority
17:      end if
18:    end if
19:    if size of matching list  $M$  is greater than one then /*Case 2 : Multiple matches*/
20:      for all matching POIs do
21:        Find category/ies of POI tags using the activity tree
22:        if multiple categories found then
23:          Add category with highest priority to list  $C$ 
24:        end if
25:      end for
26:      if categories in  $C$  conflict then
27:        Return category with highest priority
28:      end if
29:    end if
30:    if matching list  $M$  is empty then /*Case 3 : No match*/
31:      Repeat steps in Case2 with all POIs returned from Overpass.
32:    end if
33:  end for
34: end for

```

The following cases occur as a result of the text analysis:

1. *One Match*: When only one POI matches the text, the tags of that POI are checked against the activity tree, and the best suited category based on Table 2 is assigned to that tweet.
2. *Multiple Matches*: When more than one POI matches the text, the tags of all matching POIs are checked against the activity tree, and the best suited category based on Table 2 is assigned to that tweet.
3. *No Match*: When no match is found in the text, the tags of the returned POIs are checked against the activity tree, and the category with the highest priority rank based on Table 2 is assigned to the tweet.
4. *No POI*: If the text analysis did not return any POI, the tweet is not assigned to any category.

3.2.4. Data Summary

In this work we streamed posts published in the city of Barcelona in 2019, totalling 1,523,801 tweets from 108,515 users. Before the analysis, we removed the following information:

- Tweets from Barcelonian citizens, not from tourists.
- Tweets that could not be assigned to any category in the activity identification process.
- Users with less than three tweets and their tweets.

Table 3 shows the summary of the data set before and after this filtering process.

Table 3. Data set summary.

Statistics	Value
Total number of tweets in Barcelona	1,523,801
Total number of users in Barcelona	108,515
Statistics after filtering	
Total number of tweets in Barcelona	37,302
Total number of users in Barcelona	6066

3.3. Cluster Analysis

The next stage after data collection and pre-processing was cluster analysis. The goal of this stage was to identify groups of users that had a similar travel behaviour (they enjoyed the same kinds of activities, they had similar mobility patterns, they visited POIs at the same times of day, they enjoyed (or not) visiting popular places, etc.). This knowledge permitted the system to recommend to a user POIs that were visited by similar users. A clustering procedure identifies the users that have a high similarity in the values of a set of features. First, we describe the features that were chosen to represent different aspects of the travel behaviour of tourists; after that, the clustering process is described.

3.3.1. Clustering Features

In this work each user is represented by four kinds of features that represent the preferences of the user with regard to cultural and leisure activities, travel characteristics (length of stay and degree of mobility within the city), degrees of interest in popular POIs and period of the day with more touristic activity.

- *Activity interest features*. These features embed the users' interests in different kinds of touristic activities. They represent different levels of abstraction in the activity tree, as the analysis would be too general if we only considered the eight main categories of the first level. The activities associated with higher percentages of users in Barcelona were selected for the clustering process. All these features were scored as the percentages

of tweets by users that were related to the particular types of activity. The selected features were the following:

1. **Top-tier features.** These features represent some of the main categories in the activity tree. They are %Routes, %Sports, %Accommodation, %Transportation and %Nature.
2. **Middle-tier features.** These are activity features selected from the subcategories of the activity tree. They are %Food, %Enotourism, %AmusementParks, %RecreationFacilities, %Beach, %Health&Care, %NightLife, %Shopping, %Viewpoint, %CulturalAmenities, %Historic and %Religious.
3. **Bottom-tier features.** These activity features are OSM tags represented as leaves of the activity tree. They are %tourism_museum, %amenity_arts_centre and %tourism_gallery.
4. **Other features related to the activity tree.** In the analysis it was found out that the OSM tag {tourism, artwork} was quite popular in our data set, but we did not know what type of artwork was being experienced. Thus, it was decided to break down this tag into several features that represent the type of artwork, using other tags associated with the POIs. These features are %artwork_type_sculpture, %artwork_type_architecture, %artwork_type_statue and %other_artwork. The last one represents cases of undetermined type or works of art that do not belong to the other three types. Figure 4 shows the 24 activity features and the percentages of users that visited them in Barcelona (according to the content and the location of their tweets). It may be seen that the top categories were RecreationFacilities, Religious, Historic and Food, followed by Museums and Accommodation.

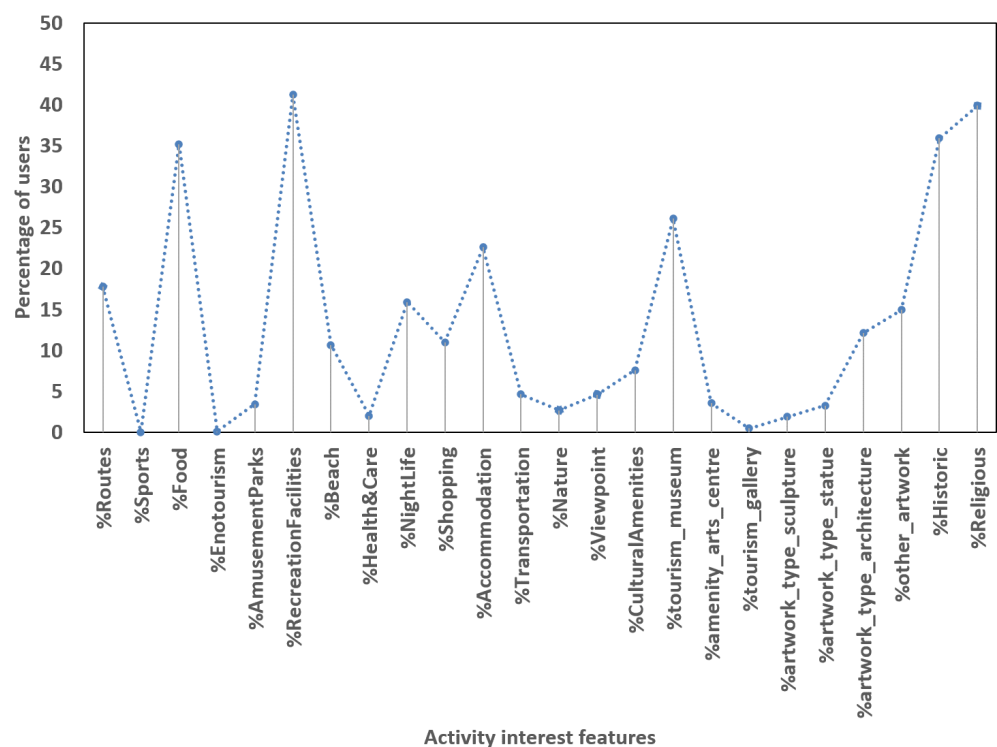


Figure 4. Activity features by percentages of users.

- **Travel features.** These features are related to the travel habits of the user. Concretely, they contain information on the durations of trips to Barcelona and the degrees of mobility within the city. They are the following:
 1. **Length of stay.** It is the maximum number of consecutive days in which the user posted tweets from Barcelona.

2. **Tweet distance maximum and average features.** Twtdistance_max and Twtdistance_avg are the maximum and average distances between the locations of the tweets of the user in Barcelona. They constitute contextual information on the user's ability to explore the city.
- *Popularity features:* These features represent the interest of the user in visiting the most well-known and popular POIs. In order to obtain a popularity order, the POIs were rated according to the numbers of users in the database that had visited them. Popularity is split into five features. The first four (%top10_tweets, %top10–20_tweets, %top20–50_tweets and %top50–100_tweets) are the percentages of tweets of the user from POIs in positions 1–10, 10–20, 20–50 and 50–100 of the ranking. The feature %top100_tweets codifies the percentage of tweets that were not sent from any of the top 100 POIs in the city.
 - *Temporal features:* These features embed the time of the day favoured by the user in his trips. There are 4 features representing the percentages of tweets that the user posted by period of the day. The features are: %Dawn_tweets (00:00–07:00), %Morning_tweets (07:00–12:00), %Afternoon_tweets (12:00–20:00) and %Night_tweets (20:00–00:00).

In summary, the preferences and travel habits of each user are represented by a vector of 35 numerical features (24 for the interest in different kinds of activities, 2 for the travel features, 5 for the interest in popular POIs and 4 to codify the time the tweets were sent out). All of them are percentages (values between 0 and 1), except the two travel features.

3.3.2. Clustering Parameters

The clustering process was done using the scikit-learn [58] Python library. The choice of parameters used in the cluster analysis was the following:

- *Algorithm.* The k-means algorithm was selected because of its speed and ability to work with large data sets.
- *Feature scaling.* In cluster analysis it is necessary to ensure the data are scaled appropriately, as features having different scales would affect the clustering process negatively. The clustering features were standardised using the *Z-Score*.
- *Number of clusters (k).* The k-means algorithm requires the number of clusters as an input parameter. Clustering is by nature an unsupervised analysis process, and therefore, the optimal number of clusters is case-dependent. In this work we were not concerned about having equally sized clusters with clear dividing boundaries, but rather clusters that represent different combinations of the clustering features in order to create user profiles with different interests and contexts. After some experimentation, we found $k = 25$ clusters was a suitable number for our data set.

Figure 5 shows a graphical representation of each cluster with a different colour; features are on the x-axis and the mean value of each is on the y-axis. The clusters display different combinations of peaks across the clustering features, showing high heterogeneity (especially in the activity interest features). On the contrary, temporal features do not exhibit great differences among the clusters. The high heterogeneity in activity features allows the recommendation of different POIs across clusters.

3.4. Association Rule Mining

After clustering, the next stage is *association rule mining* (ARM). ARM is a popular technique that enables the identification of items that co-occur frequently within a specific data set. It has been successfully applied in marketing to study purchases in retail markets, under the name *market basket analysis* (MBA). The idea is that the owner of a supermarket could potentially increase the sales of two products with high affinity by shelving them together. Similarly, in this work the idea is to recommend together POIs with high affinity, which will have been detected in this step. Two POIs will have been visited together on the same day by many tourists if they are close (or well connected by public transport) and if

they match the same set of preferences. The ARM process was implemented with the help of the mlxtend [59] Python library.

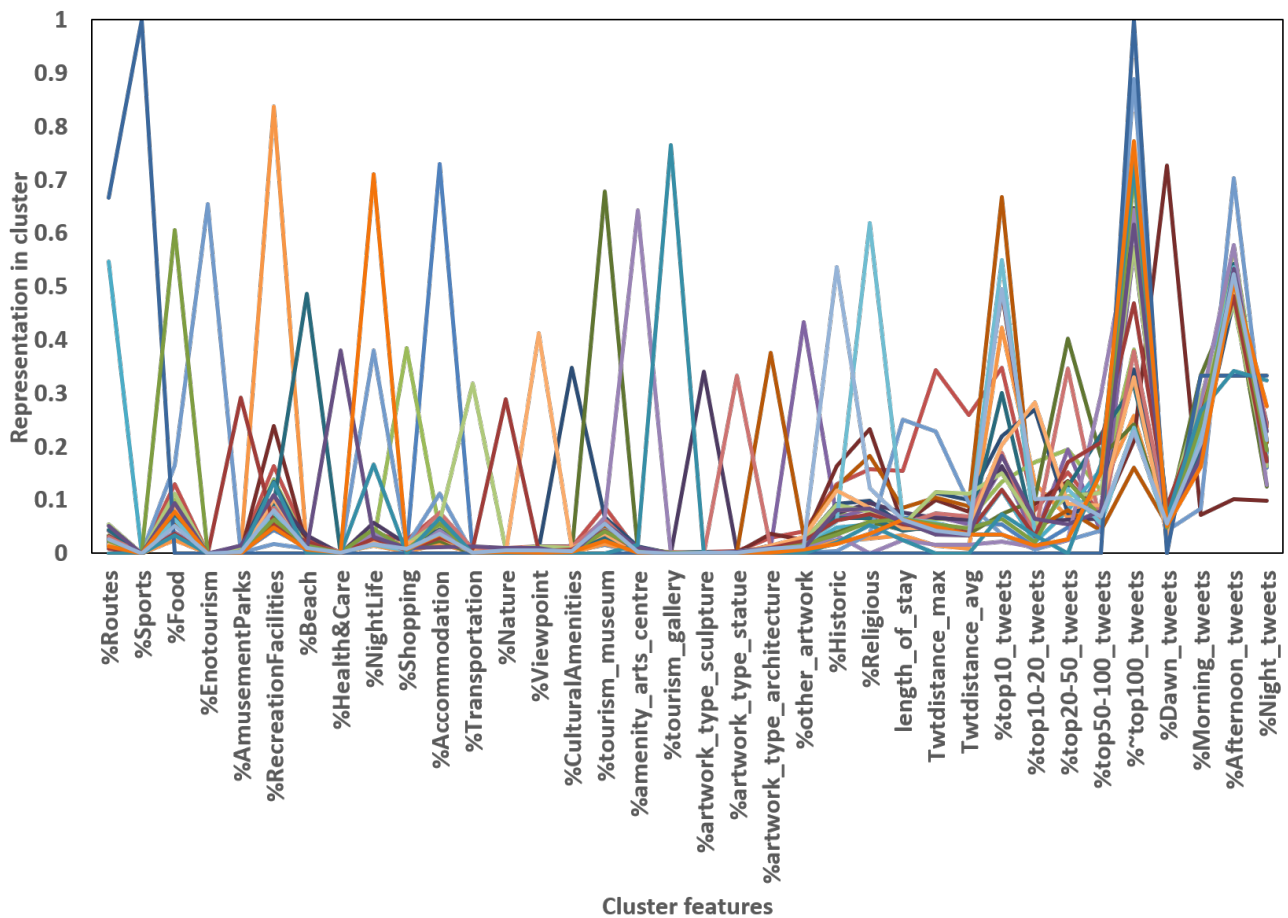


Figure 5. A cluster model showing the distribution of features in the clusters.

3.4.1. ARM Parameters

- *Pre-processing*: In MBA, the analysis is usually performed in shopping sessions; i.e., one user may have multiple baskets from different shopping sessions. In our case, it was decided to split the users into POIs experienced in the same day. This was beneficial because the system aimed to recommend POIs for daily trip planning. However, this decision also led to some loss of information, because we dropped the days in which less than two POIs were experienced.
- *Frequent itemset mining algorithm*: Frequent itemset mining (detecting sets of items that appear frequently together) is the main step in ARM. There are a variety of similar algorithms with which to perform this step. The *Apriori* algorithm [60] was chosen in this work because of its popularity and widespread acceptance. This algorithm requires *minimum support* to be provided, which is the minimum amount of times an itemset has to occur for it to be considered as frequent. This parameter was given a low value because the data set is sparse and it needed some leeway to function. The algorithm also requires the *maximum length* of the itemsets (maximum size of the sets of items appearing together frequently). The values chosen for these parameters are shown in Table 4.
- *Association rule parameters*: In ARM, multiple metrics are computed for each mined rule to evaluate its performance (they are detailed in the following subsection). In order to provide useful rules, these metrics may be used as filters; rules that do not reach a given threshold are discarded. The usual choices are *confidence*, *support* or

lift, but in our case this filtering step was not relevant because the posterior selection algorithm performed a ranking of the rules, as will be shown later. Thus, the filtering parameters were set as shown in Table 4.

Table 4. ARM algorithm parameters.

Apriori Params		Association Rule Params	
Min support	Max length	Metric	Minimum value
0.001	3	Lift	0

3.4.2. ARM Metrics

Several metrics can be used to evaluate the usefulness of mined rules. The three most popular ones are support, confidence and lift. They were used to select the best rules for the recommendation process, as will be described in the next section.

- **Support.** It indicates how frequently an itemset occurs in a data set. It is the fraction of times an itemset appears among all the transactions being analysed. It can be denoted as the probability of occurrence of the itemset $P(itemset)$. The support of a rule is the percentage of times that the antecedent and the consequent of the rule appear together.
- **Confidence.** It indicates how often a rule is found to be true. It is the proportion of times the consequent is found in the same transaction as the antecedent. It can be denoted as the conditional probability of the consequent appearing in the same transaction after the antecedent is found to be true $P(antecedent|consequent)$.
- **Lift.** It was established to solve the problem on the confidence, being dependent only on the support of the antecedent. The order of the consequent and the antecedent in the rule does not matter, which makes the confidence metric a bit skewed because it considers the consequent to be dependent on the antecedent. The lift metric modifies this fact by considering the support of both the antecedent and the consequent.

The mathematical formulation of these measures is the following:

For a rule: $X \rightarrow Y$, where X is the antecedent and Y is the consequent

Support:

$$supp(X) = P(X); supp(Y) = P(Y); supp(X \rightarrow Y) = P(X \cup Y) \quad (1)$$

Confidence:

$$conf(X \rightarrow Y) = P(X|Y) = \frac{supp(X \rightarrow Y)}{supp(X)} \quad (2)$$

Lift:

$$lift(X \rightarrow Y) = \frac{supp(X \rightarrow Y)}{supp(X) * supp(Y)} \quad (3)$$

The three metrics were combined to rank the mined rules in the selection stage, as will be detailed shortly. In summary, the following ARM steps were executed for each individual cluster to mine useful rules:

1. Create separate baskets with the POIs visited in the same day by each user.
2. Mine frequent itemsets of visited POIs with a maximum length of 3 and a minimum support of 0.001.
3. Build association rules from the itemsets uncovered in the previous step.
4. Compute the previous metrics for each rule.

3.5. Personalised Recommendations of Touristic Activities

This is the final stage of the system. First, the recommender ranks the association rules obtained in the previous stage, which are used to decide the POIs to be recommended to a user. The set of recommended POIs should ideally fulfil these conditions:

- It should contain only POIs that have been visited by other members of the same cluster.
- It should fit the user's interests regarding the preferred types of activities and the attraction towards popular items.
- It should reflect the causality of the association rules of the cluster, in order to recommend POIs with high affinity.

To achieve these conditions, the recommendation process was formulated as a ranking problem. The association rules of the user's cluster were ranked based on their performances according to certain evaluation metrics, and POIs were selected while taking into account the associations expressed in those rules. The employed evaluation metrics were antecedent support, consequent support, support, confidence, lift and two new metrics that evaluated the rules in terms of the preferences of the user towards certain kinds of touristic activities and towards popular spots. Those two new metrics are the following:

- *Preference ratio*: This metric evaluates if the POIs that appear in a rule belong to any of the activity categories preferred by the user. The activity categories coincide with the activity features used in the clustering process.

Let CAT_{pref} be a user's preferred categories, i.e., the activity categories for which the user has at least one tweet. $1_{CAT_{pref}}(\psi(POI))$ is the indicator function that has value 1 if $\psi(POI) \in CAT_{pref}$ or 0 otherwise. $\psi(POI)$ is a function that gets the category to which a POI belongs by looking up the activity tree. Furthermore, let $D_{CAT}(POI)$ be a function that signals the user's degree of interest in a preferred category, where the degree of interest is the percentage of the user's visited POIs belonging to the preferred category. If P_{rule} is the set of POIs in a rule, the preference ratio is calculated as follows:

$$preference\ ratio = \frac{1}{|P_{rule}|} \sum_{p \in P_{rule}} (1_{CAT_{pref}}(\psi(p)) * D_{CAT}(p)) \quad (4)$$

- *Popularity ratio*: This is the percentage of popular POIs in a rule. Let P_{top10} be the top 10 POIs extracted from the data set, and $1_{P_{top10}}(POI)$ is the indicator function with value 1 if $POI \in P_{top10}$ and 0 otherwise. The popularity ratio is computed with the following formula:

$$popularity\ ratio = \frac{1}{|P_{rule}|} \sum_{p \in P_{rule}} 1_{P_{top10}}(p) \quad (5)$$

After computing all the metrics as shown in Equations (1) to (5), they were combined to get an overall score for each association rule, using a weighted arithmetic mean (also known as the *weighted average* (WA) aggregation operator in decision support systems). Let A be the set of metrics to be aggregated $A = (a_1, \dots, a_n)$ and $W = (w_1, \dots, w_n)$ be the set of weights for each metric. Then,

$$WA(A) = \sum_{i=1}^n w_i a_i \quad (6)$$

The WA operator allows one to determine the relevance of each metric using weights. In this work the weight of each metric has been manually set, but it depended on the user's degree of interest in popular POIs. Two sets of weights were defined, one for users

interested in popular POIs and the other for users interested in off the beaten track POIs. The weights were linearly combined as follows:

$$W_{comb} = (1 - \beta)W_{unpop} + \beta W_{pop} \tag{7}$$

In this expression, W_{comb} is the set of adapted weights, and W_{unpop} and W_{pop} are the predefined sets of weights for the users interested in unpopular and popular POIs, respectively (shown in Table 5). The parameter β expresses the degree to which the user is interested in popular POIs. It is computed as the fraction of the user’s visited POIs that are in the top 10. Thus, when all POIs visited by the user are in the top 10, β is 1 and the W_{pop} weights are applied. Inversely, when none of the POIs visited by the user appear in the top 10, β is 0 and W_{comb} coincides with W_{unpop} .

Table 5. Weights for different cases of interest.

Case	Preference Ratio	Lift	Confidence	Support	Antecedent Support	Consequent Support	Popularity Ratio
W_{unpop}	0.5	0.15	0.15	0.1	0.05	0.05	0.0
W_{pop}	0.3	0.05	0.05	0.1	0.1	0.1	0.3

In the W_{pop} case, we wanted to give high priority to the rules containing POIs in the top 10 (while still considering the user’s preferences), so we gave higher importance to *popularity_ratio* and the three support metrics because they reflect popularity in the rule and in the cluster respectively. In the case of W_{unpop} , we zeroed out the *popularity_ratio* and relied on the other metrics, especially on the *preference_ratio*.

Given the definition of W_{pop} and W_{unpop} in Table 5, W_{comb} was adapted for each user using Equation (7). These weights were then used in Equation (6) to combine the metrics of all the association rules of the user’s cluster. The steps of the final recommendation process are the following:

1. The user that desires a recommendation is assigned to a cluster. To make this assignment, first the values of the clustering features are extracted from the analysis of the Twitter history of the user (in the future, a survey will be used to gauge the user’s preferences). Then the user is assigned to the closest cluster comparing the Euclidean distance between the user’s features and the mean of the members in each cluster. The Euclidean distance was used because it was the distance metric employed in the previous k-means clustering process.
2. The system takes the association rules of the user’s cluster and their metrics.
3. The popularity and preference metrics are computed for each rule, based on the user’s data.
4. The user’s personalised weights are computed as described in Equation (7).
5. The metrics in Table 5 are combined using the WA operator to give an overall score for each rule.
6. A final selection procedure (see Algorithm 3) is used to select the set of items to be recommended to the user.

Algorithm 3 starts by ranking the association rules by their overall scores, and then the POIs in the highest ranked rule are added to the set of POIs to be recommended R . It then loops through R , adding the POIs in the highest ranked rule for which any POI in R appears in the antecedent. This process is repeated until we have the requested amount of POIs for recommendation or the association rules of the cluster have been exhausted.

Algorithm 3 Selection pseudocode

Input: ARM rules

Output: Set of recommended items R

```

1: Let  $N$  be the number of POIs to recommend
2: Let  $R$  be the empty set of POIs to recommend
3: Sort ARM rules by their score
4: while true do
5:   Select the highest ranked rule not considered in the previous iteration
6:   Add POIs in the rule to  $R$  (if not already in  $R$ )
7:   if ARM rules exhausted then
8:     Stop process and return  $R$ 
9:   end if
10:  for  $POI$  in  $R$  do
11:    Select the highest ranked rule where  $POI$  appears in the antecedent
12:    Repeat the for loop for newly added POIs
13:  end for
14:  if size of  $R = N$  then
15:    Stop process and return  $R$ 
16:  end if
17: end while

```

4. Experiments and Results*4.1. Experimentation Details*

The primary focus of the experiments was to evaluate the effect of the social media-based clustering process on the quality of the recommendations. The system was not compared directly to other tourism recommenders, as there are not any similar approaches combining clustering and association rules. We used the final data set of tweets posted in the city of Barcelona, obtained after the pre-processing and filtering steps detailed in Section 3.2. The data set consisted of 37,302 tweets and 6066 users. It was then partitioned by users—80% users in the training set and 20% in the test set—which is one of common methods used in machine learning studies.

The training set was run through the stages of the proposed system. Firstly, the clustering features were extracted from the tweets of the users of the training set and then they were clustered using the parameters described in Section 3.2.2. Secondly, the association rules for POIs were mined with respect to each cluster, as described in Section 3.4. The clusters for which no rule was discovered in this stage were removed, so they were not considered when assigning users to clusters. Finally, the recommendation process was carried out as described in Section 3.5. The popularity ratio and preference ratio for each association rule with respect to every individual user in the test set were computed. The weighting vector was adapted for each user based on his/her interest in popular POIs. Then, all the rule evaluation metrics were combined using weighted averaging and the final selection (Algorithm 3) was carried out. It was decided to recommend 10 items to each user in the test set. POIs related to Gastronomy, Accommodation and Transportation were not considered, as we decided to focus the recommendations purely on touristic activities (a total of 1363 in the considered data set). These 1363 POIs were only considered for recommendations when clustering was not applied. After the clustering stage, the pool of POIs to select from was, on average, approximately 80 POIs per cluster, with very little overlap between pools. The average pairwise Jaccard similarity coefficient of all cluster pools was 0.129. The next subsection describes the metrics used to evaluate the quality of the recommendations.

4.2. Evaluation Metrics

The metrics were modelled as suggested by Massimo and Ricci ([21]). Let U be the set of users in the test set, R the set of POIs recommended to the users in U , R_u the set of POIs

recommended to a particular user u , V the set of POIs visited by users in the test set, V_u the set of POIs visited by a specific user u , and P the set of all possible recommendable POIs.

- **Average Precision (AP):** It is the ratio of *correct* POI recommendations made to the users in the test set. A correct recommendation was determined by the user’s degree of preference in the category of the POI. AP is formulated as follows:

$$AP = \frac{1}{|U||R_u|} \sum_{u \in U} \sum_{r \in R_u} 1_{CAT_{pref}}(\psi(r)) \tag{8}$$

The function $\psi(\cdot)$ gets the category of a POI. $1_{CAT_{pref}}$ is an indicator function that has value 1 if $\psi(\cdot)$ returns a category that is preferred by the user and 0 otherwise. A category is preferred by a user if at least one of the user’s tweets has been associated with it.

- **Average Category Recall (ACR):** It is the ratio of *preferred* recommendable POIs that are actually recommended. ACR is formulated as:

$$ACR = \frac{1}{|U|} \sum_{u \in U} \frac{|R_u \cap PREF_u|}{|PREF_u|} \tag{9}$$

where $PREF_u$ is the set of POIs preferred by a concrete user u in the test set. Preference is determined by the user’s interest in the activity category to which the POI belongs. A category is preferred if at least one of the user’s tweets has been associated to it.

- **Average Item Recall (AIR):** It is the ratio of visited POIs that are actually recommended. AIR is formulated as:

$$AIR = \frac{1}{|U|} \sum_{u \in U} \frac{|R_u \cap V_u|}{|V_u|} \tag{10}$$

- **Unified Item Recall (UIR):** It is the unified fraction of times in which the POIs recommended to a user were among the POIs actually visited by the user. The UIR is formulated as:

$$UIR = \frac{|\bigcup_{u \in U} R_u \cap V_u|}{|\bigcup_{u \in U} V_u|} \tag{11}$$

- **Similarity:** It measures how similar the POIs recommended to a user and the POIs visited by the user are in terms of their paths in the activity tree. It is formulated as:

$$similarity = \frac{1}{|U||R||V|} \sum_{u \in U} \sum_{r \in R_u} \sum_{v \in V_u} sim(\rho(r), \rho(v)). \tag{12}$$

$sim(\cdot, \cdot)$ is the Jaccard similarity coefficient that estimates the similarity of two sample sets, and $\rho(\cdot)$ produces a set representing the path of the POI in the activity tree.

- **Coverage:** It measures the width of the recommendations. It is the percentage of items recommended to the users in the test set with respect to the total set of recommendable items. It is formulated as:

$$coverage = \frac{|\bigcup_{u \in U} R_u|}{|P|} \tag{13}$$

- **Popularity:** It measures the degree to which the popularity of recommended POIs matches the user’s popularity preference. $VPOP_u$ is the fraction of top 10 POIs visited by the user u , and $RPOP_u$ is the fraction of top 10 POIs recommended to the user.

$$VPOP_u = \frac{|P_{top10} \cap V_u|}{|V_u|} \tag{14}$$

$$RPOP_u = \frac{|P_{top10} \cap R_u|}{|R_u|} \tag{15}$$

Then, the popularity measure is computed as follows:

$$popularity = \frac{1}{|U|} \sum_{u \in U} 1 - |VPOP_u - RPOP_u| \quad (16)$$

- **Personalisation:** It measures the mean dissimilarity of users in the test set, based on their recommended POIs. It is formulated as:

$$personalisation = 1 - \left(\frac{1}{C_2^{|U|}} \sum_{u,w \in U} \text{cossim}(\gamma(u), \gamma(w)) \right) \quad (17)$$

$C_2^{|U|}$ is the set of all pairs of different users without repetitions, $\text{cossim}(\cdot, \cdot)$ is the cosine similarity of two vectors and $\gamma(\cdot)$ generates one hot vector representing the POIs recommended to a user from all recommendable POIs.

- **Diversity:** It is the pairwise dissimilarity of POIs recommended to users in the test set, modelled after the diversity measure used in [61]. Borràs et al. replaced the arithmetic mean for an *ordered weighted average* (OWA) aggregation operator to avoid situations in which high values compensate for low values (e.g., aggregating (0,0,0,1,1,1) and (0.5,0.5,0.5,0.5,0.5,0.5) with arithmetic mean will have the same result, 0.5, but they are very different situations). The OWA weights are defined using a regular increasing monotone linguistic quantifier, which invokes a disjunctive policy where the lowest similarity values have higher weights. If $A = (a_1, a_2, a_3, \dots, a_n)$ is the set of values to be aggregated (in decreasing order) and $W = (w_1, w_2, w_3, \dots, w_n)$ is the weighting vector, where $\sum_{j=1}^n w_j = 1$, then OWA is defined as:

$$OWA_w(A) = \sum_{j=1}^n w_j a_j \quad (18)$$

where W is a regular increasing monotone quantifier defined as:

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right), \text{ and } Q(x) = x^2 \quad (19)$$

Diversity is then formulated as follows:

$$diversity = \frac{1}{|U|} \sum_{u \in U} \left(1 - OWA_w \left(\bigcup_{r,t \in R_u} \text{sim}(\rho(r), \rho(t)) \right) \right) \quad (20)$$

In that expression, $\text{sim}(\cdot, \cdot)$ and $\rho(\cdot)$ are the same as in the similarity measure.

4.3. Experiment Results and Discussion

We performed three identical experiments to compare the results of the recommender with and without the clustering stage (when the clustering stage was not used, all users were assumed to belong to a single unique cluster). In each experiment, the data set was randomly split into a training set (80% of 6066 users: 4853) used in the clustering and association rule mining stages, and a test set (20% of 6066 users: 1213) used to make recommendations which were evaluated with the metrics in Section 4.2. Figure 6 shows the performances of the recommender in the two cases. In almost all metrics, the use of clustering improved the performance of the system (noticeably better in some way, such as coverage and personalisation). We determined the noticeable performances by the differences between the scores in the two cases. If $diff \geq 0.05$, it was noticeably better; if $0.02 \leq diff < 0.05$ it was slightly better; and $diff < 0.02$ shows a negligible difference.

The actual performance scores in the experiments are detailed in Table 6. Noteworthy *difference* values are in bold and negligible differences are underlined. The red values show the cases in which the performance decreases when using clustering. In *UIR*, *coverage*

and *personalisation*, the use of clustering improves the results noticeably. These metrics benefit from the clustering process because they depend on the POI pool considered for recommendation. *Unified item recall* and *coverage* increased because users were placed in clusters with pools that match their preferences, so the recommended POIs were likely to have been visited. *Personalisation* increased because the pools of different clusters were quite different, as explained in Section 4.1, so users across clusters received different recommendations, causing more uniqueness. The use of clustering provoked a slight decrease in *diversity*, due to data sparsity. *Diversity* scored the set of recommended items based on their dissimilarity, which was affected by the *preference ratio* in Section 3.5. In the experiments we used the categories visited by the test users as their preferred activity categories, and therefore, diversity dropped when there were clusters and the set of visited activity categories was small. This could be solved by incorporating mechanisms to enhance diversity in the system [61]. A similar drop was also seen in *average item recall*, but in this case it was due to the streamlining of the POI pools for recommendation during the ARM process.

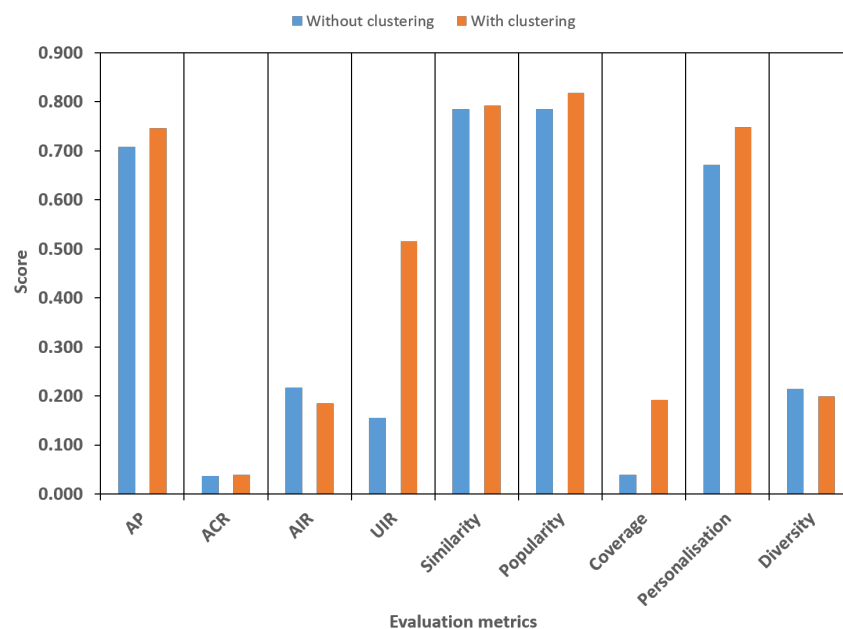


Figure 6. A bar plot of the average performance scores of the recommender system across the three experiments. Cases with and without clustering are shown in orange and blue, respectively.

Average category recall and *similarity* were not very affected by the use of clustering; they only improved to minor extents. These metrics are largely dependent on data sparsity and the number of recommendable POIs, as detailed in Section 4.1; thus, unlike other metrics, they are indifferent to clustering. Finally, *precision* and *popularity* saw just noticeable increases when clustering was applied.

A deeper analysis of the results is presented in Figure 7, which shows box plots of *precision*, *category recall*, *item recall*, *similarity*, *popularity* and *diversity* scores for each user in the test set. In experiments 2 and 3, the use of clustering was able to raise the upper quartile of the precision plot to 1, so 25% of the users in the test set were given recommendations with near 100% accuracy.

However, the most important differences the plots are present for the popularity metric. In the case with clustering, although the increase in popularity in Table 6 is only slightly noticeable, the box plots show that in all the experiments the majority of the test set scored better than without clustering. The outliers shown in the box plots were responsible for the lower overall difference value.

Table 6. Results of evaluation when with clustering and without.

Experiment 1									
Case	AP	ACR	AIR	UIR	Similarity	Popularity	Coverage	Personalisation	Diversity
Without clustering	0.707	0.036	0.223	0.161	0.785	0.779	0.039	0.654	0.220
With clustering	0.735	0.039	0.197	0.484	0.791	0.826	0.172	0.733	0.208
Experiment 2									
Without clustering	0.712	0.038	0.223	0.164	0.784	0.780	0.041	0.676	0.210
With clustering	0.762	0.040	0.181	0.543	0.794	0.806	0.206	0.753	0.191
Experiment 3									
Without clustering	0.705	0.035	0.206	0.142	0.788	0.794	0.038	0.684	0.213
With clustering	0.742	0.039	0.178	0.519	0.792	0.822	0.199	0.758	0.199
Experiment average									
Without clustering	0.708	0.036	0.217	0.156	0.786	0.785	0.039	0.671	0.214
With clustering	0.746	0.039	0.185	0.516	0.792	0.818	0.192	0.748	0.199
Difference (diff)	0.038	0.003	-0.032	0.360	0.006	0.034	0.153	0.077	-0.015

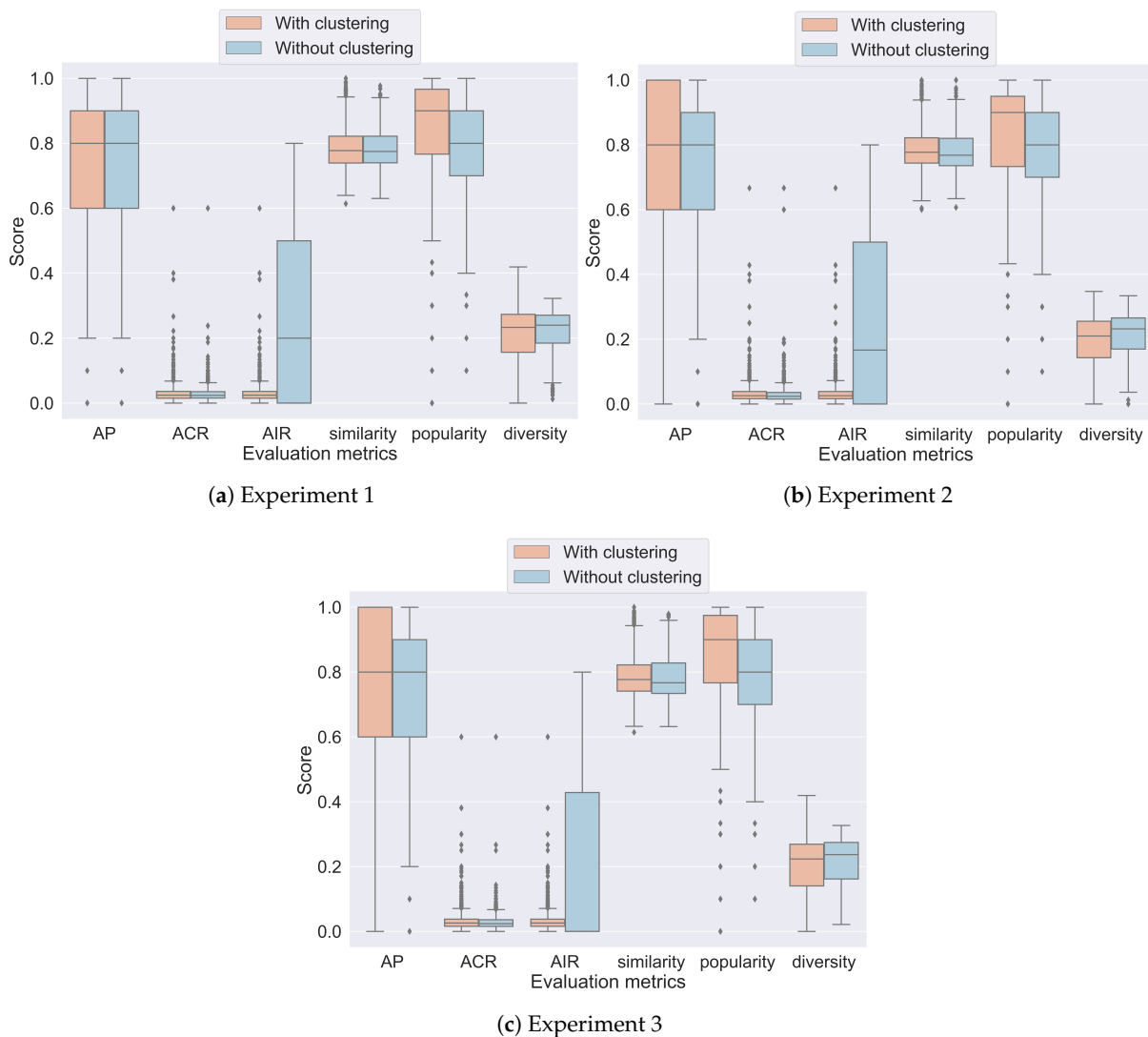


Figure 7. Box plots of precision, recall, similarity, popularity and diversity metrics for the test set in each experiment.

Concerning category recall, the box plots show that, although the concentration of the test set did not result in any differences in both cases, there were more outliers with good scores when clustering was used. In the case of item recall, the performance increased without clustering because a larger pool of POIs was considered, thereby creating the chance to have more POIs recommended that could have been visited by users. The diversity measure was the only one with better scores for all quartiles in all experiments when clustering was not used.

To summarise, we evaluated the performance of the proposed system to quantify the benefit of a social media-based clustering step in the recommendation process. When clustering was applied, the performance scores increased across all metrics except diversity and item recall. The clustering process helped to refine the pools of POIs considered for recommendations.

5. Conclusions and Future Work

In this paper, we have proposed a cluster-based user profiling technique combined with association rule mining to recommend points of interest to tourists. This technique focuses on user features extracted from social media that represent interests, habits and context. The intention of the clustering step is not to obtain any predefined number of classes of users, but rather to learn abstract generalisations implicit in the data set.

The resulting user profiles are then filtered using association rule mining to recommend touristic activities with high affinity. We evaluated the approach with geolocated tweets from Twitter, posted in the city of Barcelona in 2019, and performed a comparative analysis of the results with and without the clustering technique. We conclude that the clustering approach improves the system's accuracy and its ability to encapsulate a user's interests, by refining the pool of recommendable items to a user with the help of the association rules of his/her associated cluster.

In the recommendation process, the initial data collection and pre-processing steps were the most difficult ones, due to gaps in the Twitter data, especially in cases where tweets were not geolocated. This created a partial picture when analysing user preferences. It was also challenging to find a suitable number of clusters in the user profiling step. In addition, the unexpected negative effect of COVID-19 restrictions on tourism left 2019 as the only viable year for the analysis. Finally, we were unable to recommend diverse items without explicitly defining diversity mechanisms in the system.

In the future, we plan to incorporate more user attributes and context into the clustering features. We will consider categorical features and experiment with k-medoids as the clustering algorithm. We also plan to add an extra step, wherein we order the recommended touristic activities to improve tourist satisfaction by viewing this as an orienteering problem [62]. Finally, we will incorporate the system into a real-world application to enact more testing and fine tuning.

Supplementary Materials: Python code for all algorithms and the tweet IDs for the dataset are available at <https://www.mdpi.com/article/10.3390/app11146512/s1>.

Author Contributions: Conceptualisation, J.A.O., J.B. and A.M.; methodology, J.A.O.; software, J.A.O.; validation, J.B. and A.M.; formal analysis, J.A.O.; investigation, J.A.O.; resources, J.B. and A.M.; data curation, J.A.O. and J.B.; writing—original draft preparation, J.A.O.; writing—review and editing, J.B. and A.M.; visualization, J.A.O.; supervision, J.B. and A.M.; project administration, J.B. and A.M.; funding acquisition, J.B. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Third Party Data Restrictions apply to the availability of these data. Data were obtained from Twitter and are available at <https://twitter.com> (accessed on 1 January 2019) with the permission of Twitter. Researches are allowed to provide Tweet IDs to other researchers to download using Twitter’s API. The Tweet IDs of data presented in this paper are included in the Supplementary Material.

Acknowledgments: Jonathan Ayebakuro Orama is a fellow of Eurecat’s “Vicente López” PhD grant program.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RS	Recommender System
UNWTO	United National World Tourism Organization
GPS	Global Positioning System
POI	Point of Interest
OSM	Open Street Map
LBSN	Location-Based Social Networks
URL	Uniform Resource Locator
JSON	JavaScript Object Notation
UTC	Coordinated Universal Time
UTF	Unicode Transformation Format
BCP	Best Current Practices
SQL	Structure Query Language
QL	Query Language
NLTK	Natural Language Toolkit
ARM	Association Rule Mining
MBA	Market Basket Analysis
OWA	Ordered Weighted Average
WA	Weighted Average

Appendix A

The complete activity tree is presented below. Main categories are in bold; subcategories are italicised.

Activities

- |– **Routes**
 - | |– *SportsRoutes*
 - | | |– route_bicycle
 - | | |– route_canoe
 - | | |– route_hiking
 - | | |– route_running
 - | | |– route_ski
 - | | |– route_horse
 - | | |– route_mtb
 - | | +- route_piste
 - | |– *RelaxationRoutes*
 - | | +- route_foot
 - | |– *NatureRoutes*
 - | | +- *MountainRoutes*
 - | | |– climbing_route
 - | | +- climbing_route_bottom
 - | |– *TownRoutes*
 - | | |– highway_pedestrian
 - | | +- highway_footway
 - | +- *CultureRoutes*

- | |– route_historic
- | |– historic_path
- | +- historic_way
- |– **Sports**
- | |– *AquaticSports*
- | | |– sport_water_polo
- | | |– sport_canoe
- | | |– sport_scuba_diving
- | | |– sport_kitesurfing
- | | |– sport_surfing
- | | |– sport_water_ski
- | | |– sport_cliff_diving
- | | |– sport_sailing
- | | +- sport_rowing
- | |– *AirSports*
- | | |– sport_parachuting
- | | |– sport_paragliding
- | | +- sport_free_flying
- | |– *Climbing*
- | | |– sport_climbing
- | | +- sport_climbing_adventure
- | |– *MotorSports*
- | | +- sport_karting
- | |– *ShootingSports*
- | | |– sport_shooting
- | | |– sport_archery
- | | +- sport_paintball
- | +- *OtherSports*
- | |– sport_bullfighting
- | |– sport_cycling
- | |– sport_golf
- | |– sport_9pin
- | |– sport_10pin
- | |– sport_ice_skating
- | |– sport_fishing
- | +- sport_skiing
- |– **Gastronomy**
- | |– *Food*
- | | |– amenity_bbq
- | | |– amenity_biergarten
- | | |– amenity_cafe
- | | |– amenity_restaurant
- | | |– amenity_food_court
- | | |– amenity_fast_food
- | | |– amenity_ice_cream
- | | +- craft_bakery
- | +- *Enotourism*
- | |– craft_winery
- | |– shop_brewing_supplies
- | |– landuse_vineyard
- | |– landuse_orchard
- | +- craft_brewery
- |– **Leisure**
- | |– *Parks&Recreation*

- | | | - *AmusementParks*
- | | | | - tourism_zoo
- | | | | - tourism_theme_park
- | | | | - leisure_water_park
- | | | +- tourism_aquarium
- | | +- *RecreationFacilities*
- | | | - leisure_sports_centre
- | | | - amenity_cinema
- | | | - amenity_theatre
- | | | - leisure_stadium
- | | | - leisure_playground
- | | | - amenity_casino
- | | | - amenity_gambling
- | | | - building_stadium
- | | | - leisure_pitch
- | | | - leisure_amusement_arcade
- | | | - leisure_miniature_golf
- | | | - leisure_swimming_pool
- | | | - leisure_swimming_area
- | | | - leisure_ice_rink
- | | | - leisure_golf_course
- | | | - leisure_disk_golf_course
- | | | - leisure_bowling_alley
- | | | - leisure_horse_riding
- | | | - leisure_fishing
- | | | - leisure_garden
- | | | - leisure_park
- | | +- tourism_picnic_site
- | | - *Beach*
- | | +- natural_beach
- | | - *Health&Care*
- | | | - leisure_sauna
- | | | - shop_beauty
- | | | - shop_cosmetics
- | | +- shop_massage
- | | - *NightLife*
- | | | - amenity_nightclub
- | | | - amenity_pub
- | | | - amenity_stripclub
- | | | - amenity_bar
- | | +- amenity_brothel
- | +- *Shopping*
- | | - amenity_marketplace
- | +- shop_mall
- | - **Accommodation**
- | | - tourism_hotel
- | | - tourism_hostel
- | | - building_hotel
- | | - tourism_motel
- | | - tourism_guest_house
- | | - tourism_apartment
- | | - tourism_chalet
- | | - tourism_alpine_hut
- | | - amenity_camping

- | |– tourism_camp_site
- | |– leisure_beach_resort
- | +- leisure_resort
- |– **Transportation**
- | |– aeroway_aerodrome
- | |– building_train_station
- | |– public_transport_station
- | |– building_transportation
- | |– aerialway_station
- | +- railway_station
- |– **Nature**
- | |– *Landscape*
- | | |– *Landform*
- | | | |– natural_cliff
- | | | |– natural_cave_entrance
- | | | |– natural_peak
- | | | |– natural_glacier
- | | | |– natural_volcano
- | | | |– natural_wood
- | | | |– natural_grassland
- | | | |– natural_heath
- | | | |– natural_sand
- | | | |– natural_rock
- | | | |– natural_mountain_range
- | | | |– natural_valley
- | | | |– natural_ridge
- | | | |– natural_desert
- | | | +- natural_tree
- | | |– *CoastalAreas*
- | | | |– natural_bay
- | | | |– natural_coastline
- | | | +- natural_reef
- | | +- *InlandWaters*
- | | |– waterway_stream
- | | |– waterway_waterfall
- | | |– waterway_canal
- | | |– waterway_river
- | | |– natural_water
- | | |– natural_spring
- | | +- natural_hot_spring
- | +- *ProtectedAreas*
- | |– leisure_nature_reserve
- | |– boundary_national_park
- | +- boundary_protected_area
- +– **Culture**
- |– *Museums*
- | |– tourism_museum
- | |– amenity_arts_centre
- | |– tourism_gallery
- | +- tourism_artwork
- |– *Monuments*
- | |– *Religious*
- | | |– building_cathedral
- | | |– building_chapel

| | | – building_church
 | | | – historic_monastery
 | | | – historic_church
 | | | – building_temple
 | | | – amenity_monastery
 | | | – historic_wayside_cross
 | | +- amenity_place_of_worship
 | +- *Historic*
 | | – historic_fort
 | | – historic_battlefield
 | | – historic_cannon
 | | – historic_citywalls
 | | – historic_ruins
 | | – historic_archaeological_site
 | | – historic_tower
 | | – historic_aqueduct
 | | – historic_city_gate
 | | – historic_castle
 | | – historic_monument
 | | – historic_wayside_shrine
 | | – historic_memorial
 | | – historic_manor
 | | – historic_pillory
 | | – historic_heritage
 | +- historic_tomb
 | – *Viewpoint*
 | +- tourism_viewpoint
 +- *CulturalAmenities*
 | – amenity_fountain
 | – barrier_city_wall
 | – amenity_planetarium
 | – amenity_grave_yard
 +- amenity_crypt

References

1. UNWTO World Tourism Barometer and Statistical Annex, January 2020. *UNWTO World Tour. Barom.* **2020**, *18*, 1–48. [CrossRef].
2. International Tourism 2019 and Outlook for 2020. Available online: <https://webunwto.s3.eu-west-1.amazonaws.com/s3fs-public/2020-01/Barometro-Jan-2020-EN-pre.pdf> (accessed on 6 April 2021).
3. Rathod, A.; Indiramma, M. A Survey of Personalized Recommendation System with User Interest in Social Network. *Int. J. Comput. Sci. Inf. Technol.* **2015**, *6*, 413–415.
4. Haruna, K.; Akmar Ismail, M.; Suhendroyono, S.; Damiasih, D.; Pierewan, A.C.; Chiroma, H.; Herawan, T. Context-Aware Recommender System: A Review of Recent Developmental Process and Future Research Direction. *Appl. Sci.* **2017**, *7*, 1211. [CrossRef]. [CrossRef]
5. Quadrana, M.; Cremonesi, P.; Jannach, D. Sequence-Aware Recommender Systems. *ACM Comput. Surv.* **2018**, *51*. [CrossRef]. [CrossRef]
6. Dara, S.; Chowdary, R.C.; Kumar, C. A survey on group recommender systems. *J. Intell. Inf. Syst.* **2020**, *54*, 271–295. [CrossRef]. [CrossRef]
7. Burke, R. Hybrid Recommender Systems: Survey and Experiments. *User Model User-Adap. Inter.* **2002**, *12*, 331–370. [CrossRef]. [CrossRef]
8. Borràs, J.; Moreno, A.; Valls, A. Intelligent tourism recommender systems: A survey. *Expert Syst. Appl.* **2014**, *41*, 7370–7389. [CrossRef]. [CrossRef]
9. Massimo, D.; Ricci, F. Harnessing a Generalised User Behaviour Model for Next-POI Recommendation. In *RecSys '18, Proceedings of the 12th ACM Conference on Recommender Systems*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 402–406. [CrossRef].

10. Ma, S.; Kirilenko, A. How Reliable Is Social Media Data? Validation of TripAdvisor Tourism Visitations Using Independent Data Sources. In *Information and Communication Technologies in Tourism 2021*; Wörndl, W., Koo, C., Stienmetz, J.L., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 286–293. [CrossRef].
11. Anandhan, A.; Shuib, L.; Ismail, M.A.; Mujtaba, G. Social Media Recommender Systems: Review and Open Research Issues. *IEEE Access* **2018**, *6*, 15608–15628. [CrossRef]. [CrossRef]
12. Tsai, C.Y.; Paniagua, G.; Chen, Y.J.; Lo, C.C.; Yao, L. Personalized Tour Recommender through Geotagged Photo Mining and LSTM Neural Networks. *MATEC Web Conf.* **2019**, *292*, 01003. [CrossRef]. [CrossRef]
13. Dietz, L.W.; Sen, A.; Roy, R.; Wörndl, W. *Mining Trips from Location-Based Social Networks for Clustering Travelers and Destinations*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 22, pp. 131–166. [CrossRef].
14. Van der Zee, E.; Bertocchi, D. Finding patterns in urban tourist behaviour: A social network analysis approach based on TripAdvisor reviews. *Inf. Technol. Tour.* **2018**, *20*, 153–180. [CrossRef]. [CrossRef]
15. Manca, M.; Boratto, L.; Morell Roman, V.; Martori i Gallissà, O.; Kaltenbrunner, A. Using social media to characterize urban mobility patterns: State-of-the-art survey and case-study. *Online Soc. Netw. Media* **2017**, *1*, 56–69. [CrossRef]. [CrossRef]
16. Berndt, J.O.; Rodermund, S.C.; Lorig, F.; Timm, I.J. Modeling User Behavior in Social Media with Complex Agents. In Proceedings of the HUSO 2017—The Third International Conference on Human and Social Analytics, Nice, France, 23–27 July 2017; pp. 18–24.
17. Ishanka, U.A.; Yukawa, T. User Emotion and Personality in Context-aware Travel Destination Recommendation. In Proceedings of the 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), Krabi, Thailand, 14–17 August 2018; pp. 13–18. [CrossRef].
18. Jabreel, M.; Huertas, A.; Moreno, A. Semantic analysis and the evolution towards participative branding: Do locals communicate the same destination brand values as DMOs? *PLoS ONE* **2018**, *13*, e0206572. [CrossRef]. [CrossRef] [PubMed]
19. Jabreel, M.; Moreno, A.; Huertas, A. Do Local Residents and Visitors Express the Same Sentiments on Destinations through Social Media? In *Information and Communication Technologies in Tourism 2017*; Schegg, R., Stangl, B., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 655–668. [CrossRef].
20. Huang, F.; Qiao, S.; Peng, J.; Guo, B.; Han, N. STPR: A Personalized Next Point-of-Interest Recommendation Model with Spatio-Temporal Effects Based on Purpose Ranking. *IEEE Trans. Emerg. Top. Comput.* **2019**, *9*, 994–1005. [CrossRef]. [CrossRef]
21. Massimo, D.; Ricci, F. Next-POI Recommendations Matching User’s Visit Behaviour. In *Information and Communication Technologies in Tourism 2021*; Wörndl, W., Koo, C., Stienmetz, J.L., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 45–57. [CrossRef].
22. Baral, R.; Iyengar, S.S.; Li, T.; Balakrishnan, N. CLoSe: Contextualized Location Sequence Recommender. In *RecSys ’18, Proceedings of the 12th ACM Conference on Recommender Systems*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 470–474. [CrossRef].
23. He, R.; Kang, W.C.; McAuley, J. Translation-Based Recommendation. In *RecSys ’17, Proceedings of the Eleventh ACM Conference on Recommender Systems*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 161–169. [CrossRef].
24. Li, C.T.; Chen, H.Y.; Chen, R.H.; Hsieh, H.P. On route planning by inferring visiting time, modeling user preferences, and mining representative trip patterns. *Knowl. Inf. Syst.* **2018**, *56*, 581–611. [CrossRef]. [CrossRef]
25. Bustamante, A.; Sebastia, L.; Onaindia, E. Can Tourist Attractions Boost Other Activities Around? A Data Analysis through Social Networks. *Sensors* **2019**, *19*, 2612. [CrossRef]. [CrossRef]
26. Farnadi, G.; Tang, J.; De Cock, M.; Moens, M.F. User Profiling through Deep Multimodal Fusion. In *WSDM ’18, Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 171–179. [CrossRef].
27. Orlandi, F.; Breslin, J.; Passant, A. Aggregated, Interoperable and Multi-Domain User Profiles for the Social Web. In *I-SEMANTICS’12, Proceedings of the 8th International Conference on Semantic Systems*; Association for Computing Machinery: New York, NY, USA, 2012; pp. 41–48. [CrossRef].
28. DBpedia. Available online: <https://wiki.dbpedia.org/> (accessed on 29 April 2021).
29. Esmaeili, L.; Mardani, S.; Golpayegani, S.A.H.; Madar, Z.Z. A novel tourism recommender system in the context of social commerce. *Expert Syst. Appl.* **2020**, *149*, 113301. [CrossRef]. [CrossRef]
30. Liji, U.; Chai, Y.; Chen, J. Improved personalized recommendation based on user attributes clustering and score matrix filling. *Comput. Stand. Interfaces* **2018**, *57*, 59–67. [CrossRef].
31. Ma, X.; Lu, H.; Gan, Z.; Zhao, Q. An exploration of improving prediction accuracy by constructing a multi-type clustering based recommendation framework. *Neurocomputing* **2016**, *191*, 388–397. [CrossRef]. [CrossRef]
32. Nguyen, L.V.; Jung, J.J.; Hwang, M. OurPlaces: Cross-Cultural Crowdsourcing Platform for Location Recommendation Services. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 711. [CrossRef]. [CrossRef]
33. Nguyen, L.V.; Hong, M.S.; Jung, J.J.; Sohn, B.S. Cognitive Similarity-Based Collaborative Filtering Recommendation System. *Appl. Sci.* **2020**, *10*, 4183. [CrossRef]. [CrossRef]
34. Fránti, P.; Waga, K.; Khurana, C. Can Social Network Be Used for Location-aware Recommendation? In Proceedings of the 11th International Conference on Web Information Systems and Technologies—WEBIST, Lisbon, Portugal, 20–22 May 2015; INSTICC, SciTePress: Setúbal, Portugal, 2015; pp. 558–565. [CrossRef].

35. Fránti, P.; Mariescu-Istodor, R.; Waga, K. Similarity of Mobile Users Based on Sparse Location History. In *Artificial Intelligence and Soft Computing*; Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., Zurada, J.M., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 593–603. [\[CrossRef\]](#).
36. Najafabadi, M.K.; ri Mahrin, M.N.; Chuprat, S.; Sarkan, H.M. Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Comput. Hum. Behav.* **2017**, *67*, 113–128. [\[CrossRef\]](#). [\[CrossRef\]](#)
37. Pandya, S.; Shah, J.; Joshi, N.; Ghayvat, H.; Mukhopadhyay, S.C.; Yap, M.H. A novel hybrid based recommendation system based on clustering and association mining. In Proceedings of the 2016 10th International Conference on Sensing Technology (ICST), Nanjing, China, 11–13 November 2016; pp. 1–6. [\[CrossRef\]](#)
38. Jalalimanesh, A.; Mansoury, M.; Gandomi, H. Recommender system based on data mining: Interlibrary case study. In Proceedings of the 20th Iranian Conference on Electrical Engineering (ICEE2012), Tehran, Iran, 15–17 May 2012; pp. 806–809. [\[CrossRef\]](#)
39. Fenza, G.; Fischetti, E.; Furno, D.; Loia, V. A hybrid context aware system for tourist guidance based on collaborative filtering. In Proceedings of the 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), Taipei, Taiwan, 27–30 June 2011; pp. 131–138. [\[CrossRef\]](#).
40. Twitter Wikipedia. Available online: <https://en.wikipedia.org/wiki/Twitter> (accessed on 28 April 2021).
41. Twitter Revenue and Usage Statistics. 2020. Available online: <https://www.businessofapps.com/data/twitter-statistics/#:~:text=We%20saw%20a%20recovery%20to,%2435.01%20billion%20in%20September%202019/> (accessed on 30 March 2021).
42. Similarweb Twitter Traffic Overview. Available online: <https://www.similarweb.com/website/twitter.com/> (accessed on 30 March 2021).
43. Twitter API. Available online: <https://developer.twitter.com/en/docs/twitter-api> (accessed on 28 April 2021).
44. Twitter Object Attributes. Available online: <https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet> (accessed on 28 April 2021).
45. Lalicic, L.; Huertas, A.; Moreno, A.; Jabreel, M. Emotional brand communication on Facebook and Twitter: Are DMOs successful? *J. Dest. Mark. Manag.* **2020**, *16*, 100350. [\[CrossRef\]](#)
46. Lalicic, L.; Huertas, A.; Moreno, A.; Jabreel, M. Which emotional brand values do my followers want to hear about? An investigation of popular European tourist destinations. *Inf. Technol. Tour.* **2019**, *21*, 63–81. [\[CrossRef\]](#)
47. Enzensberger, H.M. A Theory of Tourism. *New Ger. Crit.* **1996**, 117–135. [\[CrossRef\]](#)
48. Neff, J.C. Santa Fe and the Tourist. *New Mex. Q.* **1938**, *8*. Available online: <https://digitalrepository.unm.edu/nmq/vol8/iss2/12> (accessed on 8 July 2021)
49. Waga, K.; Tabarcea, A.; Fránti, P. Recommendation of points of interest from user generated data collection. In Proceedings of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), Pittsburgh, PA, USA, 14–17 October 2012; pp. 550–555. [\[CrossRef\]](#)
50. Mariescu-Istodor, R.; Ungureanu, R.; Fránti, P. Real-time destination prediction for mobile users. *Adv. Cartogr. Gisci. Int. Cartogr. Assoc.* **2019**, *2*, 1–7. [\[CrossRef\]](#)
51. OpenStreetMap. Available online: <https://www.openstreetmap.org/> (accessed on 29 April 2021).
52. OpenStreetMap Map Features. Available online: https://wiki.openstreetmap.org/wiki/Map_features (accessed on 30 April 2021).
53. OpenStreetMap Taginfo. Available online: <https://taginfo.openstreetmap.org/tags> (accessed on 30 April 2021).
54. Moreno, A.; Valls, A.; Isern, D.; Marin, L.; Borràs, J. SigTur/E-Destination: Ontology-based personalized recommendation of Tourism and Leisure Activities. *Eng. Appl. Artif. Intell.* **2013**, *26*, 633–651. [\[CrossRef\]](#)
55. Overpass Turbo EU. Available online: <https://overpass-turbo.eu/> (accessed on 29 April 2021).
56. Overpass QL. Available online: https://wiki.openstreetmap.org/wiki/Overpass_API/Overpass_QL (accessed on 30 April 2021).
57. Bird, S.; Klein, E.; Loper, E. *Natural Language Processing with Python*, 1st ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2009
58. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res* **2011**, *12*, 2825–2830.
59. Raschka, S. MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *J. Open Source Softw.* **2018**, *3*. [\[CrossRef\]](#)
60. Agrawal, R.; Srikant, R. Fast Algorithms for Mining Association Rules in Large Databases. In *VLDB'94, Proceedings of the 20th International Conference on Very Large Data Bases*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1994; pp. 487–499.
61. Borràs, J.; Moreno, A.; Valls, A. Diversification of recommendations through semantic clustering. *Multimed. Tools Appl.* **2017**, *76*, 24165–24201. [\[CrossRef\]](#)
62. Golden, B.; Levy, L.; Vohra, R. The orienteering problem. *Nav. Res. Logist.* **1987**, *34*, 307–318. [\[CrossRef\]](#)