Best Neighborhood To Live !!!

Report by Manoj Raghorte

Introduction :

It is imperative for every individual to use one's time, energy and money wisely. That's why there have been numerous applications and ideas which minimizes one's effort and still provide certainty of a accomplished idea. Keeping in mind the same ideology, we propose an application which has the capability to minimize every individual's effort while planning to live in a city or planning to stay on a holiday at particular location. The fundamental idea is, every individual would want to stay at a place from where one could have easy access to their location of interests. We want to provide a framework by which any individual would be able to find, how each neighborhood in a city if different from each other based on their accessibility to individual's provided interest venues.

We propose a framework which takes input from user about the places they want to explore, and find the best neighbourhood which has shortest access to most number of interested venues.There are certain assumptions we would like to establish before continuing. We assume that it is imperative for everyone to live near the area which has better access to most of one's interested places. The second assumption we made in this project is that, 'near' is equivalent to 2.5 kilometers because it will take hardly 5-7 minutes to reach from one's living place to desired venue location provided they take some transportation. However, the idea of term near differs from individual to individual. Because I am doing this project on Mumbai city, I chose 2.5 kilometers as it is considered near in us. The third assumption or rather change which we had to bear is about results of venues. Initial idea was to select the user's inputted subject venues based on rank, reviews or tips from Foursquare. We found that Foursquare application is not used in Mumbai city as widely as we thought. As a result, we did not find any venues with tips or reviews. So there was no easy way to rank explored results. Finally we decided to do our project on the first results we get from the query on Foursquare API and those results are capped to 50 venues.

The work we are doing in this project can be considered more of a statistical analysis rather than core machine learning analysis. We can extend the project very easily doing the same statistical analysis by involving machine learning models but currently we did not need that as the purpose of this project is very accurately can be predicted by exploratory statistical analysis.

Broader impact :

The proposed application can be utilized by all kinds of people who are finding either the permanent housing or have temporary stay in a city. Any user who has to decide where to live in a particular city and one has their own interests to pursue while living can use this application.To decide a scale, let's say it will be suitable for parents who wants to spend holiday while living.

near maximum number of amusement venues or a researcher who wants to be in an area where most of the knowledge centres are present or an artist who is looking for employment who to live near the entertainment centers are. From students searching for proper academic institutions to a foodie wanting to stay near area where most food attractions are accessible, this application could be utilized by everyone.

Real Estate agents can use this application to decide for their next construction place idea if they just find out what if the normal city individual wishes for in surrounding places while choosing their house permanently or while on holiday. Airbnb administrators can explore his idea based on their user's analysis of surrounding needs and set up their homes in the area which has maximum access to interested venues of general user. Finally apart from specific use cases, the geolocation providers like Google Maps, Apple Maps, or any other vendor can deploy a small utility application in their SDK's which will provide users where should they stay in city based on their preferences.

System architecture:

The System is divided into five major steps. Above illustration of each step will clarify their importance and order.

1. Web Extraction : We extract the neighborhoods of the user's provided city. As an example, I have extracted the neighborhoods of Mumbai Metropolitan city. We can generally do this task by using 'Beautifulsoup' API and giving the wikipedia page containing interested citi's neighborhood information. The extracted information may be encoded in xml or pure HTML format which will need clean up to get the names of the neighborhoods. Unwanted information is purified only to access the neighbourhood names and then converted into a dataframe. The resultant dataframe may also contain some values which are not useful while building our application so that is also discarded.

2. Get Neighborhoods locations : The data frame of neighborhoods we formed needs to be appended with their respective latitude and longitude information. We used the 'Geolocator API' to get these latitudes and longitudes. For this step, we can also use Foursquare API yet using Geolocator is convenient as we only want location information and nothing else. The resultant data frame consists of neighborhood names and their respective latitude and longitude values.

3. Venue Search Request using Foursquare : Based on the user's preference, one provides the query string to search for in venues. Here we use Foursquare API request by providing query string, client credentials, search radius and center point to begin search from. It is very subjective choice to set the center of search to begin with. As I am working on Mumbai Neighborhoods, I have set a place named 'Kurla' as center and provided it latitude and longitude to begin mapping the radius and begin our search. The

results we get from query request is in json format which needs to be parsed to get the relevant information. We extract names of venues, its latitudes and longitudes from the json data. Finally we make a data frame of extracted venue results for further analysis.

4. Visualizing Both dataframe results : The user would definitely want the application to be more of an intuitive and graphical rather than just statistics oriented. That's why we create a visualization of neighborhoods. First, we create a map of all neighborhoods in the city using Folium API which uses the neighborhoods latitude and longitude. Then we create another map of venues we generated using user's search query. Then we superimpose the two maps together which results in getting a pictorial view of how neighborhoods and venues generated are situated on the citi's map.

5. Statistical Analysis : Though we provide the user a pictorial presentation of how neighborhoods and venues one requested are situated, there is not a statistical proof in numbers which extends the proof we discovered through maps. So, we calculate the distance from each neighborhood to each venue generated and do an analysis on how many venues are situated in each neighborhood's near proximity area and what is their distance. We use the 'Harvensine Formula' to calculate the distance between each venue to each neighborhood. After further analysis, we calculate how many venues fall in the near proximity of each neighborhood. The near proximity or limit is assumed as 2.5 kilometers in this project which can be variable.

Technical details:

Our project details can be summarized in the following order:

1. Data Collection : Neighborhoods of provided citi's are collected through Wikipedia by web extraction using Beautifulsoup API which is further analyzed and purified for names of neighborhoods names which we have done on Mumbai Metropolitan city in our example.The geolocation values of neighborhoods as well venues is collected using Geolocator and Foursquare API which needed further refinement.

2. Data Annotation : The dataset that we create needs to be annotated separately based on our needs. Specially the extracted web information is purified and annotated which also happens with geolocation data extracted from foursquare API.

3. Pre-processing : In our project, after making the data frame of neighborhoods, we drop the neighborhoods which do not have longitude or latitude values generated through geolocator API. Further, the extracted data from search query is formalized by changing their names and reducing the features to important ones.

4. Haversine Formula : The **haversine formula** determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Important in navigation, it is a special case of a more general formula in spherical trigonometry, the **law of haversines**, that relates the sides and angles of spherical triangles. We make a distance matrix from all neighborhoods distance with all venues generated.

Experiments :

Apart from the superimposed maps generated we provide statistical representation of our results as follows.

Andheri	0
Bandra	4
Borivali	1
Dahisar	1
Goregaon	0
Jogeshwari	0
Juhu	0
Kandivali west	1
Khar	2
Malad	0
Santacruz	1
Vile Parle	0
Ghatkopar	3
Kanjurmarg	0
Kurla	6
Mulund	0
Powai	0
Vidyavihar	4
Vikhroli	0
Chembur	3
Govandi	1
Mankhurd	1
Trombay	1
Antop Hill	6
Byculla	7
Colaba	6
Dadar	**12**
Fort	8
Girgaon	**11**
Kalbadevi	**11**
Kamathipura	**11**
Matunga	8
Parel	10
Tardeo	10

Conclusion :

In this project, we extracted the neighborhoods of Mumbai city and generated their latitudes and longitudes using Geocoder API. Then we extracted the search query results given by the user for one’s interest of venues using Foursquare API and purified the data. Then we converted the data in pandas data frames. Using Folium API we create separate maps of neighborhoods and venues and then both maps are superimposed on each other. Finally we perform statistical analysis to build a distance matrix which provides how many venues fall in the near proximity of every neighborhood. Based on our result, user can identify which neighborhood is best suitable to live and most accessible to one’s suggested interested venues.