
COURSERA CAPSTONE PROJECT

IBM Applied Data Science

TITLE: Deciding on shifting to a new place in India

NAME: Tithi Chattopadhyay

1. INTRODUCTION

A 2019 report by United Nations speculates that by 2050, the urban population will have increased to 87.7 million and the rural population will account for 78.3 million people. In today's globally linked world, shifting from one place to another for a job, or work, or family is quite common. Every minute, 25-30 people are migrating to Indian cities from rural areas in search of better livelihood and lifestyles. Reasons and the types of migrants range, from a highly skilled IT engineer eyeing IT development centers to a farm labourer migrating to work at construction sites.

But, one other thing that's common is having the freedom to choose among a few locations, at the least. Often, we get stuck in places with none of the amenities we needed or wanted. It is, in particular the location to where one is shifting, the shops around it, safety of the area, that ensures one's quality of life.

2. BUSINESS PROBLEM:

When people shift to a new location in India, a number of things need to be looked into. To start off, although city life is said to be comparatively less peaceful than rural or outskirts areas, higher population often implies greater opportunity and commerce. Population growth is good for business, according to a number of studies including a 2016 study by Forbes. Hence, for this project, I assume that the shift is for work opportunities. Linked to the same thought is the fact that people prefer to have shops, stations and restaurants close to where they are aspiring to live in. And, perhaps most importantly, particularly for women, it is very important to take into account the safety of the city. So, keeping all of this in mind, which cities in India is most suitable for you?

3. DATA

The data for this project is mainly taken from a few particular sites, one of them being Wikipedia, but cross-validated from other sources, so as to ensure the accuracy.

a. CITIES IN INDIA:

The cities in India are scraped from LatLong.net using BeautifulSoup. There are 8 pages in total.

```
In [8]: x= ['https://www.latlong.net/category/cities-102-15.html', 'https://www.l
```



```
In [9]: from bs4 import BeautifulSoup
```

```
In [10]: n1=0
count=0
for i in range(0,8):
    URL= x[i]
    page = requests.get(URL)
    soup = BeautifulSoup(page.content, 'html.parser')
    a= soup.find('table')
    r= a.find_all('tr')
    for r1 in r:
        r2= r1.find_all('td')
        n2=0
        for r3 in r2:
            count=count+1
            arr[n1][n2]=str(r3)
            n2=n2+1
        n1=n1+1
```

The total number of cities are 794.

The names of the cities contained the state name, as well. So each city, the state it is in and the coordinates were all scraped.

b. STATE POPULATION

The Wikipedia page was scraped for finding the population density of each state and union territory in India.

```
In [30]: arr= np.full([40, 2], None)
URL= 'https://en.wikipedia.org/wiki/List_of_states_and_union_territories_'
page = requests.get(URL)
soup = BeautifulSoup(page.content, 'html.parser')
a= soup.find('table', class_='wikitable sortable')
```

```
In [ ]:
```

```
In [73]: r= a.find_all('tr')
```

```
In [ ]:
```

```
In [84]: n=0
for r1 in r:
    r2= r1.find_all('td')
    p=0
    for r3 in r2:
        if (p==1): arr[n][0]= str(r3)
        elif (p==10): arr[n][1]= str(r3)
        p+=1
    n+=1
```

Once I got the population, I deleted all the places in the states with very low population (<350/km²)

c. PLACES AROUND EACH CITY:

The FourSquare API Developer app was utilized to find all available places within radius 1000 of these cities.

```
In [107]: import folium

In [*]: CLIENT_ID = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX' # your FourSquare client ID
CLIENT_SECRET = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX' # your FourSquare client secret
VERSION = '20201129' # Foursquare API version
LIMIT = 10000 # A default Foursquare API limit value
radius= 1000
venues=[]

for lat, long, place in zip(mergedf["Latitude"], mergedf["Longitude"], mergedf["City"]):
    url= "https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&version={}&radius={}&lat={}&lon={}&limit={}"
    request= requests.get(url).json()["response"]["groups"][0]["items"]

    for venue in request:
        venues.append((place, lat, long, venue['venue']['name'], venue['venue']['location']['lat'], venue['venue']['location']['lon']))
```

Only those cities which have shops, restaurants or stations around them were kept, assuming that these would be the top priorities in the list of amenities needed closest to home.

d. CRIME RATES:

The Wikipedia page was parsed for the crime in each state and union territory. Since the values were not entirely reliable, the rankings of crime rate had to be used instead.

```
In [130]: URL= 'https://en.wikipedia.org/wiki/Indian_states_and_territories_ranked_by_crime_rate'
page= requests.get(URL)
soup= BeautifulSoup(page.content, 'html.parser')
```

```
In [131]: a= soup.find('table', {'class': 'wikitable sortable'})
r= a.find_all('tr')
arr= np.full([len(r), 2], None)
```

```
In [132]: n1=0
for r1 in r:
    r2= r1.find_all('td')
    n2=0
    for r3 in r2:
        if (n2==1):
            arr[n1][0]=str(r3)
        elif (n2==8):
            arr[n1][1]=str(r3)
        n2=n2+1
    n1=n1+1
```

METHODOLOGY

A number of different packages and tools were utilized in the making of this project

a. FOURSQUARE API

```
In [109]: CLIENT_ID = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX' # your Foursquare Client ID
CLIENT_SECRET = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX' # your Foursquare Client Secret
VERSION = '20201129' # Foursquare API version
LIMIT = 10000 # A default Foursquare API limit value
radius= 1000
venues=[]

for lat, long, place in zip(mergedf["Latitude"], mergedf["Longitude"], mergedf["Place"]):
    url= "https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&version={}&radius={}&limit={}&lat={}&lon={}"
    request= requests.get(url).json()["response"]["groups"][0]["items"]

    for venue in request:
        venues.append((place, lat, long, venue['venue']['name'], venue['venue']['location']['lat'], venue['venue']['location']['lon']))
```

The developer app was used, and the elimination of a number of cities with low population ensured that the number of calls were well within the permissible limit.

b. *FOLIUM*

```
In [191]: for lat, lan, name, ven, elev, r in zip(dffin['Latitude'], dffin['Longitu
for a in ven:
    elev= int(elev)
    if ("Restaurant" in a) and ("Station in a")):
        if elev in range(0,10): col = 'white'
        elif elev in range(10,20): col = 'pink'
        elif elev in range(20,35): col = 'orange'
        else: col = 'lightgreen'
    elif ("Restaurant" in a):
        if elev in range(0,10): col = 'red'
        elif elev in range(10,20): col = 'lightred'
        elif elev in range(20,35): col = 'beige'
        else: col = 'lightblue'
    elif ("Station" in a):
        if elev in range(0,10): col = 'lightgray'
        elif elev in range(10,20): col = 'green'
        elif elev in range(20,35): col = 'blue'
        else: col = 'cadetblue'
    else:
        if elev in range(0,10): col = 'black'
        elif elev in range(10,20): col = 'gray'
        elif elev in range(20,35): col = 'darkblue'
        else: col = 'darkgreen'

    if (r==2.0): c="cloud"
    elif (r==3.0): c= "home"
    else: c= 'info-sign'
    folium.Marker(location=[lat,lan],popup = name,
                    icon= folium.Icon(color=col,
                    icon_color='yellow',icon = c)).add_to(Map)
```

```
In [192]: Map
```

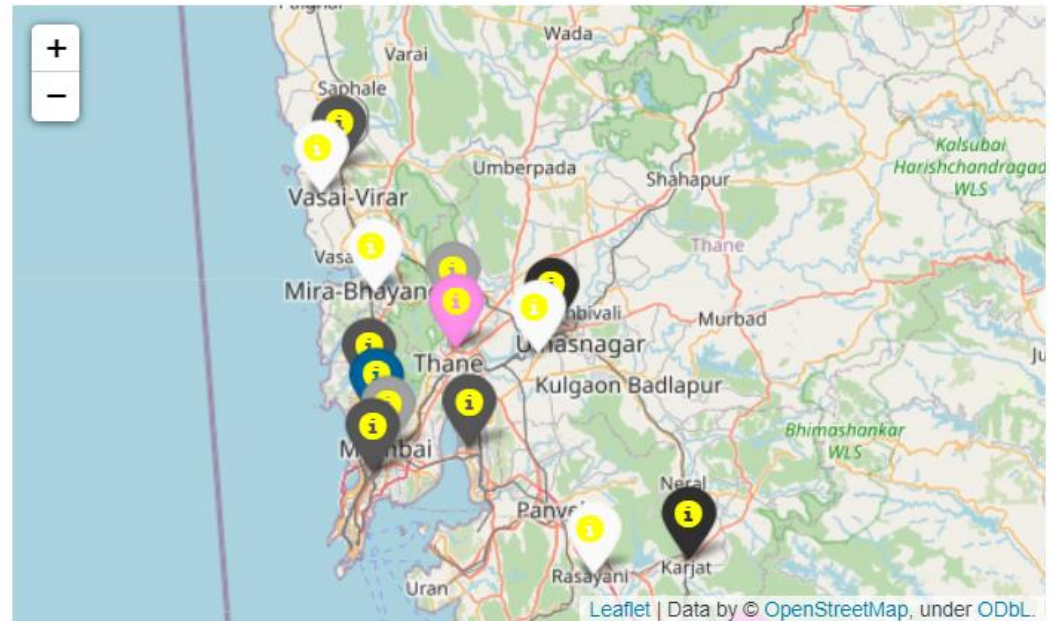
Folium was used to create a map of India, labelling all the cities according to the number of total restaurants, stations or shops nearby, with the icon on each pop-up bubble giving us an idea of the crime rate in the state. Since shops appeared in all the cities, it was assumed so while creating the colour coded markers

RESULTS:

The map thus created is interpretable by the user. If one is against living in a state with high crime rate, then one would ignore the cities with the ‘!’ sign on the pop-up. Similarly, if one doesn’t want to live in a place with only a few shops, one would ignore the black markers altogether.

```
In [207]: Map
```

```
Out[207]:
```



Note that the image is only of a small portion of the country, but gives an idea of what the user can expect to see once he/she zooms in on an area of the country.

CONCLUSION

As the world becomes more interlinked through the internet, we will expect to see more and more people being open to travelling and settling in other parts of the country. The world isn't what it used to be like. Fresh graduates and even experienced professionals hardly expect to work and live in the same city all their lives.

Although the issues that arise due to this direction of migration does cause a strain on the resources and land of the urban locations, and some professionals actually prefer to move away from cities to rural areas, to an average person looking for opportunities and higher standard of living, cities with higher population would be the destination.

Digressing from the parameters of this project, there can be numerous other considerations when a person shifts. For example, the availability of schools around, and so on. However, this project is just a starting point for such decisions.