

Capstone Project - The Battle of Neighborhoods

Introduction / Business Problem:

I am an IT professional and I have been working in India for the past 15 years. Recently, I have been transferred to Pune City and I am in search of a location, which would be closer to my office but also should be well developed and having a lot of amenities. My office location is going to be Hinjawadi. Since the cost of the rented house was to be borne by my employer, the primary focus during search was on amenities around and the distance for transportation to the office.

After discussions with a few friends and a couple of searches on the web, it came to my understanding that there are two localities in Pune, which have been recently developed with modern amenities and the rate of development in these areas was fastest within the city. These areas were Aundh and Kothrud.

Therefore, I would primarily compare these two areas on two parameters.

- (1) Number of amenities around the locality and
- (2) Distance from my workplace

Target Audience: At present, the target audience for such problems is mainly me and my family members and also a group of persons from my company with their families, who are also relocated to Pune from other areas.

Data Section:

Datasets that would need to be referred to solve the above problem include ...

- (1) For understanding the amenities / venues in these two localities, we would be using **'Foursquare location data'** as a data source. Both these locations would be explored for various amenities or venues within a specified radial distance, say 5 kms through **API calls** placed on 'Foursquare'.

FourSquare is a local search-and-discovery app which uses location-specific data to help users find the best restaurants, bars, shops, entertainment, parks and nightlife in their area. When a user visits the location they can 'check-in', which tells their friends where they are, but they can also leave tips and reviews and even upload photos they've taken.

Young people put great trust in peer reviews, particularly those written by their friends, which is why Foursquare is such a popular resource that is currently used by an estimated 55 million consumers around the world.

I would be using the following request to explore the venues around desired localities.

GET '<https://api.foursquare.com/v2/venues/explore...>'.

This request returns a list of recommended venues near the current location. As response to this request following information is retrieved from Foursquare in the form of a 'json' file, which could be further fetched in a dataframe.

Field	Description
id	A unique string identifier for this venue.
name	The best known name for this venue.
location	An object containing lat & lng - latitude and longitude
categories	An array of categories that have been applied to this venue.

The number and types of categories could then be used to compare the two localities to make an informed decision while choosing any of them.

For Example: following data is fetched from 'Kothrud, Pune' location from Foursquare, fetched in a dataframe and first five records are displayed.

```
df.head()
```

	id	name	categories	lat	lng
0	0	Hidden Place - The Hangout	Pub	18.509107	73.812280
1	1	Cafe Coffee Day	Café	18.500140	73.814254
2	2	Barometer	Café	18.498824	73.819240
3	3	Endurance Fitness Club	Gym / Fitness Center	18.499235	73.821117
4	4	CCD	Café	18.507830	73.808498

```
df.shape
```

```
(186, 5)
```

- (2) Location coordinates of office locality and proposed localities would be derived from 'Geopy'.
- (3) For knowing the travel distance to my office, I would depend on 'Geopy'. Geopy can calculate '**Geodesic distance**' between two points using the function '`geopy.distance.distance`'. The geodesic distance is the shortest distance on the surface of an ellipsoidal model of the earth. Another method, '**Great-circle distance**' uses a spherical model of the earth, using the mean earth radius. However this method results in an error of up to about 0.5%.

Here is an **example** of `distance.distance` usage:

```
>>> from geopy import distance
>>> newport_ri = (41.49008, -71.312796)
>>> cleveland_oh = (41.499498, -81.695391)
>>> print(distance.distance(newport_ri, cleveland_oh).miles)
538.39044536
```