IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Smart Tourist Guide

[1]Kavita Jain, [2]Anuja Anilkumar, [3]Ishma Kadam, [4]Dipali Dawande

[1]Assistant Professor, Dept of Computer Engineering, Xavier Institute of Engineering, Mumbai, India
[2,3,4]Students of Xavier Institute of Engineering, Mumbai, India

Abstract: With evolving technology, there are various ways and solutions for travelers to find information regarding their tour destinations. These include tourism websites, social network sites, blogs, forums, and search engines. Information found in these ways is not based on traveler's preferences. Also, many places that deserve to be visited remain untouched due to the lack of public information. Many points of interest may be located within a range of dozens of kilometers from our homes, but usually we prefer to travel hundreds or even thousands of kilometers to visit some other well-known location. This paper proposes a web application which will help to address above challenges. In this paper, we address the development and the key features of a tourist guide application named Explore Mumbai. This application offers a set of searches and filters to facilitate the exploration of new locations, all of which is achieved using python framework- flask. MySQL is used to manage the database. Various web scraping techniques and tools such as Octoparse, request and lxml libraries are used for data collection and dataset creation. Features of places derived from descriptions, reviews of other people who visited those places and history of the user are the key features used to produce a list of recommended locations for the user. Other attributes such as location of the user is kept into mind when there is a cold start problem. To achieve the above goal KNN algorithm is used to compare latitude and longitude of various locations with current user location to recommend nearby locations. Content based recommendation is achieved using description of a hotel and then analysing similarities between other hotel's description using TF-IDF. Collaborative filtering is used to compare other user's review on a place with the user's review. Cosine similarity is used to measure how many similar keywords are present in the descriptions and reviews.

***Index Terms* - Flask, Web Scraping, Content-based Filtering, Collaborative Filtering, Hybrid Filtering.**

I. Introduction

In recent years, the tourism industry has grown tremendously and there has been rapid development in field of information technology across the globe with widespread use of the internet, which has made the process of accessing large amounts of global data from potential customers (tourists) on points of interest, travel plans, and destinations extremely easy[3]. A recommendation system is a program that helps a user find items and content by predicting the user's rating on each place and giving them the items that they would rate highly.

The Recommendation systems are tools providing best items gained from algorithms for a user. These recommendations provided are aimed at supporting their users in various decision-making processes, such as where to plan a tour, what items to buy, what music to listen, or what news to read. Recommender systems helps online users to cope with the information overload therefore becoming one of the most powerful tool in electronic commerce field. Various techniques for recommendation system have been proposed like content-based, user-based and item based collaborative filtering and hybrid recommendation system. The idea behind this tour recommender systems is to automatically recommend items for a user, aiming to predict the user's interest level about the tourist places. These systems help users to deal with information overload, providing personalized recommendations of places. [4].

II. Literature Survey

The TouristEye service [5] is available as an application, with mobile clients. It offers a wide range of points of interest divided into types such as attractions, restaurants, and entertainment. Registered users can like a touristic place after visiting, provide a review stating their degree of satisfaction, and then can describe their trips, by taking notes and photos in the mobile application. In order to visualize these locations of interest, the Google Maps service is integrated within the website and mobile applications. Users can plan their trips, with points of interest, and the map service helps to show routes between these locations. The TouristEye service has an integrated recommendation system such that new locations are automatically displayed to the user.

The Guide Pal Offline City Guides [6] allows users to get different content for cities and to get information regarding restaurants, hotels, coffee shops, places to visit and other attractions. In order to list the existing places to visit, the user selects the desired city and category. Afterwards, a small content for the points of interest is shown.

The mTrip travel guide service [7] is mainly used for large cities such as Paris, Berlin, and Madrid and many others. It is available as a separate application for each one of the major cities and allows users to get information regarding best places to visit without a connection to Internet. Users can plan and create guides for the cities by providing the detailed information on the touristic attractions which they plan to visit. After specifying the dates for the trip, each user can then make its itinerary or allow mTrip to create one. Each location is accompanied by a description, a photo, opening hour, prices, as well as the reviews and ratings from other travelers. It includes an augmented reality tool to see the locations of interest near the users' current location.

The Triposo service [8] offers same attributes to those of mTrip. However, it contains more countries as well as smaller cities compared to it. When user picks the country to visit, the information retrieval regarding the points of interest for that country starts immediately, allowing to get this information later in offline mode. For big cities, it shares special information regarding the city guide about all points of interest, a list of restaurants and extended nightlife options. It also provides a travel dashboard with currency converter, weather information, and useful phrases in native language.

Foursquare [9] is a program that helps registered users to "check-in" at their current location. It provides Web and mobile applications for various users. Users with special permission can contribute with new places, such as coffee shops, restaurants, hotels and sights. The service was created in 2009 and in March 2011 a recommendation system was added for recommending location that users might like, based on their past reviews. In 2013, a new version was published making users to view the sights nearby their current location.

makemytrip.com created to help the Indian tourist with instant booking and variety of choices, the company began its journey in the US India travel market. It aims to provide best value products and services along with amazing technology and round-the-clock customer support. MakeMyTrip provides the vast selection of travel products and services in India. MakeMyTrip is the greatest market-leader with 47% market-share (PhocusWright, 2013).[10]

III. PROPOSED SYSTEM

In this section, we have analysed various roles played by different modules to reach our goal of system development.

1) First the system requires the user to login. If the user doesn't have an account on the system, he/she will be redirected to a registration form.
2) The user can then browse the system thus view various places in Mumbai that can be visited.
3) The user can like, view or review a hotel; view various restaurants, beaches, malls, religious places of Mumbai. Thus, the users browsing history is recorded and stored which can further be used for recommendation.
4) While browsing, various recommendation algorithms run behind and on every web page related places are recommended.
5) If a user wants to plan a trip, a questionnaire is provided. This data is used to filter the recommended places for a user and the places which will be most liked by the user is shown.
6) The user can save a trip and view it later for reference.

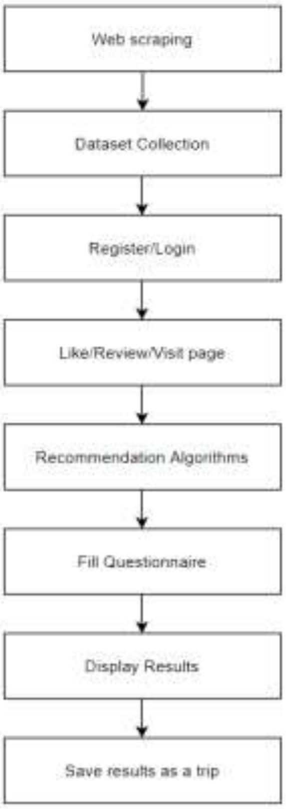


fig. 1 block diagram for proposed system

IV. IMPLEMENTATION METHODOLOGY

Following are the different modules of our system that are implemented.

4.1 Web scraping and Dataset collection

Datasets containing information about details of attractions, hotels and restaurants and their corresponding reviews were required for this project. There was no dataset readily available, thus making Data Collection a very important task of the project. To obtain these datasets, hundreds of attractions, hotels and restaurants on TripAdvisor were crawled. LXML - a python library along with requests - another python library, was used for downloading and processing HTML files using ElementTree API. Text reviews of the hotels were also scraped. To collect the required images of the locations BeautifulSoup library of Python is used. It is a light-weight, easy-to-learn and highly effective way to programmatically isolate information on a single webpage at a time. It is typically paired with the requests library, where requests will fetch a page and it will extract the resulting data. To get the geo coordinates of every destination, a scraping tool named Octoparse, Google maps website along with the address of that destination was used. Users and review dataset were taken from Kaggle dataset.

4.2 Implementation of content-based filtering model

For Content-based recommendation suggests to the user items similar to their past preferences. No other user's feedback or reviews are involved. In this project, content-based filtering is used in two ways:

- by applying it on the description of the hotels and finding the similar hotels according to the keywords
- by finding the users profile by his/her browsing history and the hotel profile (by its description) and then finding the similarities and thus recommending the hotels that will be most liked by the user.

In both the above methods, TF-IDF vectorizer, Term Frequency TF is used to evaluate how important a word is in a description of a hotel in a list of hotels. Formula for the same is:

$$\mathrm{TF(t)} = \frac{\text{Number of times t appears in a description}}{\text{Total number of terms in a description}} \qquad (1)$$

where TF(t) is the Term Frequency of term t

Inverse Document Frequency (IDF) measures how important a term is. While computing TF all terms are equally important. However certain terms like “a”, “an”, “of” etc may have a little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by the formula:

$$\mathrm{IDF(t)} = \log \frac{Total\ number\ of\ descriptions\ (i.e\ hotels)}{Number\ of\ descriptions\ with\ term\ t\ in\ it} \qquad (2)$$

To determine how similar two descriptions are, a metric called cosine similarity is used. Mathematically, it measures the cosine of the angle between two vectors (arrays containing the word counts of two descriptions) projected in a multi-dimensional space. The formula for the same is:

$$\cos\theta = \frac{\vec{\mathrm{a}}\,.\,\vec{\mathrm{b}}}{\|\vec{\mathrm{a}}\|.\,\|\vec{\mathrm{b}}\|} = \frac{\sum_1^n \mathrm{a_i b_i}}{\sqrt{\sum_1^n \mathrm{a_i^2}}\sqrt{\sum_1^n \mathrm{b_i^2}}} \qquad (3)$$

where, $\vec{a}\,.\,\vec{b} = \sum_1^n a_i b_i = a_1b_1 + a_2b_2 + a_3b_3 + \cdots\ a_nb_n$ is the dot product of the two vectors.

The implementation of content-based filtering in two cases are as given below:

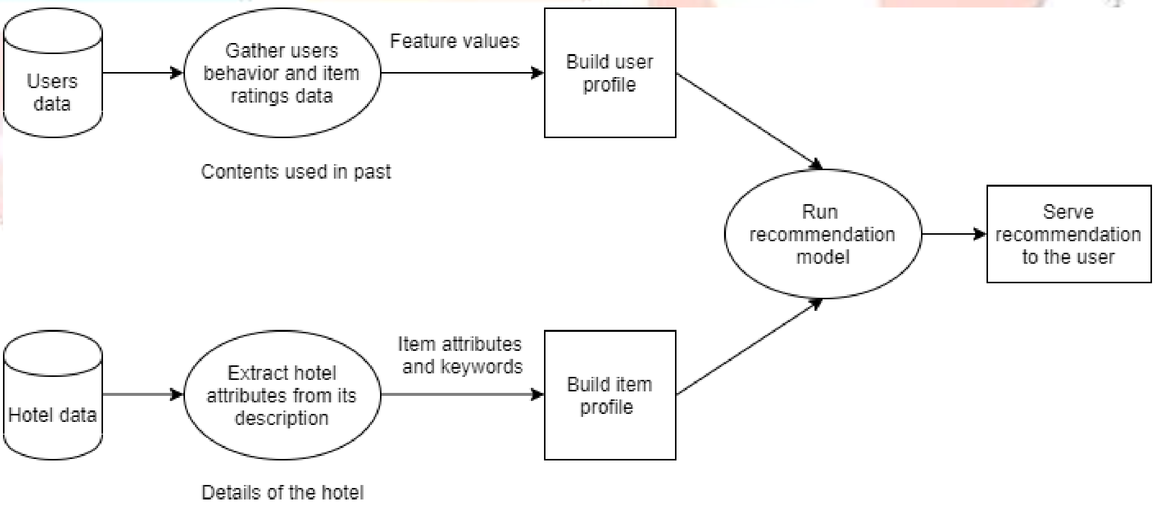

Users data
Gather users behavior and item ratings data
Feature values
Build user profile
Contents used in past
Run recommendation model
Serve recommendation to the user
Hotel data
Extract hotel attributes from its description
Item attributes and keywords
Build item profile
Details of the hotel

Fig. 2 content-based filtering using user profile and hotel description

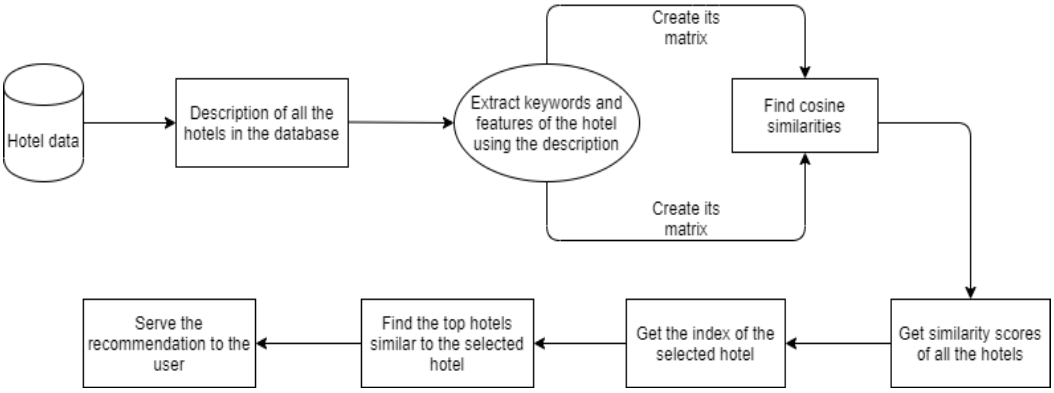

Hotel data
Description of all the hotels in the database
Extract keywords and features of the hotel using the description
Create its matrix
Create its matrix
Find cosine similarities
Get similarity scores of all the hotels
Get the index of the selected hotel
Find the top hotels similar to the selected hotel
Serve the recommendation to the user

Fig. 3 content-based filtering using only the hotel description

4.3 Implementation of popularity-based model

Popularity model is a common baseline approach. This model is not actually personalized -it simply recommends to a user the most popular hotels that the user has not previously reviewed or noticed. As the popularity accounts for the "wisdom of the crowds", it usually provides good recommendations, generally interesting for most people. The interaction score is computed by assigning a score to each type of interaction of the user with the website to provide a numerical value to each action. Here the scores are assigned as, if the user has viewed the hotel, it is assigned as score 1, if the hotel is liked then the score is 2 and if it is reviewed by the user, then the score is 3. For each hotel, an average event strength is calculated and the hotels which are not noticed by the user, and having high interaction score is considered to be more popular and thus is recommended to the user.

4.4 Implementation of recommendation based on reviews

A user is allowed to review a hotel. The review is taken in text format and stored in the database. These text reviews are then analyzed to get the sentiment i.e. if it is positive, negative or a neutral review. This analysis is done using Sentiment Intensity Analyzer library of python and converted into numbers which makes the task of processing easy. The compound value of these scores are considered as the final review score. The flowchart for the same is given in the Fig 4.

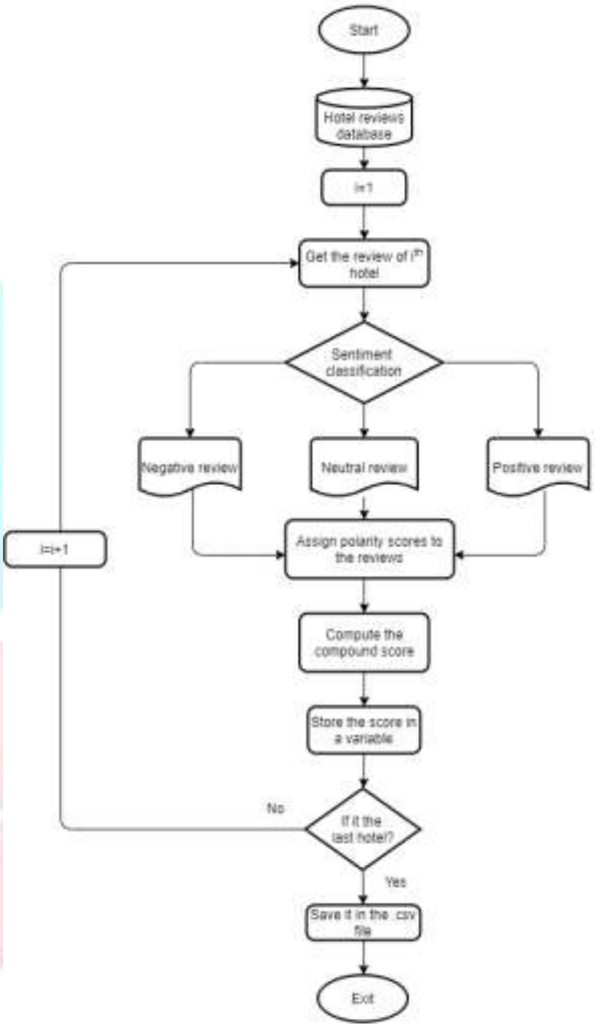


Fig. 4 flowchart for review score calculation

The recommendation process is done using the KNN algorithm. It takes input as the hotel name. The id of the hotel is found in the database using fuzz module of fuzzy wuzzy library of Python. A matrix having user ID and hotel ID as column and row and values as compound score is prepared. This matrix is used for the recommendation. The hotels which are near to the ratings of the selected hotel are then recommended to the user. Cosine similarities is used to measure the distance between two users.

4.5 Implementation of collaborative filtering

Collaborative Filtering filters (predicts) about the user’s interest by collecting preferences or information from many users. The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on a set of items, A is more likely to have B's opinion for a given item than that of a randomly chosen person. In this project information of the user's interaction with the items is collected, by saving this data in the session variable and retrieving it back into the database when the session expires. The following interactions of any registered user is saved

- the hotels he has marked as favourites by clicking on the like button
- the pages he viewed by clicking on the link of the hotel name to view the information page of that particular hotel
- the reviews that the user gave to the hotels he visited.

Each of these interactions are then given a numerical value based on the interest level. Then latent factor model named Singular Value Decomposition (SVD) is used. Latent factor models shrinks a user-item matrix into a low-dimensional representation in terms of latent factors. Hence, instead of having a high dimensional matrix containing abundant number of missing values we will be dealing with a much smaller matrix in lower-dimensional space.

4.6 Implementation of hybrid filtering

Half breed recommender frameworks join at least two suggestion systems in various manners to profit by their corresponding favorable circumstances. In this project hybrid model of content based filtering and collaborative filtering is used. A simple hybridization method was built, as an ensemble that takes the weighted average of the normalized Collaborative filtering scores with the Content-Based scores, and ranking by resulting score. In this case, as the CF model is much more accurate than the CB model, the weights for the CF and CB models are 100.0 and 1.0, respectively. The procedure is as follows:

- Getting the top-100 Content-based filtering recommendations.
- Getting the top-100 Collaborative filtering recommendations.
- Combining the results by hotel id.
- Calculating a hybrid recommendation score based on CF and CB scores.
- Sorting recommendations by hybrid score.
- Returning the recommendations.

V. Evaluation

One key part of assessment is to guarantee that the prepared model sums up for information it was not prepared on, utilizing Cross-validation systems. A straightforward cross-validation approach named holdout, in which an irregular information test (30% for this situation) are kept aside in the preparation procedure, and solely utilized for assessment. All assessment measurements announced here are figured utilizing the test set. In Recommender Systems, there are a few set measurements usually utilized for assessment. We decided to work with Top-N exactness measurements, which assesses the precision of the top suggestions gave to a client, contrasting with the things the client has really interfaced in test set. The Top-N precision metric picked was Recall@N which assesses whether the interfaced thing is among the top N things (hit) in the positioned rundown of suggestions for a client. The Recall@5 of state 0.2417, implies that about 24% of cooperated things in test set were positioned by the model among the main 5 things (from records with 100 irregular things). The following is the scrap of Recall@N measurements of the fundamental models executed:

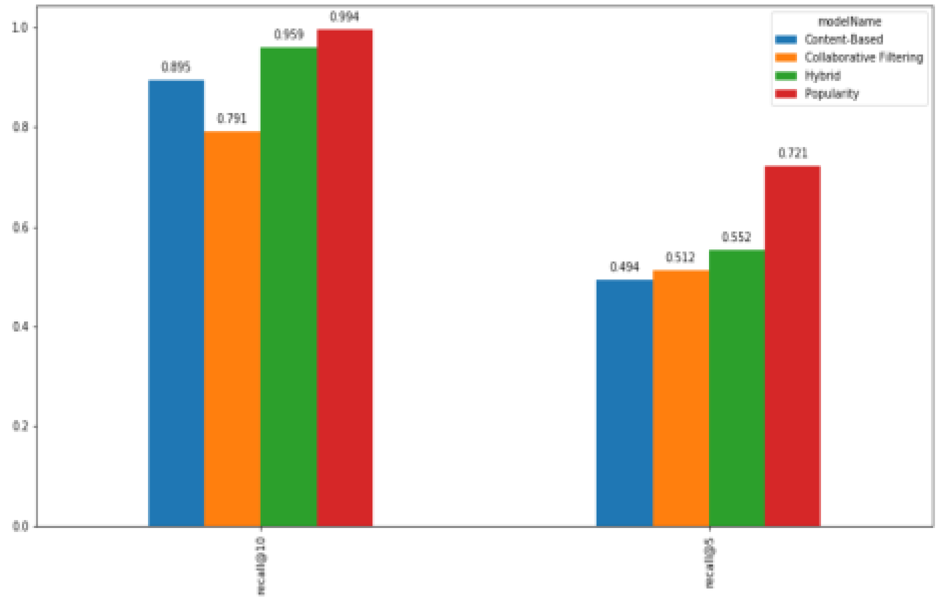
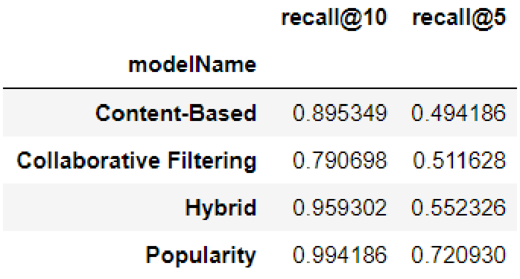


Fig. 6 graphical representation of performance metrics of all the models



	recall@10	recall@5
modelName		
Content-Based	0.895349	0.494186
Collaborative Filtering	0.790698	0.511628
Hybrid	0.959302	0.552326
Popularity	0.994186	0.720930

Fig. 7 summary of evaluation of all the models

VI. Conclusion

Our project objective is to introduce a personalized computer application that automatically predicts which place a user is most likely to visit in Mumbai. The dataset is first obtained by web scraping from TripAdvisor website. This data is then fed into various algorithms such as content-based filtering, nearest location algorithm, collaborative filtering and popularity model to recommend places to visit in Mumbai.

We have learned and demonstrated how a user can be suggested places by using the user’s history, popularity of places, description of the place and how close the place is to other places of interest. We have been successful in suggesting the best places for user using mixture of the above elements and algorithms. The system is an approach to ease the difficulty in finding the best places to visit based on user’s personal interests. We have also tried to create a hybrid model using content and collaborative model so that advantages of both the methods can be combined and used. During the recommendation process we have found out that popularity model works best and more users are susceptible to visiting places that are more popular than others.

References

[1] Wahidah Husain and Lam Yih Dih “A Framework of a Personalized Location-based Traveler Recommendation System in Mobile Application”, International Journal of Multimedia and Ubiquitous Engineering Vol. 7, No. 3, July, 2012

[2] Artem Umanetsa, Artur Ferreiraa, Nuno Leitea “GuideMe-ATourist Guide with a Recommender System and Social Interaction”, Procedia Technology 17 (2014) 407 – 414

[3] Raheleh Hassannia, Ali Vatankhah Barenji, Zhi Li and Habib Alipour “Web-Based Recommendation System for Smart Tourism: Multiagent Technology”, Sustainability 2019, 11, 323

[4] Ms. Navaneetha H P and Dr. Sharath Kumar Y H “Tourist Attraction recommendations using collaborative filtering”, International Journal of Research, Volume 03 Issue 11 July 2016

[5] J. Escribano and A. Camus. La guia de viajes inteligente que aprende de ti y tus amigos, November 2012. www.slideshare.net/ betabeers/touristeye.

[6] GuidePal, Inc, guidepal.com. GuidePal Home, 2012.

[7] mTrip, Inc, www.mtrip.com. mTrip - Intelligent Travel Guides, 2011.

[8] Triposo, Inc, www.triposo.com. Triposo Travel Guides, 2012.

[9] Foursquare, Inc, foursquare.com. Building a recommendation engine, Foursquare style, March 2011.

[10] Deanne Charan, Apoorva Chouhan, Avelon Pereira “Smart Tourist System”, International Journal Of Engineering And Computer Science ISSN: 2319-7242 Volume 5 Issue 12 Dec. 2016

[11] Omamah Alnogaithan, Sumaiah Algazlan, Aljoharah Aljuraiban, Amal A. Shargabi “Tourism Recommendation System Based on User Reviews”, 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)

[12] Y. Wang, S. C. Chan and G. Ngai, "Applicability of Demographic Recommender System to Tourist Attractions: A Case Study on Trip Advisor," *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Macau, 2012, pp. 97-101.

[13] Savitaa Venkateswaran, Subikshaa Senthilkumar and Sachin Prabhu Thandapani, “Intelligent Tourist Recommendation System”.

IJCRT