Neighbourhood Analysis for Immigrants

Author: Mahesh S.
LinkedIn: https://www.linkedin.com/in/maheshs3/

Objective

Help new immigrants to a city discover the right neighbourhoods to settle into, based on their preferences like Dining options, Cinemas, Entertainment, Parks, Schools, etc.

Who is the ideal User for this Project? How are they supposed to benefit from using it?

A User for this project would be any immigrant who is looking for neighbourhoods to move into. Each immigrant will have a unique set of needs based on their preferences and situation.

A married couple with a baby would need child-care centers and primary schools nearby, while a young bachelor would need restaurants and movie theatres nearby.

The user would log into this project and provide a list of “place types” that are of interest to them, where they would like to have places of these types in their vicinity.
The model would accept their input, and show them a list of neighbourhoods that have places belonging to the types they chose.

This way, the user will be shown a list of neighbourhoods that have a substantial presence of places pertaining to their interests.

How do we plan to implement this idea in code?

We can use Google Maps APIs to obtain the venues present in the city. Each Venue will have an associated Category to it(Ex:- Barbeque Nation, MTR will be assigned to Restaurant, etc.)

Then, divide the city into neighbourhoods(or localities), preferably based on PIN Codes.

We can then ask the User to select the Categories that are a priority to them. The model should then display the neighbourhoods that have an abundance of the categories chosen by the user. The model should also display the map of the city, indicating the different neighbourhoods in different colours.

Any Caveats we should be aware of?

Exploring the Google Maps APIs, I realised that it doesn't seem to be returning ALL the places in a given area.
This mostly has to do with the "type" of Developer account we choose to create. Premium or Paid accounts will have access to more places. A Basic or Unpaid account - like the one I used for this report - will have access to a limited number of places only.

As a result, the notebook I wrote for this project only generates a small sample of the places for each neighbourhood.

How will we obtain a list of "Neighbourhoods" and their Coordinates?

We can use Pin Codes as an efficient method of splitting Bengaluru into "neighbourhoods". We can then Google Search each Pin Code to get its coordinates.

We were able to obtain a total of 111 Pin Codes for areas in and around Bengaluru. However, people generally would not remember the Pin Codes for their areas, but would rather recognise the area by its name. So it's important to assign a name for each Pin Code as well. For this, we can go with Post Offices. Each Pin Code has a number of Post Offices in its jurisdiction. Each Post Office will in turn be named after the area it is in. We can simply show all the Post Offices associated with a Pin Code in a comma-separated list format.

The initial data would look something like this…

	Pincode	Latitude	Longitude	Post Office
0	560001	12.9766	77.5993	Bangalore Bazaar S.O, Bangalore G.P.O., Cubban...
1	560002	12.9635	77.5821	Bangalore City S.O, Bangalore Corporation Buil...
2	560003	13.0019	77.5713	Malleswaram S.O, Palace Guttahalli S.O, Swimmi...
3	560004	12.9438	77.5738	Basavanagudi H.O, Mavalli S.O, Pampamahakavi R...
4	560005	12.9980	77.6227	Fraser Town S.O

Each Pin code has multiple Post offices. We will display this data at the end along with the Places.

Finally, we will have a CSV of 111 rows and two columns. *Neighbourhood Name* and *Pin Code*.

Once we have the list of Neighbourhoods(Post Office names) and Pin codes, we are ready to start exploring Google Maps APIs to fetch the list of Places in each Pin code.

How will we fetch the Places data from Google Maps?

Understanding the Google Developer API websites

For complete beginners, navigating the website of Google Cloud/Developers can be intimidating! It can be hard to understand where to go, where to look, or even what a particular page is talking about!

So here is a simplified version of the process:-

Google provides a number of APIs for its products. Each of these APIs can be used to perform specific tasks or obtain specific information.
Example:- The Places API in GoogleMaps can be called to obtain information on the places present around a given location; the Directions API in GoogleMaps can be called to obtain Driving directions information between two input locations.

All that we need to do is to create a “Billing Account” and a “Project” in console.cloud.google.com. The former requires us to provide our Credit Card information, but we won’t be charged for a “Basic” account so no need to worry.
The Billing Account and Project will then be linked to each other. This way, Google knows that API calls made by this Project should be billed to this Billing Account. If the Billing Account is basic and the API call being made is Premium, Google will reject the API call’s request.

For “Basic” billing accounts, we are allowed to make API calls for upto 90 days, so we can’t sit around for too long after creating the account! Post that, Google will begin to reject our calls and we will need to upgrade to a premium account.

Then, we will need to generate an API Key for our Project. This is easy and can be done in a few clicks. We will specify this API Key in our code as a way of authenticating ourselves during the API calls.

Once we have successfully created the “Billing Account”, “Project”, “API Key” and linked the former two, we are ready to start making some API calls and work on the Design/Code!

Understanding Places API

We will primarily be using the Places API for our needs in this project. This API receives *a pair of coordinates* and *radius* as the input. It then returns information on the places present within the radius of the input location.
The output is usually returned as a JSON, which is processed as a Dictionary in Python. The returned dictionary has several nested keys and sub-dictionaries within it. We will only require certain parts of the dictionary which hold valuable information pertaining to our needs.

Here is a sample output from the Places API, Dictionary keys are written in **bold**:-

{**'html_attributions'**: [],
 'next_page_token': “random keys”,
 ‘Results’: [{**'geometry'**: {'location': {'lat': 12.9715987, 'lng': 77.5945627},
   'viewport': {'northeast': {'lat': 13.173706, 'lng': 77.8826809},
    'southwest': {'lat': 12.7342888, 'lng': 77.3791981}}},
  'icon': 'https://maps.gstatic.com/mapfiles/place_api/icons/v1/png_71/geocode-71.png',
  'name': 'Bengaluru',
  'photos': [{'height': 2340,
    'html_attributions': ['<a
href="https://maps.google.com/maps/contrib/107994729306954910118">Riju Pal'],
    'photo_reference':
'ATtYBwIDaZ84PuYRJvmPRbNJFDWFS_iVfAqVOT6OhHa9xrsL_66-Z6dKpFPZD1MFyQPvvF9
l7aUM7VrKw9fDEdG_7GQ-rKPDzITkz7V8v0YbnPOKBctSJC5SMgxNTIDuk1jH6j46wg68k4Xpe
wRzSrGVHqJlrs1u1jgIgsnk_8SjmsYZk2wt',
    'width': 4160}],
  'place_id': 'ChIJbU60yXAWrjsR4E9-UejD3_g',
  'reference': 'ChIJbU60yXAWrjsR4E9-UejD3_g',
  'scope': 'GOOGLE',
  'types': ['locality', 'political'],
  'vicinity': 'Bengaluru'},
{**'business_status'**: 'OPERATIONAL',
  'geometry': {'location': {'lat': 13.0410825, 'lng': 77.6243329},
   'viewport': {'northeast': {'lat': 13.0424112802915,
     'lng': 77.6257892802915},
    'southwest': {'lat': 13.0397133197085, 'lng': 77.6230913197085}}},
  'icon': 'https://maps.gstatic.com/mapfiles/place_api/icons/v1/png_71/restaurant-71.png',
  'name': 'Adyar Ananda Bhavan',
  'opening_hours': {'open_now': True},
  'photos': [{'height': 3456,
    'html_attributions': ['<a
href="https://maps.google.com/maps/contrib/105093897653620368683">Raj Arora'],
    'photo_reference':
'ATtYBwJME6lgIsQYcuHV5oj82HMQ7oKG_a2kh_WONIMExNTHc8dmehR8JtMQoQnxNVATlp

m6h6jyA-9LFhYdBPUsLTYhFaH-z8_RY_t5xvq79tkXYPDMxXpYX7pSmugq45rdbx_NvAqiMPE-H2XrN5VwpocTCe0iC9BDF50Az3AxyjTvvilC',
'width': 4608}],
'place_id': 'ChIJ13LvgGcXrjsRDyX-w-CUAtM',
'plus_code': {'compound_code': '2JRF+CP Bengaluru, Karnataka, India',
'global_code': '7J5V2JRF+CP'},
'price_level': 2,
'rating': 3.7,
'reference': 'ChIJ13LvgGcXrjsRDyX-w-CUAtM',
'scope': 'GOOGLE',
'types': ['restaurant',
'food',
'point_of_interest',
'store',
'establishment'],
'user_ratings_total': 5115,
'vicinity': '141, Thanisandra Main Road MS Ramaiah North City Manayata Tech Park Nagawara, signal, Bengaluru'}],
'status': 'OK'}

As we can see, we're mainly interested in the **Results** list. It contains a "list" of the Places that exist within the radius of the input coordinates. Each of these "places/items" will then have it's own nested dictionary of information.
However, not every place will have the same set of information. **Example:-** The **'business_status'** item is present only for the second place in the above sample output.

Within the results list, we're mainly interested in the name, types, rating, and vicinity.
The above four pieces of data should give us enough information to share with the user.

Taking User Input and Filtering the Data

Once we have called the Places API, obtained the data and formatted it to the way we need it - We will need to ask the user for their input on what "place types" are they interested in.

We can display the list of "place types" available, and request the user to select the ones of interest to them. We will also need to check that the user's input matches the types available to us(user says "banks" while the place type is "bank" in the model), if not, we can simply ask the user to input again.

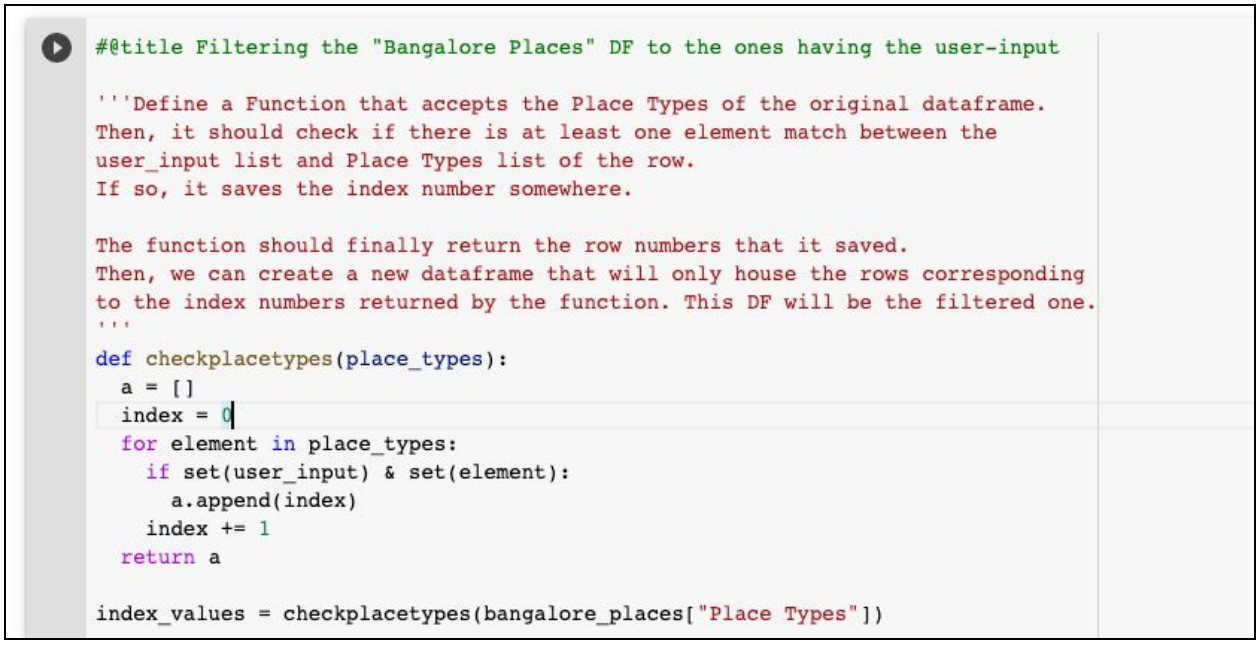
This way, we'll have the "place types" that we'll need to filter our data to.

The code that asks the user for their input, and checks to see if their input matches the "place types" in our data.

We will then need to analyse our data to find the right neighbourhoods for the user, i.e., the ones that have a high volume of places matching the user-input types.
As we can see from the Dictionary structure above, the **types** key holds a List consisting of the place types that that place belongs to.

A place can have multiple types associated with it. A "restaurant" can also be a "point of interest" or "tourist" if it is popular with tourists or is considered as a famous landmark in the city/neighbourhood.

We can consider that a particular place matches the users criteria if at least one of the types mentioned in its "types" dictionary belongs to the list of types selected by the user.



The code to filter the original data to only the rows that have a place type of interest to the user.

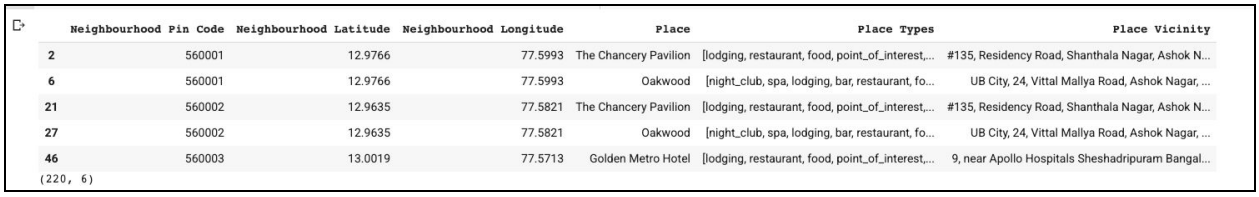
This way, we will filter the original data(with all the places and neighbourhoods) to only those places that match the user's criteria.

Data Manipulation

We know that every place listed in this filtered data will be of interest to the user as they belong to at least one of the types selected by the user.

We plan to show aggregate functions like Count of places belonging to each type per neighbourhood. However, we face a constraint in doing this due to the structure of the data. Note that the **Types** key is a list in the original dictionary. So it would be difficult to perform aggregate operations like *Count of places of a particular type*.

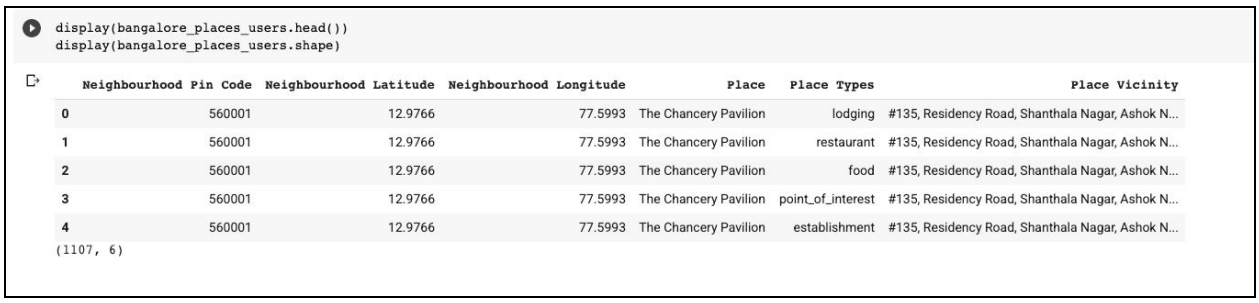
In order to overcome this issue, we will need to “Unnest” the types column. In other words, if a particular row shows a place having four different types to it: We must unnest that row into four separate rows where all four will have the same place, but with different types.



	Neighbourhood Pin Code	Neighbourhood Latitude	Neighbourhood Longitude	Place	Place Types	Place Vicinity
2	560001	12.9766	77.5993	The Chancery Pavilion	[lodging, restaurant, food, point_of_interest,...	#135, Residency Road, Shanthala Nagar, Ashok N...
6	560001	12.9766	77.5993	Oakwood	[night_club, spa, lodging, bar, restaurant, fo...	UB City, 24, Vittal Mallya Road, Ashok Nagar, ...
21	560002	12.9635	77.5821	The Chancery Pavilion	[lodging, restaurant, food, point_of_interest,...	#135, Residency Road, Shanthala Nagar, Ashok N...
27	560002	12.9635	77.5821	Oakwood	[night_club, spa, lodging, bar, restaurant, fo...	UB City, 24, Vittal Mallya Road, Ashok Nagar, ...
46	560003	13.0019	77.5713	Golden Metro Hotel	[lodging, restaurant, food, point_of_interest,...	9, near Apollo Hospitals Sheshadripuram Bangal...

(220, 6)

This is how the data looks like Before Unnesting. Note that the “Place Types” column is a list of multiple values.



```
display(bangalore_places_users.head())
display(bangalore_places_users.shape)
```

	Neighbourhood Pin Code	Neighbourhood Latitude	Neighbourhood Longitude	Place	Place Types	Place Vicinity
0	560001	12.9766	77.5993	The Chancery Pavilion	lodging	#135, Residency Road, Shanthala Nagar, Ashok N...
1	560001	12.9766	77.5993	The Chancery Pavilion	restaurant	#135, Residency Road, Shanthala Nagar, Ashok N...
2	560001	12.9766	77.5993	The Chancery Pavilion	food	#135, Residency Road, Shanthala Nagar, Ashok N...
3	560001	12.9766	77.5993	The Chancery Pavilion	point_of_interest	#135, Residency Road, Shanthala Nagar, Ashok N...
4	560001	12.9766	77.5993	The Chancery Pavilion	establishment	#135, Residency Road, Shanthala Nagar, Ashok N...

(1107, 6)

This is how the data looks like After Unnesting. We now have one row per Place Type. So a row with multiple Place Types would be split into multiple rows.

This way, the size of our data will increase as we are unnesting individual rows into multiple rows. However, since we have already removed the unwanted rows(the ones corresponding to places that the user isn’t interested in), the overall size mustn’t be too large here.

The main reason for splitting the rows is to be able to perform aggregation functions in order to display a “report” to the user indicating which neighbourhood has what quantity of places of a particular type.

Remember earlier that we had split the city into Neighbourhoods based on the Pin Codes? Remember that we had used Post Offices as a means to give names to each Pin codes? It is now time to join those Post Office names to our data. Basically, the final data will look just like the table above - but with the added column of “Post Office”.

	Pincode	Latitude	Longitude	Place	Place Types	Place Vicinity	Post Office
0	560004	12.9438	77.5738	Al-Ameen College Of Pharmacy	university	Opp. Lalbagh Main Gate, Hosur Road, Bengaluru	Basavanagudi H.O, Mavalli S.O, Pampamahakavi R...
1	560004	12.9438	77.5738	Al-Ameen College Of Pharmacy	point_of_interest	Opp. Lalbagh Main Gate, Hosur Road, Bengaluru	Basavanagudi H.O, Mavalli S.O, Pampamahakavi R...
2	560004	12.9438	77.5738	Al-Ameen College Of Pharmacy	establishment	Opp. Lalbagh Main Gate, Hosur Road, Bengaluru	Basavanagudi H.O, Mavalli S.O, Pampamahakavi R...
3	560004	12.9438	77.5738	Annamalai University	university	1st floor, near Vijaya College,IOC Petrol pump...	Basavanagudi H.O, Mavalli S.O, Pampamahakavi R...
4	560004	12.9438	77.5738	Annamalai University	point_of_interest	1st floor, near Vijaya College,IOC Petrol pump...	Basavanagudi H.O, Mavalli S.O, Pampamahakavi R...
...	...	...	...	...	...	...	...
230	562120	12.8788	77.1715	CHETHAN PROVISION STORE	point_of_interest	Mathikere	Chamarajasagara B.O
231	562120	12.8788	77.1715	CHETHAN PROVISION STORE	establishment	Mathikere	Chamarajasagara B.O
232	562157	13.1726	77.6307	SRI VENKATESWARA COLLEGE OF ENGINEERING	university	Kempegowda International Airport Bengaluru, Ro...	Bettahalsur S.O, Chikkajala B.O, Doddajala B.O...
233	562157	13.1726	77.6307	SRI VENKATESWARA COLLEGE OF ENGINEERING	point_of_interest	Kempegowda International Airport Bengaluru, Ro...	Bettahalsur S.O, Chikkajala B.O, Doddajala B.O...
234	562157	13.1726	77.6307	SRI VENKATESWARA COLLEGE OF ENGINEERING	establishment	Kempegowda International Airport Bengaluru, Ro...	Bettahalsur S.O, Chikkajala B.O, Doddajala B.O...

235 rows × 7 columns

We now have all the data we need in a single table! We have the Pincode, Post offices, Places and their types!

We can now delete all the other “tables” we’d created up until now! The above table is all we’ll need!

We can now display this data to the user to show them the neighbourhoods that should be of interest to them!

Showing a Table of the User’s selected Neighbourhoods

We now take the final table(generated at the end of the previous section) and manipulate it to show the rows displaying the places of interest to the user.

```
======================================
We have successfully filtered all the Neighbourhoods and Places in the city
to those of interest as per your input.

Here, we are showing a distribution of the Places by Types -
incl. those not selected by you.
This way, you can explore other Types of Places available in
the Beighbourhoods that also have Places of your interest.
======================================
```

Place Types	Pincode
establishment	27
point_of_interest	27
food	13
lodging	13
restaurant	13
bar	11
night_club	11
movie_theater	10
spa	9
shopping_mall	6
cafe	2
store	2

Showing a count of places present in the Neighbourhoods that contain Places that are of interest to the user

This doesn't mean the user chose every single "type" on display here. After we filter the neighbourhoods to the ones that have at least one place of the type selected by the user, we will proceed to show the count of places for the other types as well.
This is to provide the user with an idea of what other places are present in their selected neighbourhoods.

```
======================================
Here, we are showing a Count of the Places at a
(Neighbourhood x Types) level
======================================
```

Post Office	Pincode	Place Types	Place
CMP Centre And School S.O, Museum Road S.O	560025	night_club	2
Richmond Town S.O	560025	lodging	2
CMP Centre And School S.O, Museum Road S.O	560025	establishment	2
		point_of_interest	2
		restaurant	2
		spa	2
Richmond Town S.O	560025	spa	2
		restaurant	2
		point_of_interest	2
		night_club	2
		food	2
CMP Centre And School S.O, Museum Road S.O	560025	food	2
Richmond Town S.O	560025	establishment	2
		bar	2
CMP Centre And School S.O, Museum Road S.O	560025	bar	2
		lodging	2
Kanteeravanagar S.O, Nandinilayout S.O	560096	night_club	1
		point_of_interest	1
		restaurant	1

A (Post Office x Type) distribution of the Places in the filtered Neighbourhoods.

For every Neighbourhood(ID'ed by Post Office or Pin code) and Place Types - how many places of that Type are present in that Neighbourhood?

This is the data that'll show the user the Neighbourhoods and how much of each Place Type it has.

In the next section, we will also show how we present this data in a map for better visualisation.

Displaying the Neighbourhoods to the User

We also want to show a map of the city along with their pin codes, and the specific pin codes that the user chose based on their inputs for preferred “place types”.

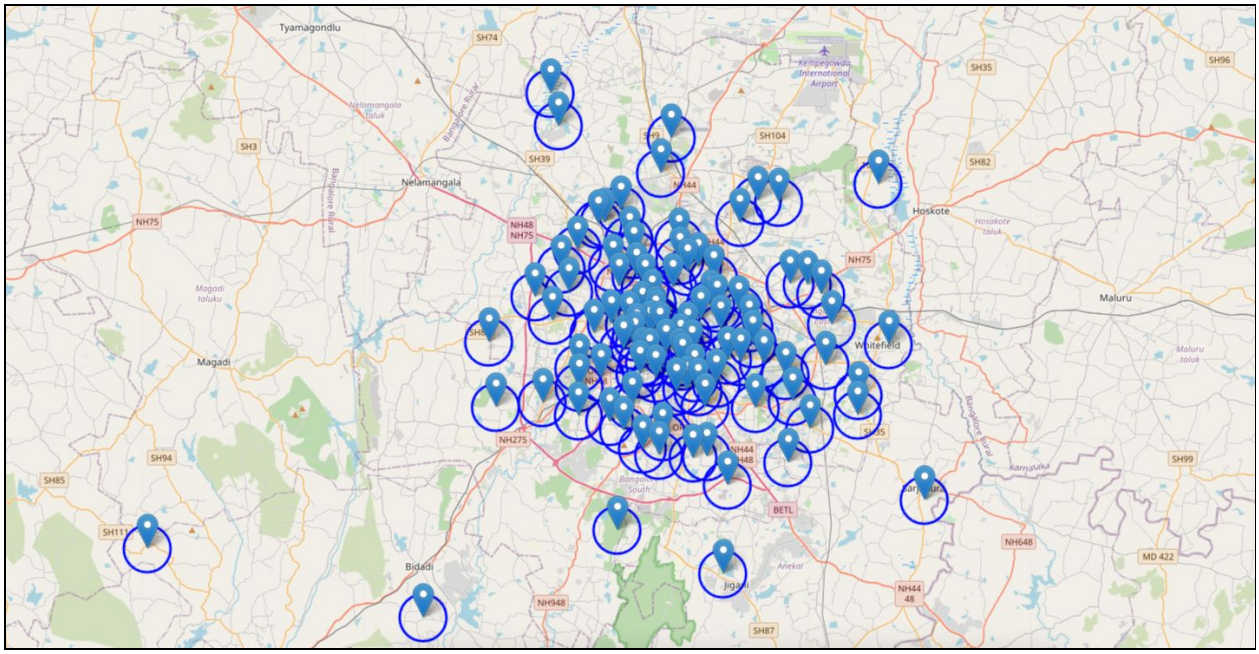
We use the Folium library to achieve this.

```
##@title Displaying the Map for the User's Selected Pin Code

map_bengaluru = folium.Map(location = [bangalore_places_users_pincode['Latitude'].iloc[0],
                                       bangalore_places_users_pincode['Longitude'].iloc[0]],
                           zoom_start = 15)
folium.Marker([bangalore_places_users_pincode['Latitude'].iloc[0],
               bangalore_places_users_pincode['Longitude'].iloc[0]],
              popup='Alexanderplatz').add_to(map_bengaluru)
for lat, lon in zip(bangalore_places_users_pincode['Place Latitude'].unique(),
                    bangalore_places_users_pincode['Place Longitude'].unique()):
    folium.CircleMarker([lat, lon], radius=2, color='blue', fill=True, fill_color='blue', fill_opacity=1).add_to(map_bengaluru)
    folium.Marker([lat, lon]).add_to(map_bengaluru)
map_bengaluru
```

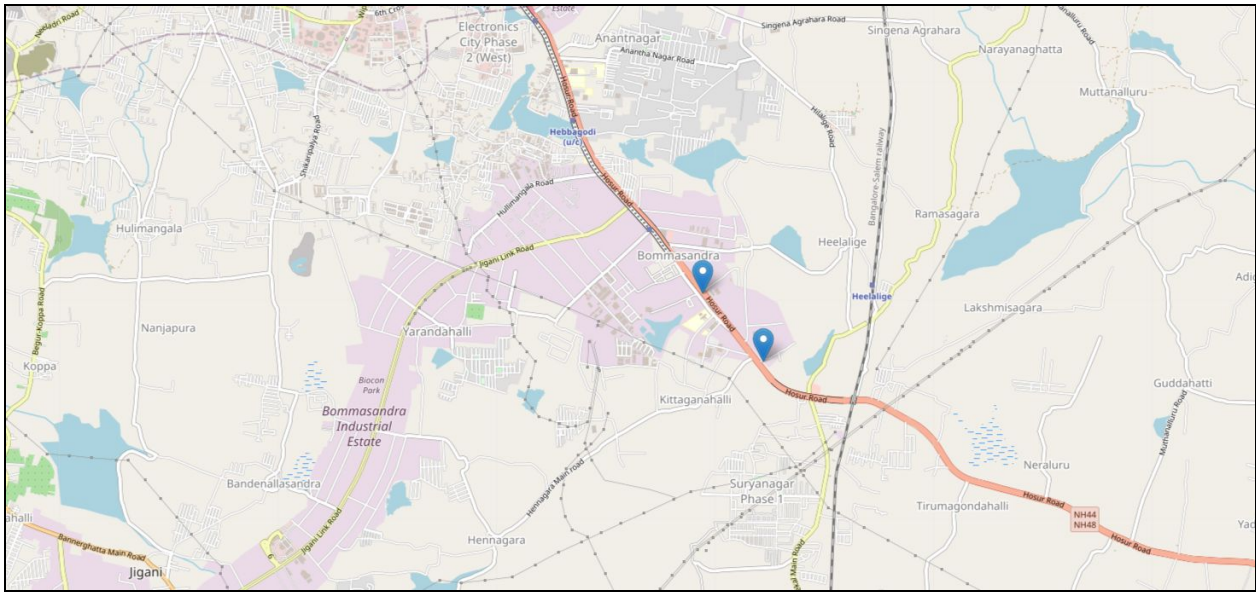
Code to display the map

Folium library requires only a few lines of code to get a map drawn on the display!



Marked location of all the Pin Codes and their radius.

As we can see, there are multiple Pin Codes that overlap quite strongly over each other. Then, there are Pin Codes that are stand-alone and quite far from the city.
Since there were no “approved or verified” list of Pin Codes that are marked as being a “part of the city”, we had to make do with Google Search results.



The two places returned for Pin Code 560025

Unfortunately, Folium was only able to highlight the place but the user will have to rely on the tabular output to know of their names.

Challenges I encountered in designing and writing the code?

Understanding the Google Maps APIs - How do I call them to obtain the list of venues in a given geography? Do I need to create some sort of a Google Cloud account? Would I need to pay money? How much time/effort will go into understanding the API functions and implementing them in code? If the output of the Google Maps API will be in JSON, how do I parse it to a format I prefer?

This process took some time to understand. However, as described in the “Understanding the Google Developer API websites” section of this document, we needn’t understand the entire Google Console system.

How will I “split” the city into neighbourhoods? Are we fine with having some overlap? - Since this model is for a specific city, it’s a one-time work and I can manually create a table with the Pin codes, locality name and coordinates. It’s also fine if there is some overlap between localities, what matters is the radius I choose. Despite the overlap, we would still be able to inform the user of the locality that suits their needs.

How simple/complicated should the user input be?
We decided to keep this as simple as possible. Let’s show the User a list of unique “place types” and ask them to select a few.

Later on, we will also show a list of pin codes/localities that have places of interest to the user, asking them to select a pin code for us to display a map of all the places in that pin code.

The basic objective here is to help the user make a decision on the locality they could consider to settle into. Asking for additional inputs from the user, that might not necessarily put us in a position to serve them better would be unnecessary here.

How will I draw the map and highlight the areas?
This was the most difficult part of the entire project! I realised that for some reason GMaps libraries were not loading in Colab. They were supposedly working fine in Jupyter notebooks, but Google Colab was simply refusing to display the map.
The library would generate the HTML file for the map, which would then work when I load it on a new tab in my browser, but trying to display it Colab would simply not work!
I spent almost two weeks trying to debug this!

I then chose to go with a different Map library called Folium. This library worked(thankfully!) in Colab. I was able to draw the maps.
It is obvious that the map isn't from Google Maps, but as long as I am able to highlight the places, any decent map would work fine!