The Battle of the Neighbourhoods:

A tale of three ancient capitals

Rahul Pradhan

Introduction

Nepal is a developing country which is much reliant on the means of tourism. Before the unification of Nepal, Kathmandu, Bhaktapur and Lalitpur were the most prominent places in Nepal. Each one has their own separate allure to it. These can be seen in its tradition and culture. The three places are quite the popular tourist and vacation destinations for people all around the world. They are diverse and multicultural and offer a wide variety of experiences that are widely sought after. We try to group the neighbourhoods respectively and draw insights to what they look like now. We try to compare and contrast the areas of interest in each place and find its diversity.

Problem

The aim is to help tourists choose their destinations depending on the experiences that the neighbourhoods have to offer and what they would want to have. This also helps people make decisions if they are thinking of creating any area of interest or business or even if they want to relocate neighbourhoods within the city. Our findings will help stakeholders make informed decisions and address any concerns they have including the different kinds of cuisines, provision stores and what the city has to offer.

Interest

The people of the country, the future business owners and researchers would find this report the most interesting. On a much further analysis people can find a relation between the number of areas of interest with the population density and the economy and land. Which would be considerate to find the proper step to take in order to increase the revenue of the land exponentially.

Data Acquisition and Cleaning

Data sources

All the data that are used in this report are obtained from

https://bhaktapurmun.gov.np/en/ward-profile

https://lalitpurmun.gov.np/en/ward-profile

https://old.kathmandu.gov.np/en/ward-profile

The scraped data from these were then combined into a csv file

https://github.com/unknown095/IDS-projects/blob/main/Book%20(1).csv

Geocoder library and google map were really helpful to find the longitude and latitude of the ward office and the city locations.

The data including venues, categories and other details were gathered from foursquare API.

Data Cleaning

Data error

1. Manual data entry errors

Humans are prone to making errors, and even a small data set that includes data entered manually by humans is likely to contain mistakes. Typos, data entered in the wrong field, missed entries and so on are virtually inevitable.

2. OCR errors

Machines can make mistakes when entering data, too. In cases where organizations must digitize large amounts of data quickly, they often rely on Optical Character Recognition, or OCR, technology to do so. OCR technology scans images and extracts text from them automatically. It can be very useful when, for example, you want to take thousands of addresses that are printed on paper and enter them into a digital database so you can analyze them using Hadoop. The problem with OCR is that it is almost always imperfect.

If you're OCR ing thousands of lines of text, you're almost certainly going to have some characters or words that are misinterpreted – zeroes that are interpreted as eights, for example, or proper nouns that are read as common words because the OCR tool fails to distinguish properly between capital and lowercase letters. The same sorts of issues arise with other types of automated machine entry of data, such as text-to-speech

3. Lack of complete information

When compiling a data set, you frequently run into the problem of not having all information available for every entry. For example, a database of addresses may be missing the zip codes for some entries because the zip codes couldn't be determined via the method that was used to compile the dataset.

4. Ambiguous data

When building a database, you may find that some of your data is ambiguous, leading to uncertainty about whether, how and where to enter it.

For example, if you are creating a database of phone numbers, some of the numbers you seek to enter may be longer than the typical ten digits that you have in a United States phone number. Are those longer numbers simply typos, or are they international phone numbers that include

more digits? In the latter case, does the number contain complete international dialing information?

These are the sorts of questions that are hard to answer quickly and systematically when you're working with a large body of data.

5. Duplicate data

You may find that two or more data entries are mostly or completely identical.

For example, maybe your database contains two entries for a John Smith living at 123 Main St. Based on this information, it's difficult to know whether these entries are simply duplicates (maybe John Smith's information was entered twice by mistake) or if there are two John Smiths (a father and son, perhaps) living at the same address. You need to sort out seemingly duplicate entries like this to make the best use of your data.

6. Data transformation errors

Converting data from one format to another can lead to mistakes.

As a simple example, you may have a spreadsheet that you convert to a comma-separated value, or CSV file. Because data fields inside CSV files are separated by commas, you may run into issues when performing this conversion in the event that some of the data entries in your spreadsheet contain commas inside them.

Unless your data conversion tools are sufficiently smart, they won't know the difference between a comma that is supposed to separate two data fields and one that is an internal part of a data entry. This is a basic example; things get much more complicated when you must perform complex data conversions, such as taking a mainframe database that was designed decades ago and converting it to NoSQL, a category of database that has become popular in just the last few years.

Solution

To help improve data quality, .isnull() and isna() were seen to be present using .sum(). TO remove the error .dropna() and replace(“”,0) is used to replace any extra space with zero.

```
In [5]: data1.isnull().sum()

Out[5]: Location                        1
        Ward                            0
        City                            0
        latitude                        1
        longitude                       1
        Population Density (per sq km)  0
        dtype: int64

In [6]: data1=data1.replace("",0)

In [7]: data1.isna().sum()

Out[7]: Location                        1
        Ward                            0
        City                            0
        latitude                        1
        longitude                       1
        Population Density (per sq km)  0
        dtype: int64

In [8]: data1=data1.dropna()

In [9]: data1.isnull().sum()

Out[9]: Location                        0
        Ward                            0
        City                            0
        latitude                        0
        longitude                       0
        Population Density (per sq km)  0
        dtype: int64

In [10]: data1.isna().sum()

Out[10]: Location                        0
         Ward                            0
         City                            0
         latitude                        0
         longitude                       0
         Population Density (per sq km)  0
         dtype: int64
```

Commands like .shape(), .info() and .describe() are consistently used to check for any data type issue.

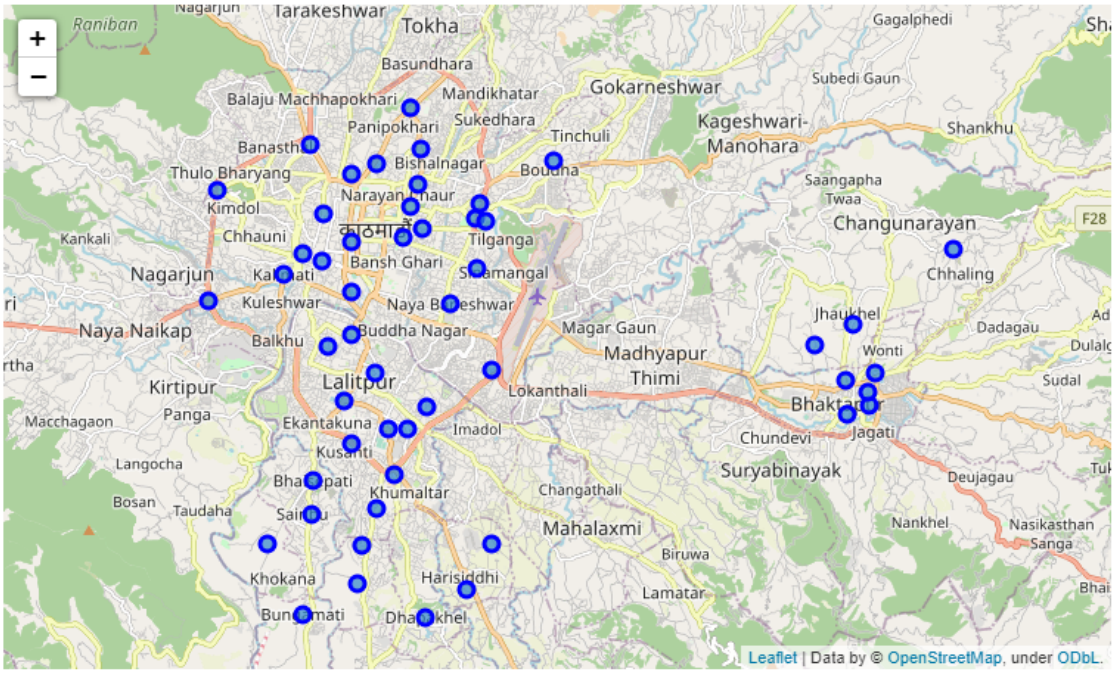
In [15]:
```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 51 entries, 0 to 52
Data columns (total 7 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Location                        51 non-null     object
 1   Ward                            51 non-null     object
 2   City                            51 non-null     object
 3   latitude                        51 non-null     float64
 4   longitude                       51 non-null     float64
 5   Population Density (per sq km)  51 non-null     int64
 6   Coordinates                     51 non-null     object
dtypes: float64(2), int64(1), object(4)
memory usage: 3.2+ KB
```

Feature Selection

The wards were marked in the map to verify their location before using the foursquare API.

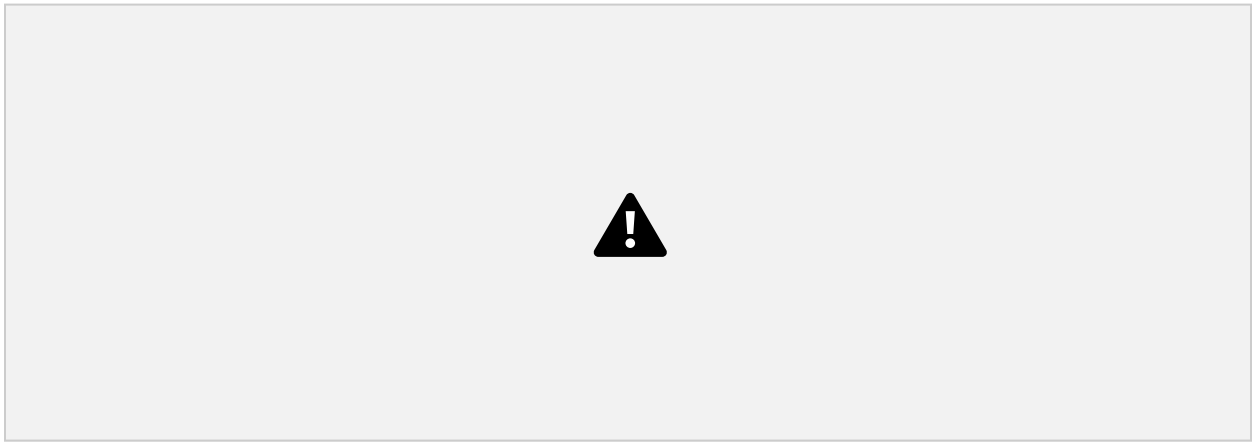
Out[18]:



Out[21]:

	Location	Ward	City	latitude	longitude	Population Density (per sq km)	Coordinates
0	Naxal	Ward 1	Kathmandu	27.712678	85.328703	5795	(27.7126782, 85.3287033)
1	Lazimpat	Ward 2	Kathmandu	27.721508	85.320765	16438	(27.7215082, 85.3207646)
2	Maharajgunj	Ward 3	Kathmandu	27.733079	85.328807	11300	(27.7330791, 85.3288072)
3	Baluwatar,	Ward 4	Kathmandu	27.724603	85.331017	14139	(27.7246028, 85.3310167)
4	Tangal	Ward 5	Kathmandu	27.717231	85.330449	23263	(27.7172307, 85.3304488)

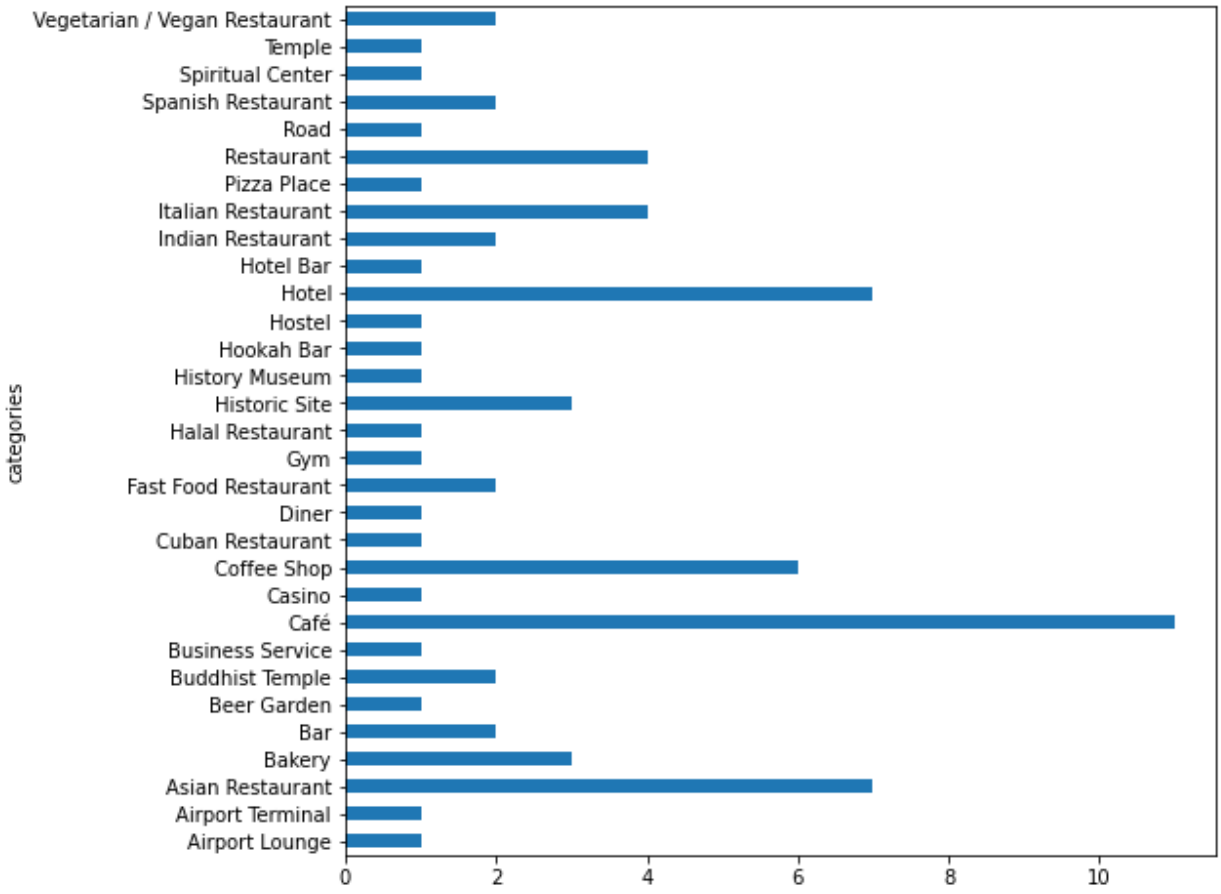
The following data was received after a data transaction with foursquare API.



Exploratory Data Analysis

Details on Kathmandu:

```
categories
Airport Lounge                  1
Airport Terminal                1
Asian Restaurant                7
Bakery                          3
Bar                             2
Beer Garden                     1
Buddhist Temple                 2
Business Service                1
Café                           11
Casino                          1
Coffee Shop                     6
Cuban Restaurant                1
Diner                           1
Fast Food Restaurant            2
Gym                             1
Halal Restaurant                1
Historic Site                   3
History Museum                  1
Hookah Bar                      1
Hostel                          1
Hotel                           7
Hotel Bar                       1
Indian Restaurant               2
Italian Restaurant              4
Pizza Place                     1
Restaurant                      4
Road                            1
Spanish Restaurant              2
Spiritual Center                1
Temple                          1
Vegetarian / Vegan Restaurant   2
dtype: int64
```

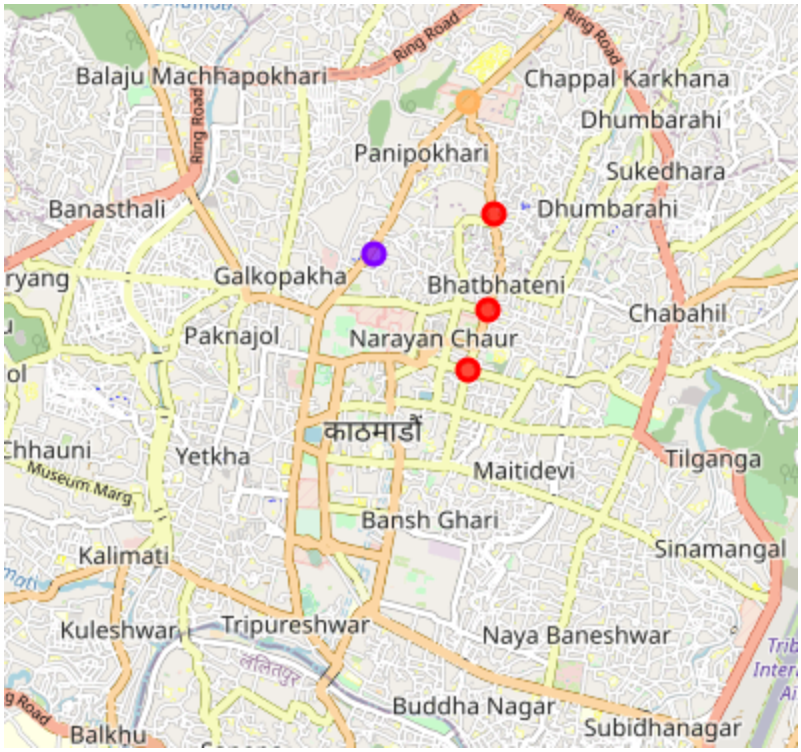


As per the details Cafe, Hotel and Asian Restaurant are the among the top three areas of interest with high numbers. There were a total of 74 counts of data with 31 unique values.

Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Baluwatar,	8	8	8	8	8	8
Bhimsengola	5	5	5	5	5	5
Bhimsenthan	17	17	17	17	17	17
Bouddha	20	20	20	20	20	20
Chhetrapati	28	28	28	28	28	28
Dillibazar	10	10	10	10	10	10
Gaushala	4	4	4	4	4	4
Gyaneshwor	4	4	4	4	4	4
Jayabageshwori	8	8	8	8	8	8
Kalanki	4	4	4	4	4	4
Kalimati	5	5	5	5	5	5
Koteshwor	3	3	3	3	3	3
Lainchaur	33	33	33	33	33	33
Lazimpat	23	23	23	23	23	23
Maharajgunj	3	3	3	3	3	3
Mitrapark	5	5	5	5	5	5
Naxal	7	7	7	7	7	7
New Baneshwor	3	3	3	3	3	3
Om Bahal	19	19	19	19	19	19
Purano Buspark	13	13	13	13	13	13
Sohrakhutte	1	1	1	1	1	1
Swayambhu	6	6	6	6	6	6
Tangal	20	20	20	20	20	20
Tripureshwor	12	12	12	12	12	12

Among more investigations, the most of the venues were centered around chhetrapati, Tangal and Bouddha.

The following clusters were found as provided in the map.

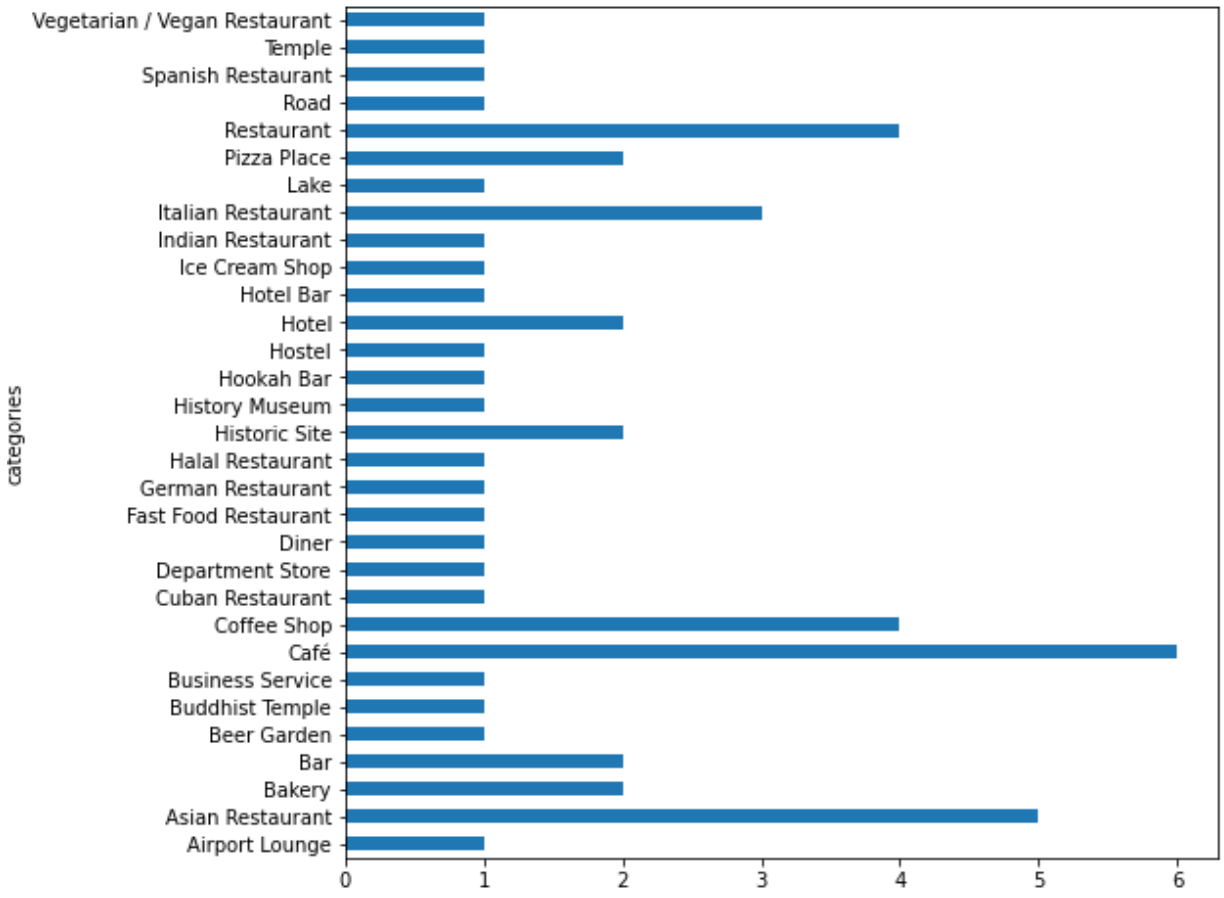


Cluster Label	Cluster color	Description about category
0	red	High diversity
1	purple	Medium diversity
4	pink	Low diversity

Details on Lalitpur

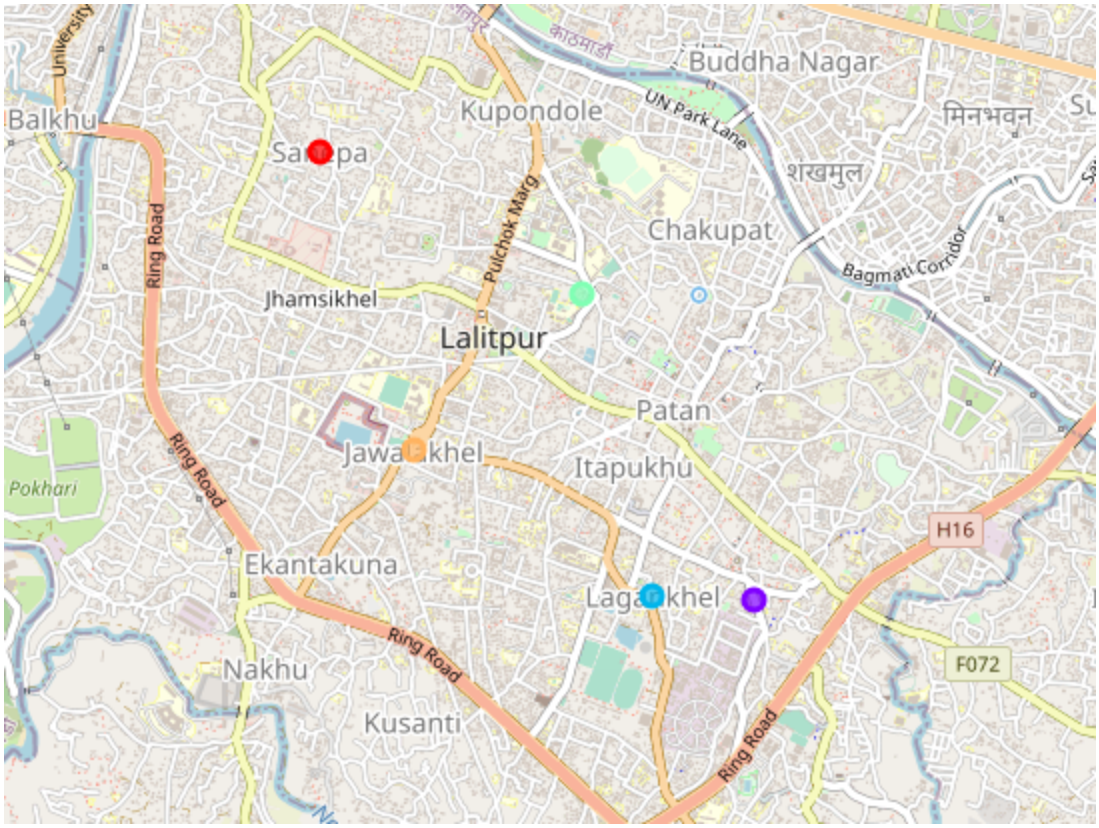
The category of area of interest is quite diverse with more inclination towards restaurants.With cafe,restaurant and coffee shop being on the highest frequency.

```
categories
Airport Lounge                  1
Asian Restaurant                5
Bakery                          2
Bar                             2
Beer Garden                     1
Buddhist Temple                 1
Business Service                1
Café                            6
Coffee Shop                     4
Cuban Restaurant                1
Department Store                1
Diner                           1
Fast Food Restaurant            1
German Restaurant               1
Halal Restaurant                1
Historic Site                   2
History Museum                  1
Hookah Bar                      1
Hostel                          1
Hotel                           2
Hotel Bar                       1
Ice Cream Shop                  1
Indian Restaurant               1
Italian Restaurant              3
Lake                            1
Pizza Place                     2
Restaurant                      4
Road                            1
Spanish Restaurant              1
Temple                          1
Vegetarian / Vegan Restaurant   1
dtype: int64
```



As seen on the table below Jawalakhel is the most diverse and populated area.

Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Jawalakhel	36	36	36	36	36	36
Kanibahal	4	4	4	4	4	4
Lagankhel	4	4	4	4	4	4
Pulchowk	11	11	11	11	11	11
Sanepa	11	11	11	11	11	11



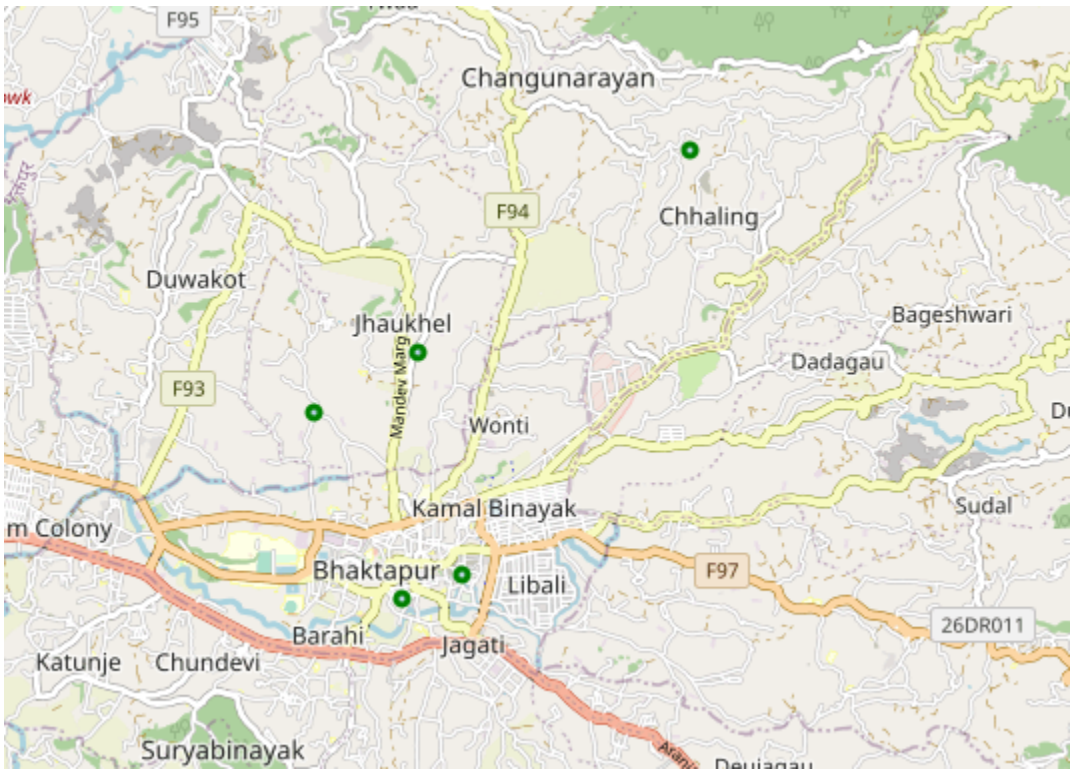
As seen on the map above the clusters available are:

Cluster label	Cluster color	description
0	red	Mix of restaurant and sports Medium diversity
1	turquoise	High diversity Restaurant based
2	purple	Restaurant based Low population density
3	Light blue	Restaurant based Low population density
4	orange	High diversity Low population density Restaurant based

Details on bhaktapur

The wards offices are marked on the below table and map.

	Neighbourhood	Ward	City	latitude	longitude	Population Density (per sq km)	Coordinates
0	mini Bus park	Ward 1	Bhaktapur	27.684014	85.422796	9901	(27.68401389, 85.42279647)
1	Duwakot	Ward 2	Bhaktapur	27.703937	85.455068	10553	(27.70393659, 85.4550683)
2	Nagarkot Road	Ward 4	Bhaktapur	27.688485	85.431673	8248	(27.68848453, 85.43167256)
3	Rammandir road	Ward 5	Bhaktapur	27.669910	85.430347	6425	(27.66991041, 85.43034742)
4	Incho - hanumanghat road	Ward 6	Bhaktapur	27.671735	85.435497	7212	(27.67173475, 85.43549726)



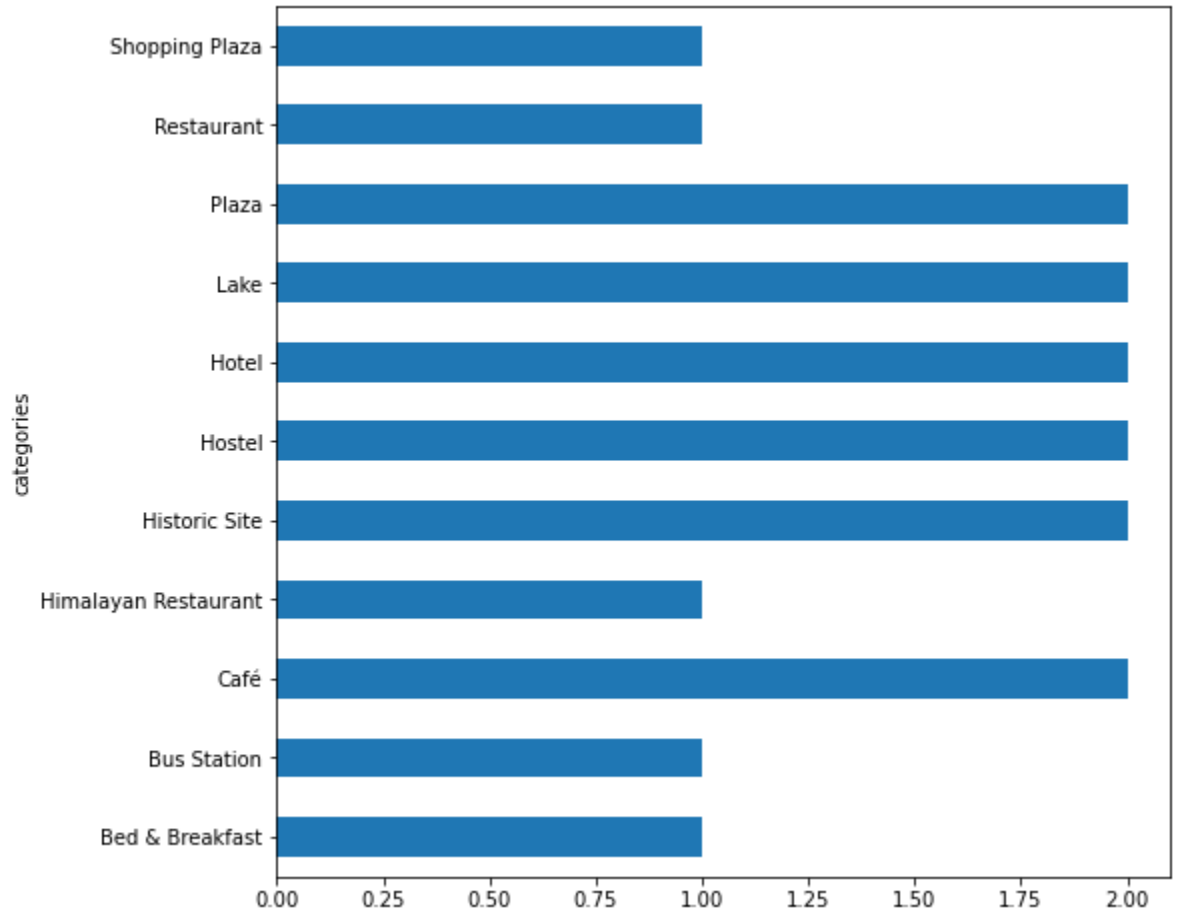
The data type is described and categories are counted. Compared to lalitpur and Kathmandu, Bhaktapur is quite less diverse as shown in the API data. Although known for its Historical architecture and culture, the restaurant and hotel are quite in demand.

```
nearby_venues['categories'].describe()
```

```
]: count       17
   unique      11
   top       Lake
   freq         2
   Name: categories, dtype: object
```

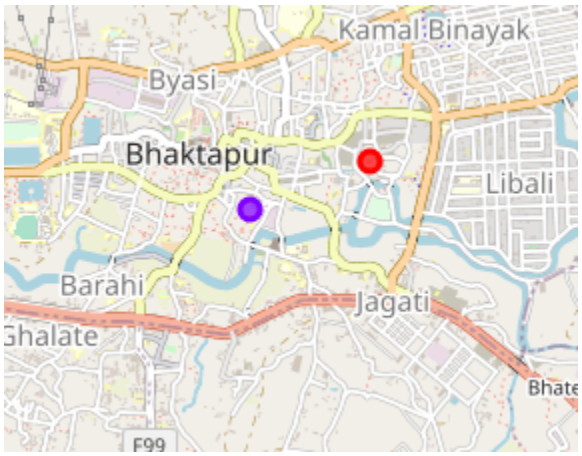
```
nearby_bha=pd.pivot_table(nearby_bha,columns=['categories'], aggfunc='size')
nearby_bha
```

```
]: categories
   Bed & Breakfast         1
   Bus Station             1
   Café                    2
   Himalayan Restaurant    1
   Historic Site           2
   Hostel                  2
   Hotel                   2
   Lake                    2
   Plaza                   2
   Restaurant              1
   Shopping Plaza          1
   dtype: int64
```



Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Incho - hanumanghat road	4	4	4	4	4	4
Rammandir road	14	14	14	14	14	14

Among the two major venues, ram mandir is to have quite the diversity with many restaurants, hotels and monuments.



As seen on the map there are two data clusters that were able to be calculated. The difference being obvious that the cluster 1 would be more diverse that cluster 0.

	Neighbourhood	Ward	City	latitude	longitude	Population Density (per sq km)	Coordinates	Cluster Labels	1st Most Common Venue	2nd Most Common Venue
3	Rammandir road	Ward 5	Bhaktapur	27.669910	85.430347	6425	(27.66991041, 85.43034742)	1	Plaza	Hostel
4	Incho - hanumanghat road	Ward 6	Bhaktapur	27.671735	85.435497	7212	(27.67173475, 85.43549726)	0	Restaurant	Market

Most populated

As per the population metrics Lalitpur and Kathmandu hold the most diverse and populated areas. As seen in the table below.

	Neighbourhood	Ward	City	latitude	longitude	Population Density (per sq km)	Coordinates
9	Kanibahal	Ward 6, 7 and 8	Lalitpur	27.666727	85.328007	89212	(27.6667273, 85.3280068)
14	Tangal	Ward 5	Kathmandu	27.717231	85.330449	23263	(27.7172307, 85.3304488)
5	Sanepa	Ward 2	Lalitpur	27.683772	85.309353	17148	(27.6837719, 85.309353)
11	Lazimpat	Ward 2	Kathmandu	27.721508	85.320765	16438	(27.7215082, 85.3207646)
13	Baluwatar,	Ward 4	Kathmandu	27.724603	85.331017	14139	(27.7246028, 85.3310167)

Limitations

1. The data are incorrectly input lowering data quality
2. Most of the actual data in bhaktapur have not been entered as there is not consciousness or awareness about it
3. Limited interaction with foursquare API as given for a free account.

Conclusions

In retrospect of the whole project, There are areas where the process can be improved like using borders instead of ranges. The API being fully integrated with the governmental database to give the correct and exact information. But in all this project would serve as a base project for realtors, business owners, migrating population, tourists and government administration.

References:

1. https://bhaktapurmun.gov.np/en/ward-profile
2. https://lalitpurmun.gov.np/en/ward-profile
3. https://old.kathmandu.gov.np/en/ward-profile
4. https://github.com/unknown095/IDS-projects/blob/main/Book%20(1).csv

5. Geocoder library
6. Google map
7. Foursquare API.

Python Library used:

1. folium
2. selenium
3. shapely
4. numpy
5. pandas
6. matplotlib
7. seaborn
8. geocoders
9. sklearn
10. json
11. requests
12. html
13. bs4