VACATION PLANNING USING GEO LOCATION CLUSTERING

B.Tech. Major Project Report

BY

Doppalapudi Gokul Sai (15115035)

Pathipati Lokesh (15115061)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY RAIPUR

C.G. (INDIA)

MAY, 2019

VACATION PLANNING USING GEO LOCATION CLUSTERING

MAJOR PROJECT REPORT

Submitted in the partial fulfilment of the Requirements for the award of degree

Of

Bachelor of Technology

In

Department of Computer Science and Engineering

BY

Doppalapudi Gokul Sai (15115035)

Pathipati Lokesh (15115061)

DEPARTMENT OF COMPUTER SC. & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY RAIPUR

C.G. (INDIA)

MAY, 2019

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Major Project Report entitled *"***Vacation Planning Using Geo location clustering***"* in partial fulfilment of the requirements for the award of the **Bachelor of Technology** in **Computer Sc. & Engineering** and submitted to the Department of Computer Sc. & Engineering of National Institute of Technology Raipur is an authentic record of my own work carried out during a period from January 2019 to May 2019 under the supervision of **Dr. Dilip Singh Sisodia, CS&E Department**.

The matter presented in this report has not been submitted by me for the award of any other degree elsewhere.

Signature of Candidate
Doppalapudi Gokul Sai
Roll No. 15115035

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of Supervisor
Date: **Dr. Dilip Singh Sisodia**
Assistant Professor (CS&E)

Head of Department
Computer Science & Engineering Department
National Institute Of Technology Raipur

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Major Project Report entitled *"***Vacation Planning Using Geo location clustering***"* in partial fulfilment of the requirements for the award of the **Bachelor of Technology** in **Computer Sc. & Engineering** and submitted to the Department of Computer Sc. & Engineering of National Institute of Technology Raipur is an authentic record of my own work carried out during a period from January 2019 to May 2019 under the supervision of **Dr. Dilip Singh Sisodia, CS&E Department**.

The matter presented in this report has not been submitted by me for the award of any other degree elsewhere.

Signature of Candidate

Pathipati Lokesh

Roll No. 15115061

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of Supervisor

Date: **Dr. Dilip Singh Sisodia**

Assistant Professor (CS&E)

Head of Department

Computer Science & Engineering Department

National Institute Of Technology Raipur

Abstract

The discovery of places of interest in large cities is a major problem for those who are interested in sightseeing. We propose a project to find the travel route that helps the visitor in order to cover maximum number of places with in a given limited amount of time. This project also has a benefit of trip planning and location recommendation. The effective implementation of this project faces two problems, how to identify the far away points and how to form the clusters for the selected locations. After addressing the above challenges, we need to find a route map as the user specified.

This project consists of two modules, one for local site seeing (probably less than one day) and another for global site seeing (normally more than one day). For local site seeing we will be applying path finding algorithm for the input time and for the selected places.

The locations which the user want to visit, time, start location and the destination will be sent to back end. For the selected locations we are using three algorithms, HDBSCAN clustering algorithm, k-means clustering algorithm and path finding algorithm.

By using HDBSCAN (Hierarchal Density Based Spatial Clustering of Applications with Noise) algorithm, we will be able to find the noise points or the outliers (the points that are far from the remaining locations). The noise points can be removed from the actual plan. HDBSCAN computes clusters based on the density (reachability) of the selected places.

For the remaining places (outcome of HDBSCAN (without outliers)), using k-means algorithm, we will get the day wise plan for the visitor to visit the city. K-means algorithm computes clusters based on the minimum mean square distance from the centroid. In this algorithm we are going to cluster the places based on the number of days.

By using the path finding algorithm, we will be providing the greedy path which covers the maximum number of places which visitor can cover in the time span given by the user .In that path plan we will be providing user the locations which he must travel in a sequential to ensure covering maximum places. The output will be consisting of the route map from source to destination through the places that are specified by the user. Along with the names we are also providing the visiting time of a particular place on the map.

ACKNOWLEDGEMENT

We would like to thank our project advisor, **Dr. Dilip Singh Sisodia,** Assistant Professor, Dept of CS&E, NIT Raipur, for introducing this project and for his inspiring guidance, constructive criticism, valuable suggestions and support. We also thank him for his constant feedback and expertise that paved a way forward for us.

We would also like to thank our Head of Department. **Dr. Pradeep Singh** for providing us with the necessary facilities and constant guidance and motivation.

We would also like to thank **Dr. A.M. Rawani,** Director NIT Raipur for providing us with all the necessary facilities and infrastructure.

Doppalapudi Gokul Sai (15115035)

Pathipati Lokesh (15115061)

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1. Introduction

The main goal of this application is to provide an optimal visit plan for a tourist visiting a location and selecting some places out of given tourism attraction places in a city. The trip time, start location, end location, city and places of interest were taken as an input from the user. We have developed this process in 2 phases. In the first phase we have worked on the front end as we want to give the flavour of client-server model, In this phase our primary focus is on providing the user with set of functionalities along with good design interface along with redirecting the page to the server internet protocol address with the data fields submitted by user as query parameters with these our first phase ends. In our second phase we have primarily focused on extracting the values of the query parameters and storing these values in the global variables and performing join operations to identify the places selected by the user. One of the query parameters contains number of days of visit or the time visitor is going to spend in the city

This project consists of two modules, one for less than one day and another for greater than one day. For less than one day we will be applying path finding algorithm for the input time and for the selected places. Output consists of a graph showing the path containing places from source to destination covering maximum number of places in the city.

If the time is more than one day we apply clustering algorithms to identify the places that can be visited on a day wise basis. First, we need to prune the given locations and has to identify the far away points (noise). For this we are using HDBSCAN algorithm. By using HDBSCAN clustering algorithm we will be able to locate the far away points from the remaining in the form of outliers or noise. These outliers are excluded from the original plan. For the remaining points we apply k means algorithm and then plotting these locations with their respective latitude and longitude on the map giving one colour to each cluster to identify whether clustering was properly performed or not, along with the centroids of each cluster. For each cluster we will apply the minimum path finding algorithm to find the effective path covering all places of a particular day in optimal or minimal distance path.

We will be asking the user to submit his feedback for the places he visited in the form review and rating. Based on the review and rating the new rating will be generated either by simple mean or Bayesian mean we update the rating of all the locations based on the feedback in the database.

CHAPTER 2

REQUIREMENTS

2. Requirements

2.1. Python

It is a high level and interpreted programming language. It is easily understandable and easy torrent to create blocks. Python has a rich set of libraries which shows its importance in various applications like machine learning, data analysis, artificial intelligence etc.

2.2. Numpy

For scientific computation with high performance array objects and arrays we have this famous package available in python [2].

2.3. Pandas

In order to make working with relational or labelled data easy and understand we use pandas which is python package providing fast, flexible and expressive data structures. For doing practical and real world data analysis in python pandas serve as a high-level building block. Pandas aims to become the flexible open source data analysis or manipulation tool available and most powerful in any language [1].

2.4. HTTP Server

This is used to implement a python program as a server program with the help of some methods present in-built in it. We can specify the actual port in which the program should run in order to give the flavour of server application.

2.5. Beautiful soup

Whenever you have some data in hypertext mark-up language file (.html) or in a .xml file, you want to extract data from those files we use beautiful soup [1].

2.6. Url Parse

This library is used to separate the query parameters from the url(uniform resource locator) and we can extract the value of the query parameters from the methods specified in this library.

2.7. Extracting the data Set

Open Google maps and login to your account and open your places in Google maps then go to MAPS then click on create map which will take you to a new window. In the search bar search the location for which you need the latitude and longitude and add a marker at that place. Repeat the above statement for all the locations you have and add the markers to all the locations and name the map and click on the Export to KML option present over there. Which will exports all the locations and their respective names to .kml file with the name of file being the name of the map you create.

Co-ordinates are being stored in <coordinates > </coordinates> tag which we can extract the latitude and longitude and store it in a data structure and similarly fetch the name of the location by using the <name></name> tag and apply the same procedure to store the names in the array. Save the file in same directory as python execution working directory.

2.8. Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits [7].

2.9. SQL DataBase

SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system.

However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database. This tutorial will provide you with the instruction on the basics of each of these commands as well as allow you to put them to practice using the SQL Interpreter.

2.10. Xampp server

The full form of XAMPP is X stands for Cross-platform, (A) Apache server, (M) MariaDB, (P) PHP and (P) Perl. The Cross-platform usually means that it can run on any computer with any operating system. XAMPP is an open source software developed by apache friends. XAMPP software package contains Apache distributions for Apache server, MariaDB, PHP, and Perl. And it is basically a local host or a local server. This local server works on your own desktop or laptop computer. The use of XAMPP is to test the clients or your website before uploading it to the remote web server. This XAMPP server software gives you the suitable environment for testing MYSQL, PHP, Apache and Perl projects on the local computer.

CHAPTER 3

ALGORITHM USED

3. Algorithm used

3.1. k-Means clustering algorithm

3.1.1. Description

Provided n set of observations $(x_1, x_2, \ldots, x_n)$ in a d-dimensional space, k-means clustering algorithm divides the given n observations into k clusters $\{C_1, C_2, \ldots, C_k\}$ where (k<=n) inorder to minimize the sum of squares with in clusters. The objective is to find

$$\arg\min \sum_{i=1}^{k} \sum_{x \in S_i} |x - \mu_i|^2 = \arg\min \sum_{i=0}^{k} |S_i| Var\, S_i \tag{1}$$

Where μ_i = mean of points in S_i form Eq.(1), the above values are equivalent to the minimizing pair-wise square deviations of the same cluster[14][20]. The sum of square of distances from mean to each point in the cluster Eq.(2) is minimized.

$$\arg min \sum_{i=0}^{n} \frac{1}{2|S_i|} \sum_{x,y \in S_i} |x - y|^2 \tag{2}$$

3.1.2. Initialization

In this k-Means clustering, we made a greedy approach includes choosing k random observations out of provided n observations as the initial centroids of the k clusters. The random partitioning method assigns each observation a cluster and finds the mean of observations as the centroid of the cluster and goes for an update step and repeats the above process till we get the same set of means in consecutive iterations and stop [7]. This algorithm follows a heuristic approach, so there will be no optimum solution but it can provide the most approximate results. The initial selected mean [Fig 1] observations may affect the final result of the formed clusters [18][23]. In worst case k-Means will converge too slowly.

K-Means algorithm step wise procedure

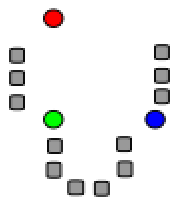


Figure 1- Selecting k (k=3) observations as initial mean [7]

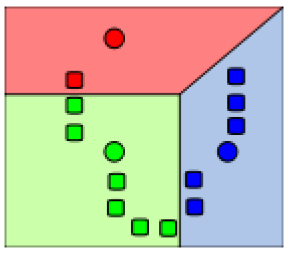


Figure 2- Clusters are created with nearest mean [7]

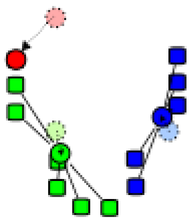


Figure 3- Generated new means for the above formed clusters [7]

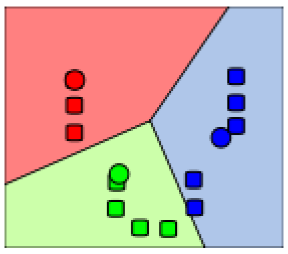


Figure 4- Repeat above 2 steps until convergence is reached [7]

Figure 1-4 shows how the k-means algorithm reaches the optimal solution.

3.1.3. Algorithm

It uses the iterative approach with a modification (closing to the convergence) of centroids in each iteration.

Given with the initial set of observations (x_1, x_2, …, x_n) select k initial means say m_1,…,m_k .

Step 1: Assigning

Assign a cluster to each and every observation whose mean has least distance [24]. This way each observation will be assigned to a single cluster.

$$S_i{}^t = \left\{x_p : \left\|x_p - m_i{}^t\right\|^2 \leq \left\|x_p - m_j{}^t\right\|^2 \ \forall j, 1 \leq j \leq k\right\} \tag{3}$$

Eq.(3) assigns the cluster to the point x_p by calculating square of Euclidian distance from mean point.

Step 2: Updating

Eq.(4) calculates the centroids for each cluster.

$$m_i{}^{(t+1)} = \frac{1}{\left|S_i{}^t\right|} \sum_{x_j \in S_i{}^t} x_j \tag{4}$$

If the set of centroids matches with the previous output of this step then stop further process, if not repeat the above steps until there is no change in the set of centroids. Till now we had reached the convergence. There is no guarantee for the optimum solution.

In k-Means the distance is calculated as the Euclidian distance, but many modifications are done for this as earth is not flat, it is spherical but Euclidian distance is the flat distance between two points in a space. Fig 5 represents the clusters and their centroids.

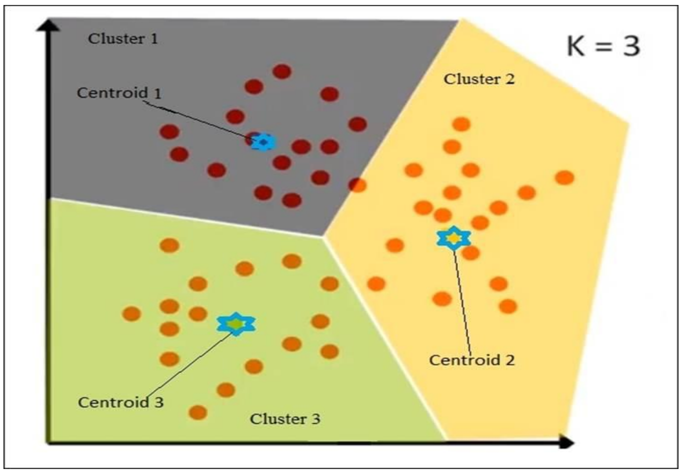


Figure 5- Clustering using k-Means representing the centroids [7].

3.1.4. Complexity

"It is an NP-hard problem[26]. If k and d (the dimension) are fixed, the problem can be exactly solved in time O($n^{(dk+1)}$), where n is the number of entities to be clustered [5]."

3.2. HDBSCAN

HDBSCAN (Hierarchal Density Based Spatial Clustering of Applications with Noise). As the earth isn't flat, when we made clustering by k-means we end up by clusters having done by Euclidian distance means k-means uses Euclidian distance to calculate the distance between two points where as hdbscan uses haversine formula to calculate the distance between 2 points meaning lengths of the shortest curve between two points along the surface of the Earth [6].

Surprisingly hdbscan will not allow us to choose no of clusters whereas k-means allow us to specify the no of clusters we desire to distribute the data set. Minimum size of cluster in a hdbscan is two [28].

3.2.1. Haversine formula

The haversine formulae determine the curved distance between two points on the curve rather than giving the straight line distance between the 2 given points. Very useful in calculating distance between two given on earth because they two are connected by curve on the sphere (earth) [8].

Between any two points on a sphere the central angle _Θ is defined in Eq.(5)

$$\Theta = \frac{d}{r} \tag{5}$$

where:

- d is the distance between the two points in Eq.(5),
- r is the radius of the sphere in Eq.(5).

The haversine formula is defined in Eq.(6)

$$\text{hav}(\Theta) = \text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\, hav(\lambda_2 - \lambda_1) \tag{6}$$

Where,

- φ_1, φ_2: latitude of point 1 and latitude of point 2 in Eq.(6),
- λ_1, λ_2: longitude of point 1 and longitude of point 2 in Eq.(6).

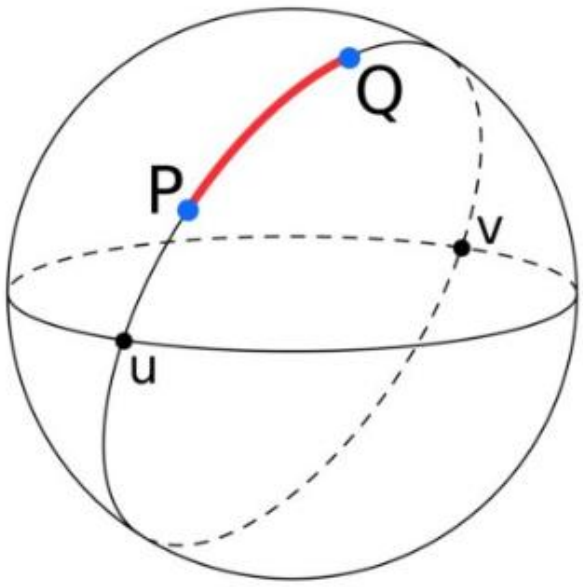


Figure 6- Shortest curve between two points along the surface of the Earth [10].

Fig 6 represents the shortest curved distance between 2 points P and Q along the surface of the earth.

For an angle θ the haversine function is defined as below:

$$hav(\theta) = \sin(\frac{\theta}{2})^2 = \frac{1-\cos\theta}{2} \tag{7}$$

Applying the inverse haversine hav^{-1} to solve for the distance d, to the central angle Θ:

$$d = rhav^{-1}(h) = 2r\, arc \sin(\sqrt{h}) \tag{8}$$

Where h=hav(Θ), or more explicitly:

$$\begin{aligned} d &= 2r \arcsin\left(\sqrt{\text{hav}(\varphi_2-\varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\,\text{hav}(\lambda_2-\lambda_1)}\right) \\ &= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right)(\varphi_2-\varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \end{aligned} \tag{9}$$

Eq.(7-9) are used to calculate the curved surface distance between 2 points having latitudes φ_1, φ_2 and longitudes at λ_1, λ_2 .

3.2.2. **Working of Density Based SCAN**

To understand the working of hdbscan in practice, we need an approximation of density and how far the k^{th} nearest neighbour is. Otherwise if somehow we can find the distance of every point from every other point that is if we can form matrix consisting of distance of a point from every other point we can look from it. Let us denote the distance of k^{th} nearest neighbour from appoint x as core_k(x) [6][8]. Our aim now is to disperse the points with low density, one of the way to do the above mentioned is to define a new metric distance between the points that are called, which we often refer as mutual reachability distance. We define the mutual reachability distance as mentioned in [eq (10)]

$$d_{mreach-k}(a,b) = max\{core_k(a), core_k(b), d(a,b)\} \tag{10}$$

Above $d_{mreach-k}$ from Eq.10 is the distance used to disperse the points in the cluster. This process is followed upto i iterations until the point remains in the same cluster[11].

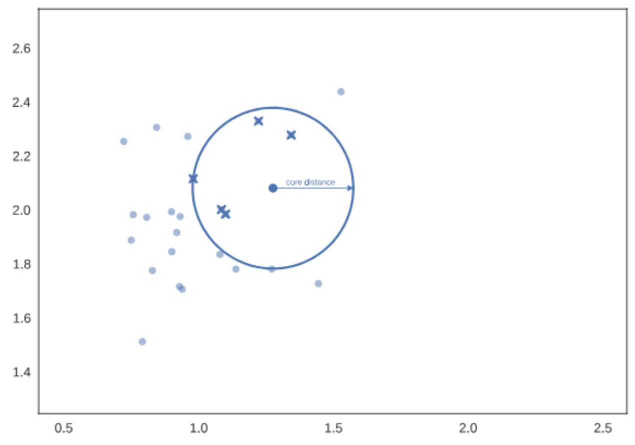


Figure 7- Draw a circle with radius r [8]

With a different set of neighbours pick another point and repeat the same thing. The original distance between a and b from above formulae is d(a,b). The points with low core distance often referred as metric dense points remain at same distance from each other whereas the points with higher density often referred as sparser points are pushed away.

The condition we can obviously specify is dependent on the choice of k, larger the value of k more points are interpreted as being in the sea. So use the values of k somewhere between five to ten. So for every given point with the core distance as radius draw a circle which will touch the sixth nearest neighbour.

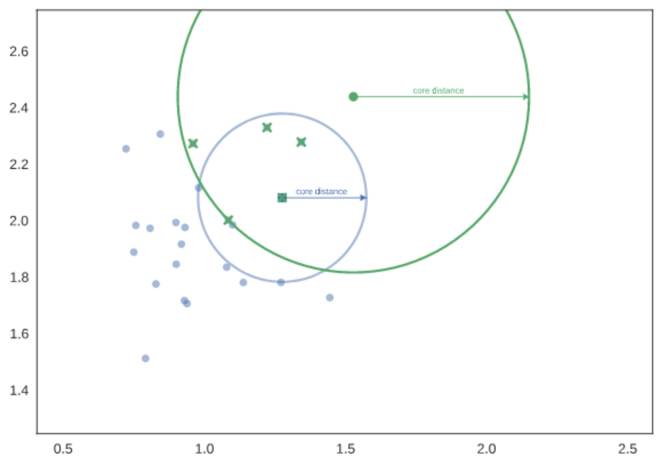


Figure 8- circle touching sixth nearest neighbour [8]

With another set of nearest neighbours repeat the same thing as to get a good measure with other circle of different radius

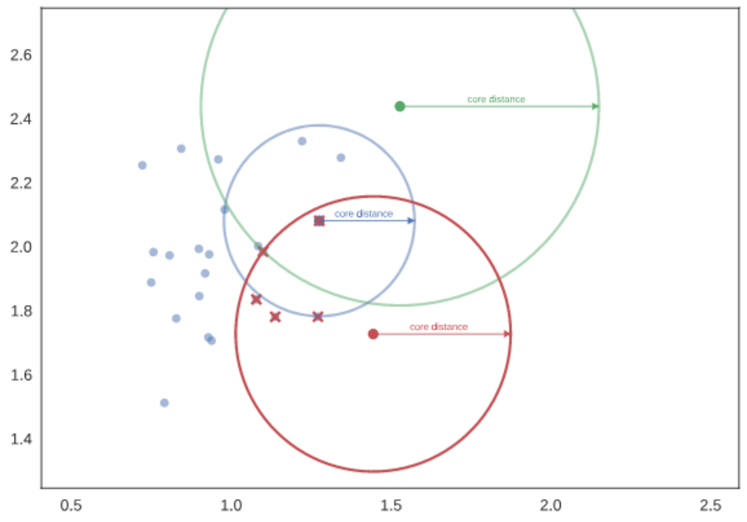


Figure 9- Repeat the same procedure of drawing circles [8]

For finding the distance between the blue and green points we can do that by drawing an arrow which gives the distance between green and blue. The distance between green and red is same as the distance from red to green as the distance from red to green is greater than core distance.

3.3. Path finding algorithm

3.3.1. Description

Provided n set of points in a 2-dimensional plane {$A_1(x_1 , y_1)$, $A_2(x_2 , y_2)$,… $A_n(x_n, y_n)$}, along with the time taken to process a particular point, path finding algorithm finds the path between two points (namely source and destination), by covering maximum number of points between them within the given amount of time. The main objective is to find a path between the source and the destination by covering maximum number of places within a mentioned time.

3.3.2. Initialization

In this algorithm, the execution will start from the source. We will calculate the time taken to move from source to destination. If the above calculated time is less than the available time, then we find the new source by finding minimum of visit time and travelling time combined [19]. In this process we find the new source and we mark the previous source as visited. The algorithm will stop after failing the first mentioned condition [19].

3.3.3. Algorithm

1. Mark all the points as not visited, take input from user the available time.
2. Identify the source and the destination, note visit time (vt) for all the points on the plane.
3. Calculate time taken to move from source to destination (consider the points are on earth (geographical), distance will be in kilometers (using Eq.(9) in 3.2.1)(haversine distance of sphere) and assume the speed to be 30kmph).
4. While the time taken is less than the available time, then do step 5 to 10
5. Find the updated visit times (uvt) to all the points on the plane as

$$uvt(A_i) = \{vt(A_i) + [(\text{distance between source }, A_i)/speed]*60\} \quad (12)$$

6. Find the place with minimum uvt.
7. Mark the source as visited.
8. Make the place with minimum uvt as new source.
9. Update the available time as

$$\text{Available time} = \text{available time} - uvt(A_i) \quad (13)$$

10. Calculate time taken to move from source to destination and goto step 4.

3.3.4. Demonstration

Here, let us consider the map given below (Figure 1) as the initial map to go for the path finder algorithm. Consider the available time =200 minutes.

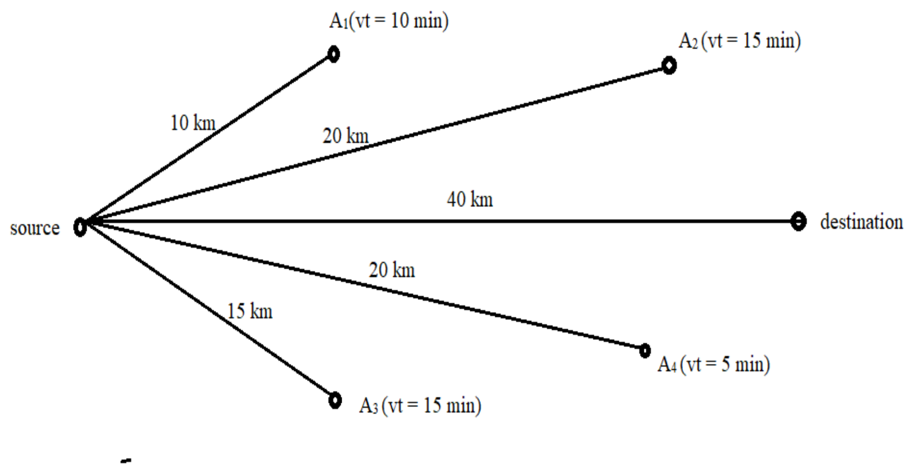


Figure 10- Initial map showing places with visit times and distances.

Before start of the algorithm,

Place	Visit time(min)	uvt(min)	visited
source	0	0	0
A_1	10	0	0
A_2	15	0	0
A_3	15	0	0
A_4	5	0	0
destination	0	0	0

Table 1-Iteration 0 of path finding algorithm

Calculate the time taken for source to destination (in the above case it is 80 minutes).

The calculated time is less than the available time. So, go for iteration 1.

Calculate the updated visit times (uvt) for every place ($A_{i\ (\text{where } i = 1 \text{ to } 4)}$) using Eq.(12).

Choose the place with minimum uvt from the below table as a new source (in this case A_1) if it is not visited, mark the previous source (Source) as visited

Calculate the remaining time using (Eq.13)

The uvt for places will be as follows

Place	Visit time(min)	uvt(min)	visited
source	0	0	1
A_1	10	30	0
A_2	15	55	0
A_3	15	45	0
A_4	5	45	0
destination	0	80	0

Table 2-After iteration 1 of path finding algorithm

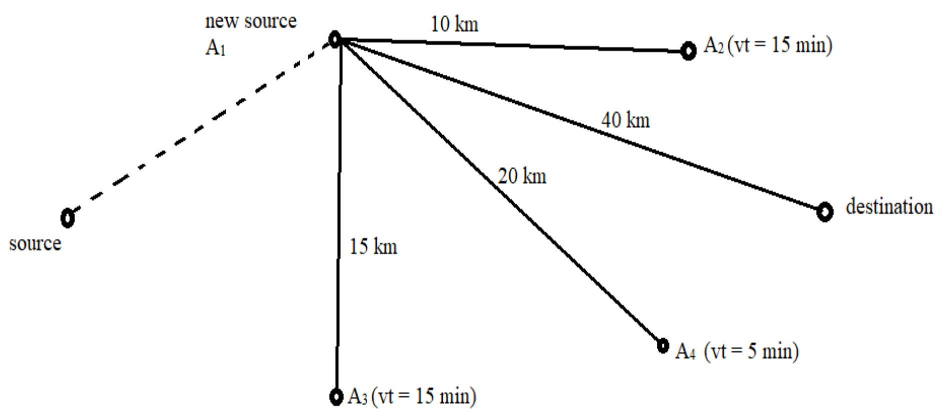


Figure 11- Map showing places with visit times and distances after iteration 1.

Do the above procedure for the new source and destination (Figure 2).

CHAPTER 4
IMPLEMENTATION

4. Implementation

4.1. Introduction

In this chapter, we will get to know how the connection is established from user end and the server, how the data is flowing throughout the process, how the data is sent from user to the server and how the data is processed in the python code and the user will be given the output.

4.2. Connecting client and server

When user fills all the mandatory fields and click on submit button it will navigate or redirect him to a specific url with the help of window.location.href property available in JavaScript, url along with ip address and port number of the server contains query parameters and the values associated with it so that server will be able to parse the url and extract the values associated with the query parameters and store these values in the server

4.3. Data flow

1. Turn on the server

Run the python code containing the server. The server will execute forever (server enters the infinite loop) on execution of the following line.

httpd.serve_forever()

2. Taking input from user

The user input consist of two fields, number of days the visitor is willing to visit the city and the places that visitor wants to visit in Raipur. The places are selected using checkboxes. On clicking the submit button at the user end the user input is send to the server where the python program for clusters is running. User will be provided with a web page containing two main fields as location and number of days he would like to spend in the particular location selected. After selecting these values user will be provided with a number of tourist places available in the respective location and required to check all the places he like to visit. After selecting the required places user clicks on the submit button available in the form so that the user input is redirected to server.

3. Processing .kml file

The .kml file is parsed using bs4 package. This .kml file consists of co-ordinates and the names of the places (places of interest) of Raipur inside the tags coordinates and name. These tags are parsed and the values of coordinates and name are stored in a numpy array (a multi-dimensional array). The kml file is parsed using below syntax

kml_soup = Soup(data, "lxml-xml")

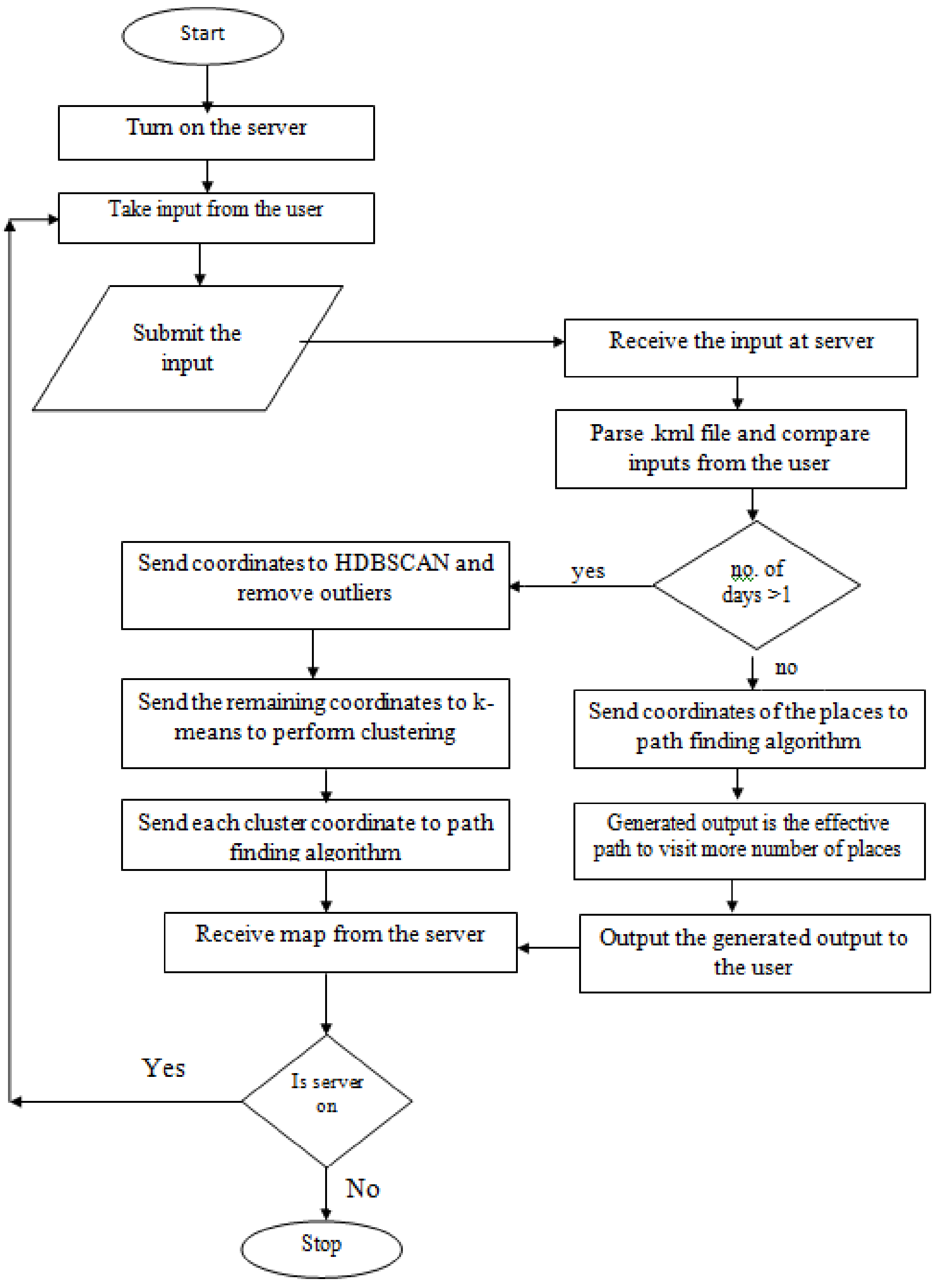
4. Creating a Dataframe

The user input collected at the server is processed and the names of the places off the visitors wish are compared with the names in the database(in this case Placemark.kml) file and the matched names location(latitude and longitude) are stored into a dataframe.

5. Processing input

The dataframe that is created is given as an input to the path finding algorithm. The final output of the path finding algorithm is given to the user at the user end. After receiving the user input parse the input through url to extract the query parameters and store the value of these query parameters in the global variables. After storing the variables in global parameters, we perform the path finding algorithm on the received data by following the steps mentioned in section 3.1.3. It will be giving the output of the places that are to be visited in order so that one can visit a maximum number of places with in the available time. Using matplotlib.pyplot package, plot the locations (those are in output) on the map (using latitude and longitude of the place) along with the path connecting them in order. With these one can get an idea how to visit the places in a city [9].

4.4. Flow chart



Start
Turn on the server
Take input from the user
Submit the input
Receive the input at server
Parse .kml file and compare inputs from the user
no. of days >1
yes
no
Send coordinates to HDBSCAN and remove outliers
Send the remaining coordinates to k-means to perform clustering
Send each cluster coordinate to path finding algorithm
Send coordinates of the places to path finding algorithm
Generated output is the effective path to visit more number of places
Output the generated output to the user
Receive map from the server
Is server on
Yes
No
Stop

Figure 12– Flow of data

CHAPTER 5
RESULTS

5. Results

In this section, we had included the snapshots of the output of server and client interface. The below snapshots are taken by selecting Raipur as example and by selecting all places available for consideration. The latter are the snapshots of the output of the server interface showing the latitude and longitude details of the selected locations from the user and the division of clusters.

We had included the plots along with latitude and longitude of the selected locations after implementing path finding algorithm.

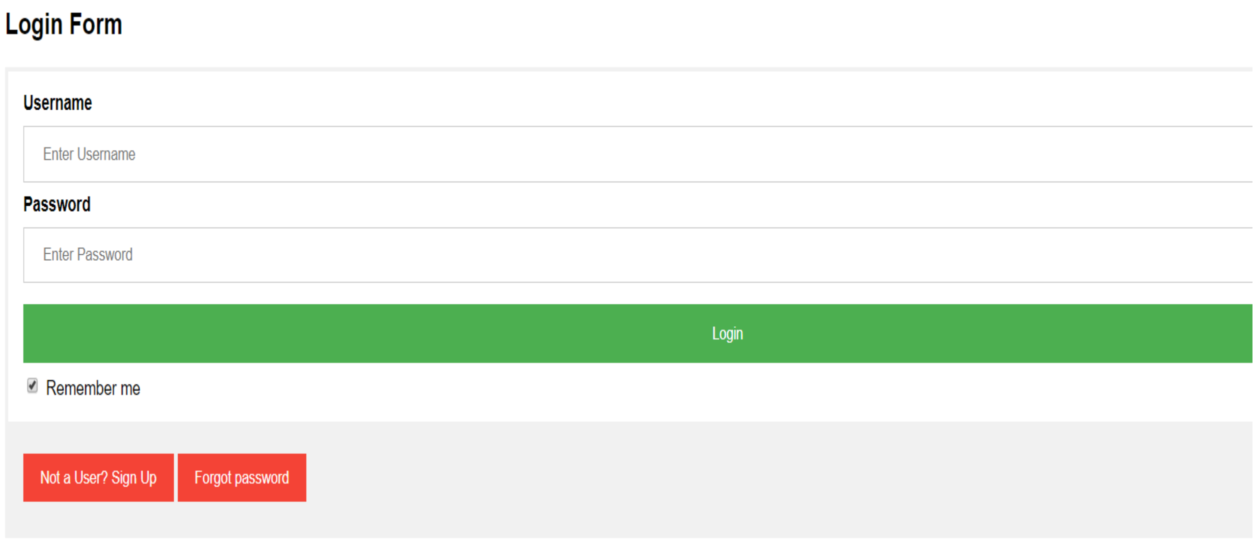
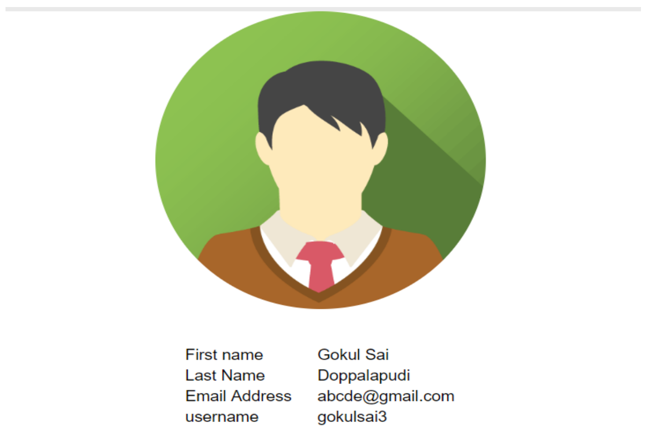


Figure 13-Login form



First name	Gokul Sai
Last Name	Doppalapudi
Email Address	abcde@gmail.com
username	gokulsai3

Figure 14-Home page

Fig 13 showing the login form of the user. The registered login details were stored in a database, after filling the user name nd password, click on submit button. After clicking submit, the login details will be verified from the database and if matched, redirects to the home page (fig 14). There will be a sign up form available for the new users who wants to register in vacation planner.

Fig 14 is the home page, which consists of the details of the user such as image ,first name, last name, email.id and the user name. Here there will be two options, one for planning the vacation and another for reviewing the previously planned vacation.

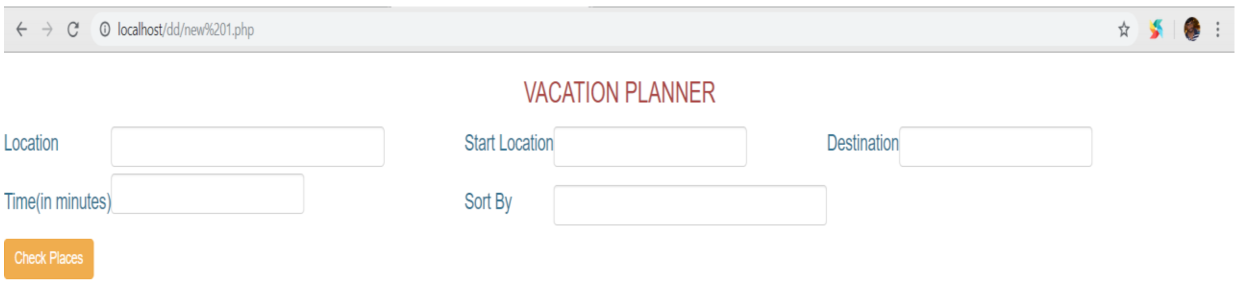
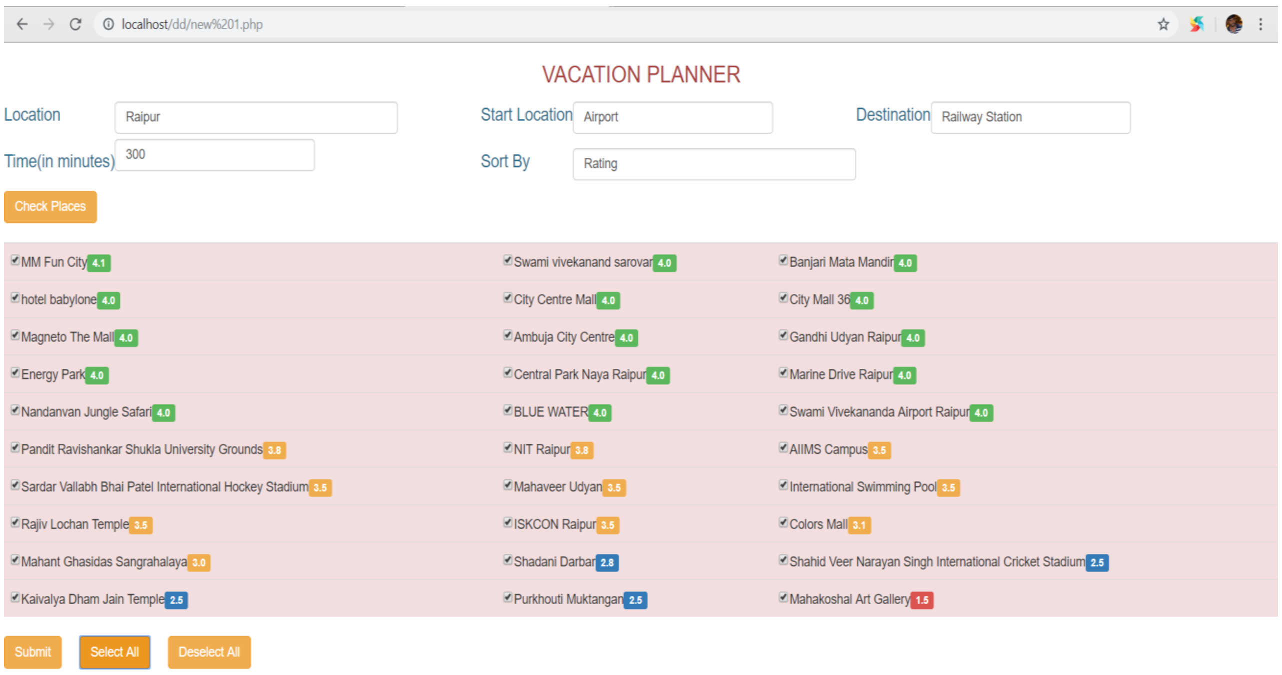


Figure 15-User Interface

Fig 15 represents the User interface of the visitor to select the places and location.This is a html page where user selects locations using check boxes and when clicks on submit button are submitted to server.

Fig 16 shows the page after the user enters the above mentioned 5 fields , and after clicking check places, user will get a chance to select the places he/she wish to visit in a city. This page consists of extra options like select all (to select all the places available) and deselect all (to clear all the check boxes). After clicking submit button the selected places will be sent to the server to find the path.

Fig 17 shows us the final list of selected places by the visitor he/she wishes to visit in the city, along with the latitude, longitude coordinates and the time taken to visit a particular place under the columns latitude, longitude and time respectively.



localhost/dd/new%201.php

VACATION PLANNER

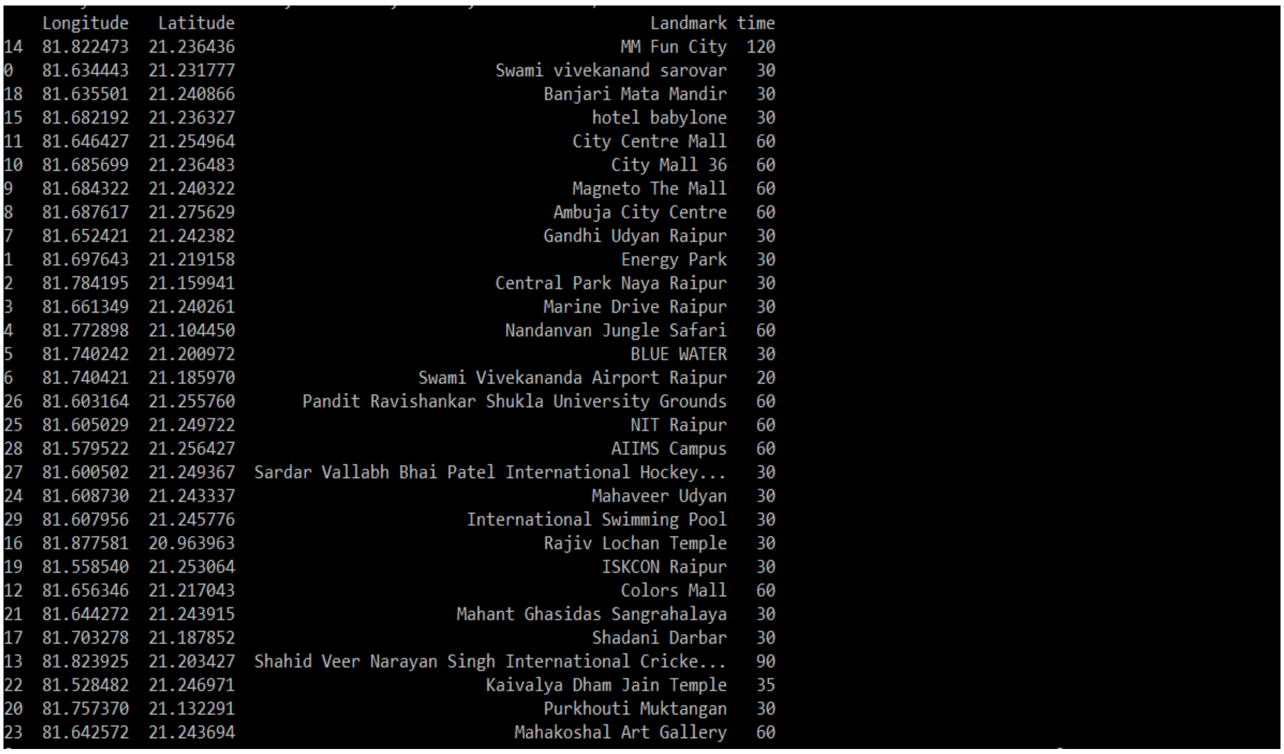
Location	Raipur	Start Location	Airport	Destination	Railway Station
Time(in minutes)	300	Sort By	Rating		

Check Places

✓ MM Fun City 4.1	✓ Swami vivekanand sarovar 4.0	✓ Banjari Mata Mandir 4.0
✓ hotel babylone 4.0	✓ City Centre Mall 4.0	✓ City Mall 36 4.0
✓ Magneto The Mall 4.0	✓ Ambuja City Centre 4.0	✓ Gandhi Udyan Raipur 4.0
✓ Energy Park 4.0	✓ Central Park Naya Raipur 4.0	✓ Marine Drive Raipur 4.0
✓ Nandanvan Jungle Safari 4.0	✓ BLUE WATER 4.0	✓ Swami Vivekananda Airport Raipur 4.0
✓ Pandit Ravishankar Shukla University Grounds 3.8	✓ NIT Raipur 3.8	✓ AIIMS Campus 3.5
✓ Sardar Vallabh Bhai Patel International Hockey Stadium 3.5	✓ Mahaveer Udyan 3.5	✓ International Swimming Pool 3.5
✓ Rajiv Lochan Temple 3.5	✓ ISKCON Raipur 3.5	✓ Colors Mall 3.1
✓ Mahant Ghasidas Sangrahalaya 3.0	✓ Shadani Darbar 2.8	✓ Shahid Veer Narayan Singh International Cricket Stadium 2.5
✓ Kaivalya Dham Jain Temple 2.5	✓ Purkhouti Muktangan 2.5	✓ Mahakoshal Art Gallery 1.5

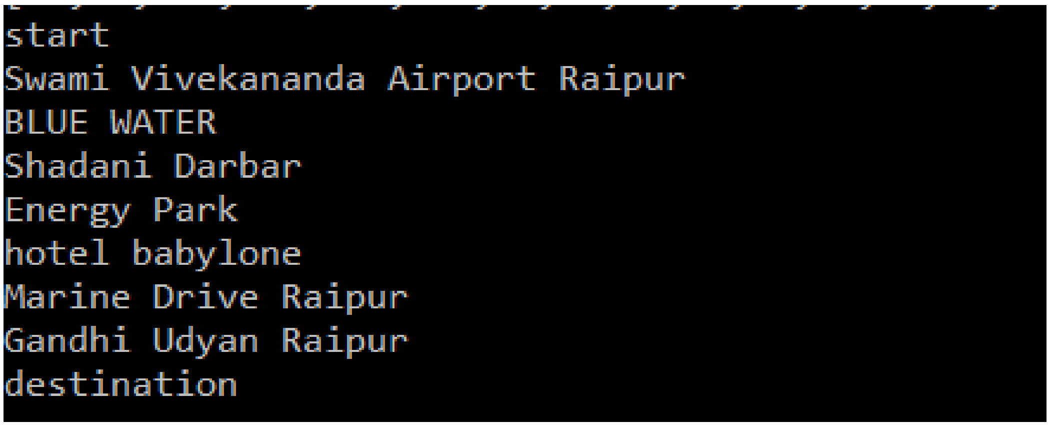
Submit Select All Deselect All

Figure 16- Places that are to be selected by the visitor



```
    Longitude   Latitude                                       Landmark  time
14  81.822473  21.236436                                    MM Fun City   120
0   81.634443  21.231777                        Swami vivekanand sarovar   30
18  81.635501  21.240866                             Banjari Mata Mandir   30
15  81.682192  21.236327                                  hotel babylone   30
11  81.646427  21.254964                                City Centre Mall   60
10  81.685699  21.236483                                    City Mall 36   60
9   81.684322  21.240322                                Magneto The Mall   60
8   81.687617  21.275629                              Ambuja City Centre   60
7   81.652421  21.242382                             Gandhi Udyan Raipur   30
1   81.697643  21.219158                                     Energy Park   30
2   81.784195  21.159941                        Central Park Naya Raipur   30
3   81.661349  21.240261                             Marine Drive Raipur   30
4   81.772898  21.104450                         Nandanvan Jungle Safari   60
5   81.740242  21.200972                                      BLUE WATER   30
6   81.740421  21.185970                  Swami Vivekananda Airport Raipur   20
26  81.603164  21.255760      Pandit Ravishankar Shukla University Grounds   60
25  81.605029  21.249722                                      NIT Raipur   60
28  81.579522  21.256427                                     AIIMS Campus   60
27  81.600502  21.249367  Sardar Vallabh Bhai Patel International Hockey...   30
24  81.608730  21.243337                                  Mahaveer Udyan   30
29  81.607956  21.245776                     International Swimming Pool   30
16  81.877581  20.963963                              Rajiv Lochan Temple   30
19  81.558540  21.253064                                    ISKCON Raipur   30
12  81.656346  21.217043                                     Colors Mall   60
21  81.644272  21.243915                     Mahant Ghasidas Sangrahalaya   30
17  81.703278  21.187852                                  Shadani Darbar   30
13  81.823925  21.203427  Shahid Veer Narayan Singh International Cricke...   90
22  81.528482  21.246971                       Kaivalya Dham Jain Temple   35
20  81.757370  21.132291                              Purkhouti Muktangan   30
23  81.642572  21.243694                           Mahakoshal Art Gallery   60
```

Figure 17- Places selected by the visitor.



```
start
Swami Vivekananda Airport Raipur
BLUE WATER
Shadani Darbar
Energy Park
hotel babylone
Marine Drive Raipur
Gandhi Udyan Raipur
destination
```

Figure 18-Path to be followed by the visitor

Fig 18 shows the names of the places among the selected places to be visited in order so as to cover maximum number of places within the given time.

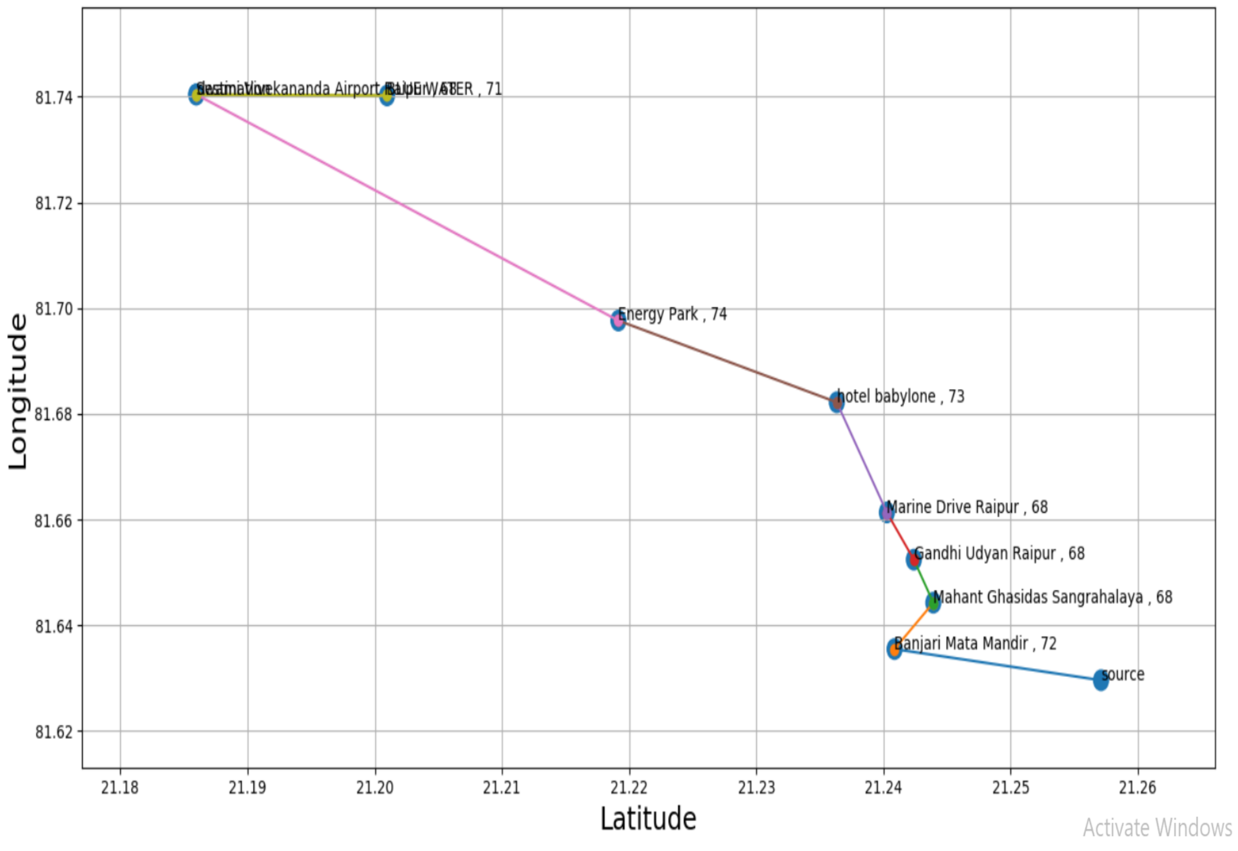


Figure 19- Plot showing the path for the places in fig.14

Fig 19 is showing the plot for the path the visitor has to visit to cover maximum number of places.

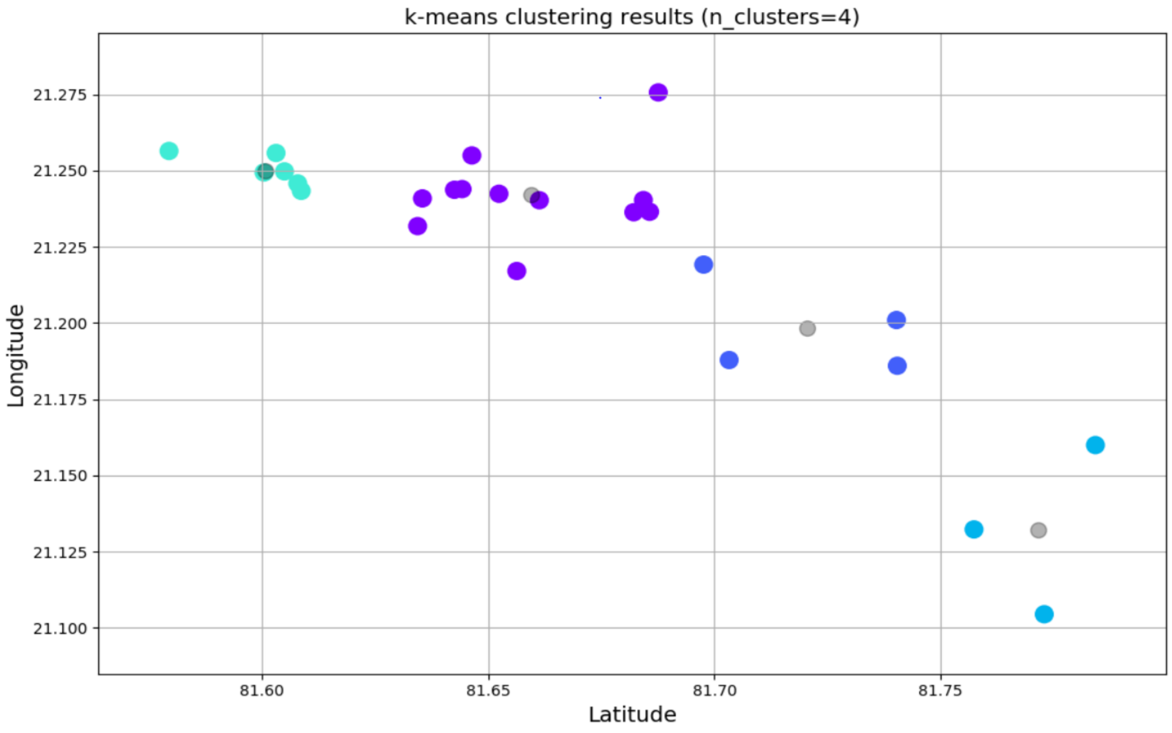


Figure 20-*k*-means clustering results (4 clusters)

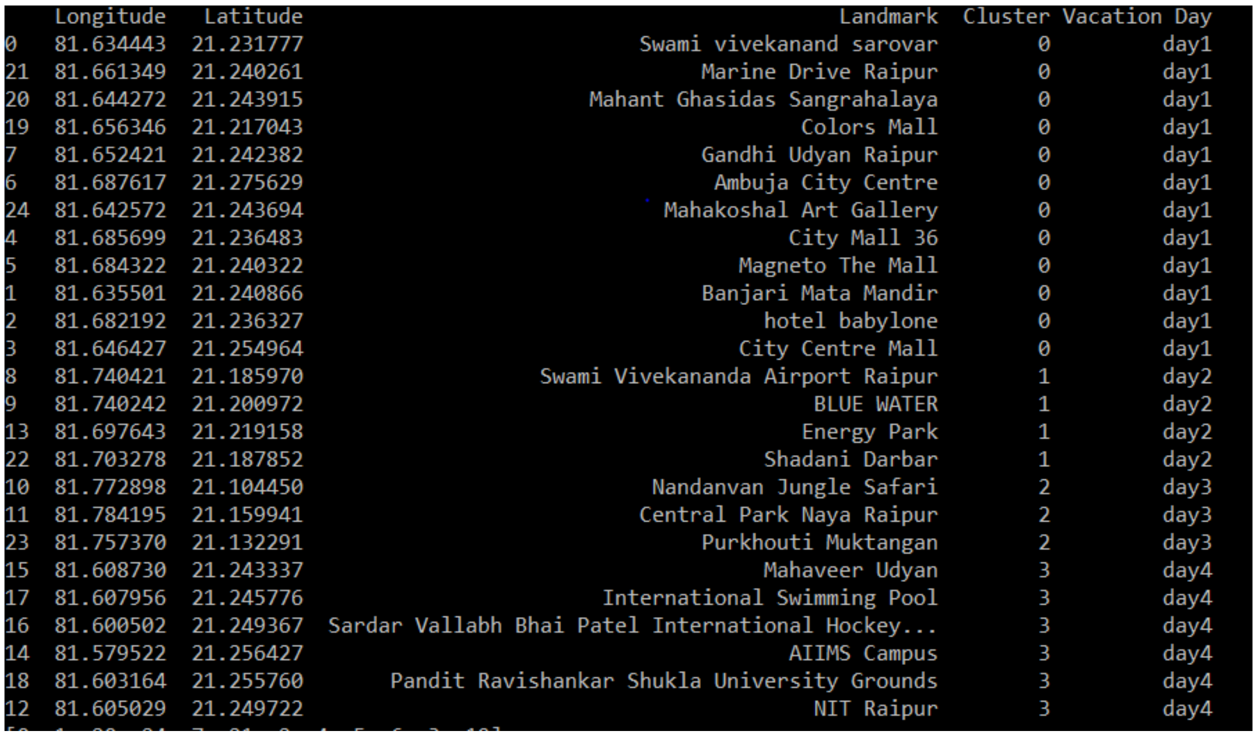

```
    Longitude   Latitude                                        Landmark  Cluster Vacation Day
0   81.634443  21.231777                           Swami vivekanand sarovar        0         day1
21  81.661349  21.240261                              Marine Drive Raipur        0         day1
20  81.644272  21.243915                       Mahant Ghasidas Sangrahalaya        0         day1
19  81.656346  21.217043                                       Colors Mall        0         day1
7   81.652421  21.242382                              Gandhi Udyan Raipur        0         day1
6   81.687617  21.275629                               Ambuja City Centre        0         day1
24  81.642572  21.243694                             Mahakoshal Art Gallery        0         day1
4   81.685699  21.236483                                      City Mall 36        0         day1
5   81.684322  21.240322                                  Magneto The Mall        0         day1
1   81.635501  21.240866                               Banjari Mata Mandir        0         day1
2   81.682192  21.236327                                     hotel babylone        0         day1
3   81.646427  21.254964                                  City Centre Mall        0         day1
8   81.740421  21.185970                    Swami Vivekananda Airport Raipur        1         day2
9   81.740242  21.200972                                        BLUE WATER        1         day2
13  81.697643  21.219158                                       Energy Park        1         day2
22  81.703278  21.187852                                    Shadani Darbar        1         day2
10  81.772898  21.104450                              Nandanvan Jungle Safari        2         day3
11  81.784195  21.159941                            Central Park Naya Raipur        2         day3
23  81.757370  21.132291                               Purkhouti Muktangan        2         day3
15  81.608730  21.243337                                     Mahaveer Udyan        3         day4
17  81.607956  21.245776                        International Swimming Pool        3         day4
16  81.600502  21.249367  Sardar Vallabh Bhai Patel International Hockey...        3         day4
14  81.579522  21.256427                                       AIIMS Campus        3         day4
18  81.603164  21.255760             Pandit Ravishankar Shukla University Grounds        3         day4
12  81.605029  21.249722                                         NIT Raipur        3         day4
```

Figure 21-Day wise plan for the selected places

Fig 20 showing the clusters that are formed after applying the k-means algorithm by taking the number of clusters as 4. K-means is applied to the places after removing the far away points (noise) from the selected set of places, the noise points are found using the HDBSCAN algorithm (noise means the points contained by the clusters with less than minimum number of points). Fig 21 is the day-wise plan assuming number of days equal to number of clusters taken and each cluster to be visited in a day (in any order but one cluster in a day)

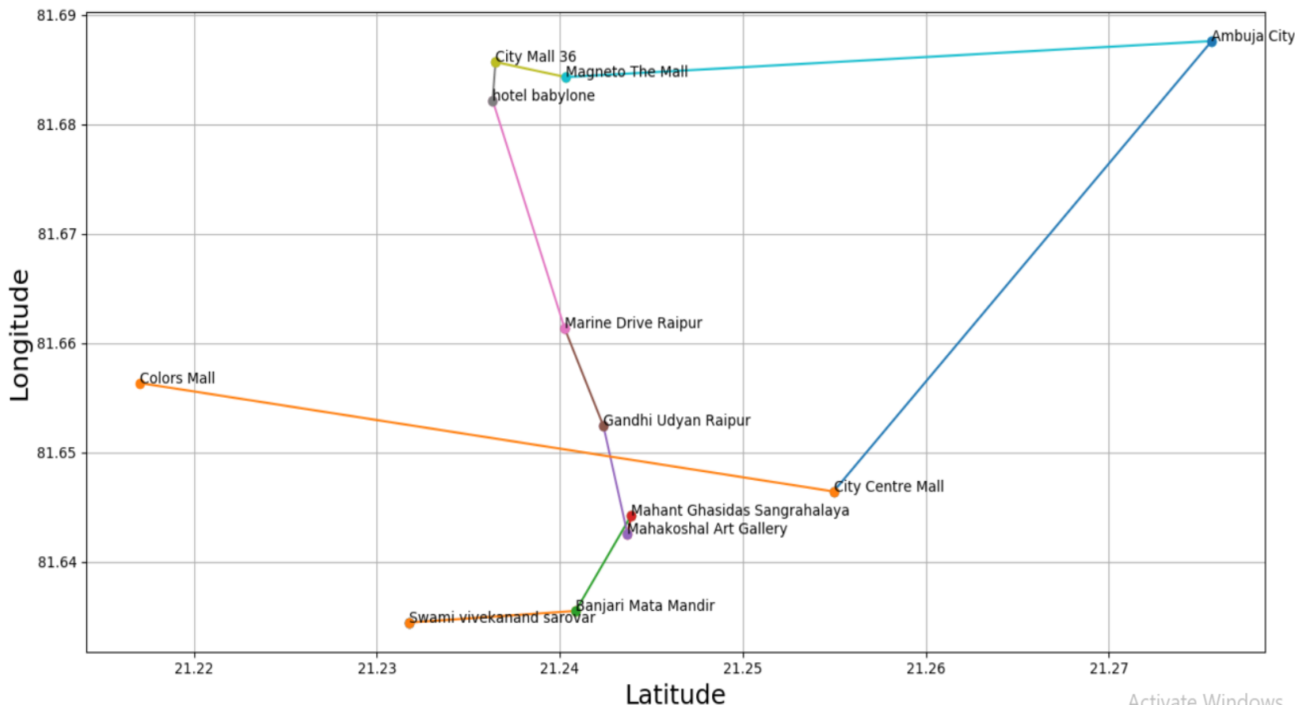
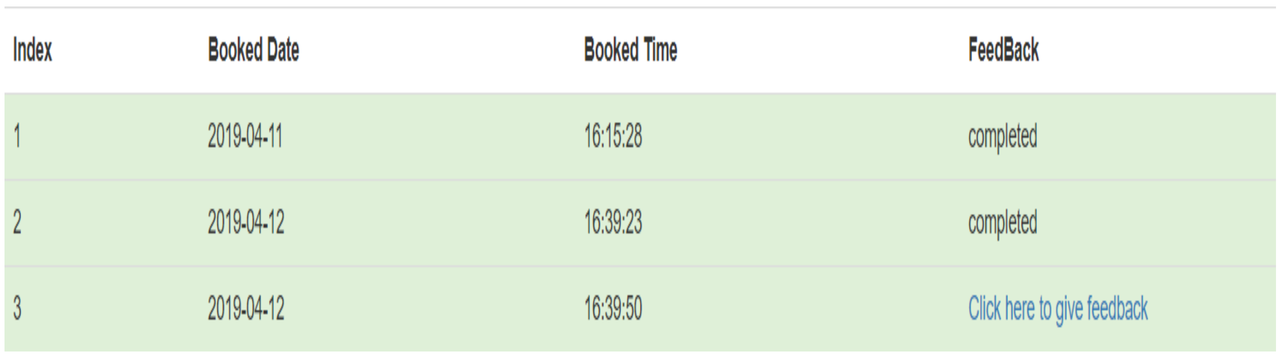


Figure 22-Route map for a single day

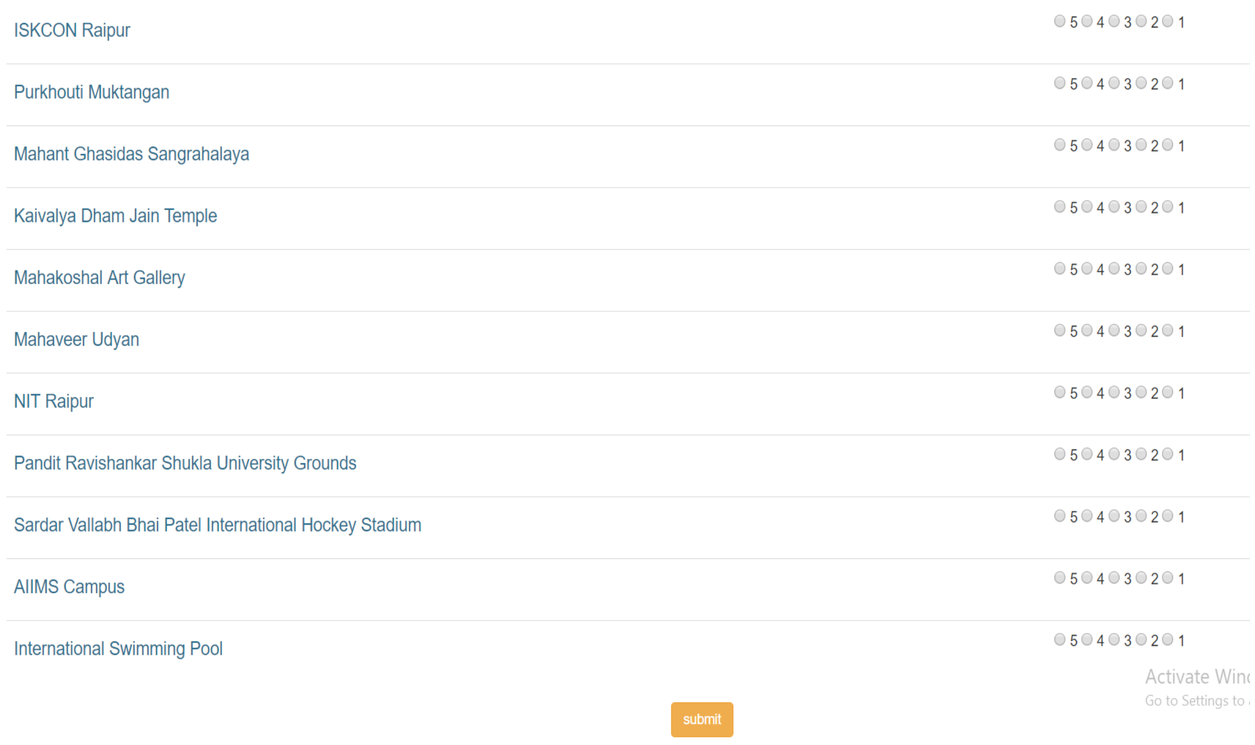
Fig 22 is the route plan for the user on a particular day of the trip. Starting at the Swami Vivekananda sarovar, ends at Colors mall. Here the source is taken randomly from the cluster and the next location on the route is the nearest location from the present location.



Index	Booked Date	Booked Time	FeedBack
1	2019-04-11	16:15:28	completed
2	2019-04-12	16:39:23	completed
3	2019-04-12	16:39:50	Click here to give feedback

Figure 23-Feedback page 1

Fig 23 is the feedback page. Whenever a user plans a vacation one new record (row) will be added in his account to give the feedback. It consists of time and date when the user plans a vacation and a feedback field with two values- completed and click here to review. Completed means user has given the feedback, the later means user needs to give the feedback. On clicking it page (fig 24) will be opened.



ISKCON Raipur	○ 5 ○ 4 ○ 3 ○ 2 ○ 1
Purkhouti Muktangan	○ 5 ○ 4 ○ 3 ○ 2 ○ 1
Mahant Ghasidas Sangrahalaya	○ 5 ○ 4 ○ 3 ○ 2 ○ 1
Kaivalya Dham Jain Temple	○ 5 ○ 4 ○ 3 ○ 2 ○ 1
Mahakoshal Art Gallery	○ 5 ○ 4 ○ 3 ○ 2 ○ 1
Mahaveer Udyan	○ 5 ○ 4 ○ 3 ○ 2 ○ 1
NIT Raipur	○ 5 ○ 4 ○ 3 ○ 2 ○ 1
Pandit Ravishankar Shukla University Grounds	○ 5 ○ 4 ○ 3 ○ 2 ○ 1
Sardar Vallabh Bhai Patel International Hockey Stadium	○ 5 ○ 4 ○ 3 ○ 2 ○ 1
AIIMS Campus	○ 5 ○ 4 ○ 3 ○ 2 ○ 1
International Swimming Pool	○ 5 ○ 4 ○ 3 ○ 2 ○ 1

Figure 24-feedback page 2

Fig 24 showing the user stars (ranging 1 to 5) to rate the places. The rating of the location will be updated based on the input given by the user considering mean (using number of users and previous rating) of ratings for a particular location.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6. Conclusion and future scope

6.1 Conclusion

In this project, we had successfully implemented to find the path between the source and the destination by covering maximum number of places in a given limited time.

6.2 Future enhancements

1. Bayesian average for Rating and reviewing.

In order to detect the fraud and fake reviews and ratings, it is preferable to use Bayesian average. If a same person is giving same rating many times, only one time it has to be considered.

2. Priority of choices to be given

Importance factor or Weightage factor for each location has to be taken into account to provide the user the more important places if the time constraint taken into account

3. Considering actual distance

In our project, we had calculated the displacement between the two places, which may vary from the actual distance. In order to calculate the actual distance we can use Google Maps API, which is useful for both distance finding and also for calculating time by taking traffic into consideration.

CHAPTER 7

REFERENCES

7. References:

[1] "Python Official site," [Online]. Available: https://www.python.org/.

[2] "NUMPY (Python library)," [Online]. Available: http://www.numpy.org/.

[3] Basic idea available on: https://towardsdatascience.com/using-unsupervised-learning-to-plan-a-paris-vacation-geo-location-clustering-d0337b4210de

[4] Sander, Jorg (1998). Generalized Density-Based Clustering for Spatial Data Mining. München: Herbert Utz Verlag. ISBN 3-89675-469-6

[5] X. Cui, T.E. Potok, P. Palathingal. "Document clustering using particle swarm optimization", Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005., 2005

[6] “HDBSCAN” https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html.

[7] “Matplotlib,”[Online]. Available: https://matplotlib.org.

[8] Martin Ester, Hans-Peter Kriegel, Joerg Sander, Xiaowei Xu (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Institute for Computer Science, University of Munich. Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining.

[9] Campello, Ricardo JGB, Davoud Moulavi, Arthur Zimek, and Jörg Sander. “A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies.” Data Mining and Knowledge Discovery 27, no. 3 (2013): 344-371.

[10] Campello, Ricardo JGB, Davoud Moulavi, Arthur Zimek, and Jörg Sander. “Hierarchical density estimates for data clustering, visualization, and outlier detection.” ACM Transactions on Knowledge Discovery from Data (TKDD) 10, no. 1 (2015): 5

[11] R. Campello, D. Moulavi, and J. Sander, *Density-Based Clustering Based on Hierarchical Density Estimates*

[12] McInnes L, Healy J. *Accelerated Hierarchical Density Based Clustering*

In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, pp 33-42. 2017 [pdf] <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=& arnumber=8215642>

[13] R.C.; Hennig, C. (2015). "Recovering the number of clusters in data sets with noise features using feature rescaling factors". Information Sciences. 324: 126–145.:aRXiV1602.06989. doi:10.1016/j.ins.2015.06.039

[14] Alon Vinnikov and Shai Shalev-Shwartz (2014). "K-means Recovers ICA Filters when Independent Components are Sparse" (PDF). Proc. Of Int'l Conf. Machine Learning (ICML 2014).

[15] D., Manning, Christopher (2008). Introduction to information retrieval. Raghavan, Prabhakar., Schütze, Hinrich. New York: Cambridge University Press. ISBN 978-0521865715. OCLC 190786122

[16] Mirkes, E.M. "K-means and k-medoids applet". Retrieved 2 January2016.

[17] Bradley, Paul S.; Fayyad, Usama M. (1998). "Refining Initial Points for k-Means Clustering". Proceedings of the Fifteenth International Conference on Machine Learning

[18] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii "Scalable K-means++" 2012 Proceedings of the VLDB Endowment.

[19] Chen, L.; Lv, M.; Chen, G. A system for destination and future route prediction based on trajectory mining. Pervasive Mob. Comput. 2010, 6, 657–676.

[20] https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm.

[21] Zhang, J.; Zhu, M.; Papadias, D.; Tao, Y.; Lee, D.L. Location-based spatial queries. In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, CA, USA, 10–12 June 2003.

[22] Zaïane, O.R.; Foss, A.; Lee, C.H.; Wang, W. On data clustering analysis: Scalability, constraints, and validation. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Taipei, Taiwan, 6–8 May 2002.

[23] Ragesh Jaiswal and Nitin Garg. Analysis of k-means++ for separable data. In Proceedings of the 16th International Workshop on Randomization and Computation, pp. 591?602, 2012

[24] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In Proceedings of the 18th annual ACM-SIAM symposium on Discrete Algorithms (SODA?07), pp. 1027-1035, 2007.

[25] Andrea Vattani. k-means requires exponentially many iterations even in the plane. In Proc. of the 25th ACM Symp. on Computational Geometry (SoCG), pages 324-332, 2009

[26] Arthur, David, Bodo Manthey, and H. Roglin. ”k-Means has polynomial smoothed complexity.” Foundations of Computer Science, 2009. FOCS’09. 50th Annual IEEE Symposium on. IEEE, 2009.

[27] Agarwal, Manu, Ragesh Jaiswal, and Arindam Pal.”k-means++ under Approximation Stability.” Theory and Applications of Models of Computation: 84.

[28] https://buildmedia.readthedocs.org/media/pdf/hdbscan/0.8.13/hdbscan.pdf

[29] N. Dalvi, R. Kumar, A. Machanavajjhala, and V. Rastogi. Sampling hidden objects using nearest-neighbor oracles. In SIGKDD, 2011.

[30] C. Xia, W. Hsu, M. L. Lee, and B. C. Ooi. Border: efficient computation of boundary points. Knowledge and Data Engineering, IEEE Transactions on, 18(3):289–303, 2006.

[31] G. Karypis, E.-H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. Computer, 1999.

[32] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In KDD, 1996.