CapstoneProject: TheBattleOfNeighborhoods

Introduction

An entrepreneur is planning to setup a food and beverages business in Indiranagar, Bangalore. They have approached me to identify locations in and around the area where a business can be set up. The project should provide information about a specific location in Indiranagar which is of high interest among consumers and should also provide information about the type of services the business needs to provide to attract customers.

Data Requirements

- The data about venue and its details will be sourced from Foursquare APIs
- Google search to find additional information about the location and venue details

The following Foursquare APIs seems to have the required information.

Foursquare API to search for venues-
https://developer.foursquare.com/docs/api/venues/search

Foursquare API to get venue details -
https://developer.foursquare.com/docs/api/venues/details

Methodology

Step 1 - Understanding Foursquare API to search for Venues

In this step we will look at the documentation of The Search for Venues and understand the usage of the API. We will need to figure out the following:

1. Request URL
2. Mandatory information that needs to be provided while requesting services
3. Optional information that can used to enhance the search results
4. The response format
5. How to extract, clean and transform the result

From the documentation the request url that needs to be used for searching for venues around a location is

GET https://api.foursquare.com/v2/venues/search

The above request returns a list of venues near the current location, optionally matching a search term. Given below are the input parameters that can be used to customise the search request.

Name	Example	Description
ll	40.74224,-73.99386	**required** unless near is provided. Latitude and longitude of the user's location. Optional if using intent=global
near	Chicago, IL	**required** unless ll is provided. A string naming a place in the world. If the near string is not geocodable, returns a failed_geocode error. Otherwise, searches within the bounds of the geocode and adds a geocode object to the response.
intent	checkin	One of the values below, indicating your intent in performing the search. If no value is specified, defaults to checkin. Value Description checkin Finds venues that the current user (or, for userless requests, a typical user) is likely to checkin to at the provided ll, at the current moment in time. This is the intent we recommend most apps use. global Finds the most globally relevant venues for the search, independent of location. Ignores all parameters other than query and limit. browse Find venues within a given area. Unlike the checkin intent, browse searches an entire region instead of only finding venues closest to a point. A region to search can be defined by including either the ll and radius parameters, or the sw and ne. The region will be circular if you include the ll and radius parameters, or a bounding box if you include the sw and ne parameters. match Finds a venue that is the near-exact match for the given parameters. This intent is primarily used when trying to harmonize an existing place database with Foursquare's and is highly sensitive to the provided location. The result will take into account distance and spelling variations. name and ll are the only required parameters for this intent, but we also suggest sending phone, address, city, state, zip, and twitter for better results. There's no specified format for these parameters—we do our best to normalize them and drop them from the search if unsuccessful.
radius	250	Limit results to venues within this many meters of the specified location. Defaults to a city-wide area. Only valid for requests with intent=browse, or requests with intent=checkin and categoryId or query. Does not apply to intent=match requests. The maximum supported radius is currently 100,000 meters.

sw	44.3,37.2	With ne, limits results to the bounding box defined by the latitude and longitude given by sw as its south-west corner, and ne as its north-east corner. The bounding box is only supported for intent=browse searches. Not valid with ll or radius. Bounding boxes with an area up to approximately 10,000 square kilometers are supported.
ne	44.1,37.4	See sw.
query	tacos	A search term to be applied against venue names.
limit	10	Number of results to return, up to 50.
categoryId	4bf58dd8d488d11094, 4bf58dd8d1bd941735	A comma separated list of categories to limit results to. If you specify categoryId. Specifying a radius may improve results. If specifying a top-level category, all sub-categories will also match the query. Does not apply to intent=match requests.
llAcc	10000	Accuracy of latitude and longitude, in meters.
alt	0	Altitude of the user's location, in meters.
altAcc	10000	Accuracy of the user's altitude, in meters.
url	http://nymag.com/food	A third-party URL which we will attempt to match against our map of venues to URLs.
providerId	nymag	Identifier for a known third party that is part of our map of venues to URLs, used in conjunction with linkedId.
linkedId	1002207971611	Identifier used by third party specified in providerId, which we will attempt to match against our map of venues to URLs.

Using the above details we can customize the search url as follows:

'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search_query, radius, LIMIT)

Where
CLIENT_ID = 'QXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
CLIENT_SECRET = 'yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy'
VERSION = '20180604'
LIMIT = 30
search_query = 'Hotel Bar'
radius = 2000

To fetch the mandatory information parameters we can use the below code:

```
address = 'Indiranagar, Bangalore'
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

We can then use python request module to use the url to fetch a response from the server. The response as pre documentation will be of the following form:

Field	Description
id	A unique string identifier for this venue.
name	The best known name for this venue.
location	An object containing none, some, or all of address (street address), crossStreet, city, state, postalCode, country, lat, lng, and distance. All fields are strings, except for lat, lng, and distance. Distance is measured in meters. Some venues have their locations intentionally hidden for privacy reasons (such as private residences). If this is the case, the parameter isFuzzed will be set to true, and the lat/lng parameters will have reduced precision.
categories	An array, possibly empty, of categories that have been applied to this venue. One of the categories will have a primary field indicating that it is the primary category for the venue. For the complete category tree, see categories.

With this we conclude step 1. Please note the process of extracting and cleaning data as desired in covered in the Exploratory Data Analysis step.

Step 2 - Get details of all the values returned from Step 1

In this step we will look at the documentation of The Get venue details API and understand the usage of the same. We will need to figure out the following:

1. Request URL
2. Mandatory information that needs to be provided while requesting services
3. Optional information that can used to enhance the search results
4. Identify the parameters from step 1 that can be used to pass as parameters to get venue details(to be used in 2 or 3)
5. The response format

6. How to extract, clean and transform the result.

From the documentation the url to search the details of a venue is as given below:

GET https://api.foursquare.com/v2/venues/VENUE_ID

The input parameters that goes with the above request are:

Name	Example	Description
VENUE_ID	XXX123YYY	required ID of the venue to retrieve

Now we can use the list of Venue Id that we fetched in Step 1 to get the details of all 30 venues. The following response is return

Field	Description
id	A unique string identifier for this venue.
name	The best known name for this venue.
contact	An object containing none, some, or all of twitter, phone, and formattedPhone. All are strings.
location	An object containing none, some, or all of address (street address), crossStreet, city, state, postalCode, country, lat, lng, and distance. All fields are strings, except for lat, lng, and distance. Distance is measured in meters. Some venues have their locations intentionally hidden for privacy reasons (such as private residences). If this is the case, the parameter isFuzzed will be set to true, and the lat/lng parameters will have reduced precision.
categories	An array, possibly empty, of categories that have been applied to this venue. One of the categories will have a primary field indicating that it is the primary category for the venue. For the complete category tree, see categories.
verified	Boolean indicating whether the owner of this business has claimed it and verified the information.
stats	Contains checkinsCount (total checkins ever here), usersCount (total users who have ever checked in here), and tipCount (number of tips here).
url	URL of the venue's website, typically provided by the venue manager.

hours	Contains the hours during the week that the venue is open along with any named hours segments in a human-readable format. For machine readable hours see venues/hours
popular	Contains the hours during the week when people usually go to the venue.
menu	An object containing url and mobileUrl that display the menu information for this venue.
price	An object containing the price tier from 1 (least pricey) - 4 (most pricey) and a message describing the price tier.
rating	Numerical rating of the venue (0 through 10). Returned as part of an explore result, excluded in search results. Not all venues will have a rating.
hereNow	Information about who is here now. If present, there is always a count, the number of people here. If viewing details and there is a logged-in user, there is also a groups field with friends and others as types.
storeId	The manager's internal identifier for the venue.
description	Description of the venue provided by venue owner.
createdAt	Seconds since epoch when the venue was created.
mayor	user who is the mayor (absent if there is no mayor), and count, the number of times they have checked in within the last 60 days.
tips	Contains the total count of tips and groups with friends and others as groupTypes. Groups may change over time.
listed	A grouped response of lists that contain this venue. Contains a summary string representing the acting user's relationship to these lists. If an acting user is present, groups may include todos, created, edited, followed, friends, and others. If this venue is on the acting user's todo list, those items will be included in the todos group.
beenHere	Contains count of the number of times the acting user has been here. Absent if there is no acting user.
shortUrl	A short URL for this venue, e.g. http://4sq.com/Ab123D
canonicalUrl	The canonical URL for this venue, e.g. https://foursquare.com/v/foursquare-hq/4ab7e57cf964a5205f7b20e3
photos	A count and groups of photos for this venue. Group types are checkin and venue. Not all items will be present.
likes	The count of users who have liked this venue, and groups containing any friends and others who have liked it. The groups included are subject to change.

like	Indicates if the current user has liked this venue.
dislike	Indicates if the current user has disliked this venue.
phrases	List of phrases commonly seen in this venue's tips, as well as a sample tip snippet and the number of tips this phrase appears in.
attributes	Attributes associated with the venue, such as price tier, whether the venue takes reservations, and parking availability.
roles	Present if and only if the current user has at least one assigned role for this venue. The value is a list of all of the current user's assigned roles for this venue. Possible values for each element of the list are manager and employee. Subject to change as additional roles may be defined.
page	user is the branded page associated with the venue. If the venue is part of a chain, this will be a user object referring to the chain. For venues that are being managed and not part of a chain, this will contain a user object that uniquely refers to this venue.
bestPhoto	Photo we have determined to be the best photo for the venue based on user upvotes and our internal algorithm.

Our customised query will look something like this:

'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(bar_id, CLIENT_ID, CLIENT_SECRET, VERSION)

Mst of the details are the same as the ones before except for bar_id. We need to loop through each venue If from our search API result and use it a the bar_id to iterate and fetch the data for all 30 locations. We can then analyse the details data to arrive at the desired outcome. This is discussed in the following step.

Step 3 - Exploratory Data Analysis and Discussion

Now that we have the data from our two APIs. Let us put them together and analyse them. Since venue is unique for a given venue and is available in both the sets we can merge the two datasets based on this key. Also to help in analysis I will be converting the data set into a pandas dataframe. Since not all venues have details related data I will do an outer join to retain the data where details are not available.

First we identify the location with the highest ratings:

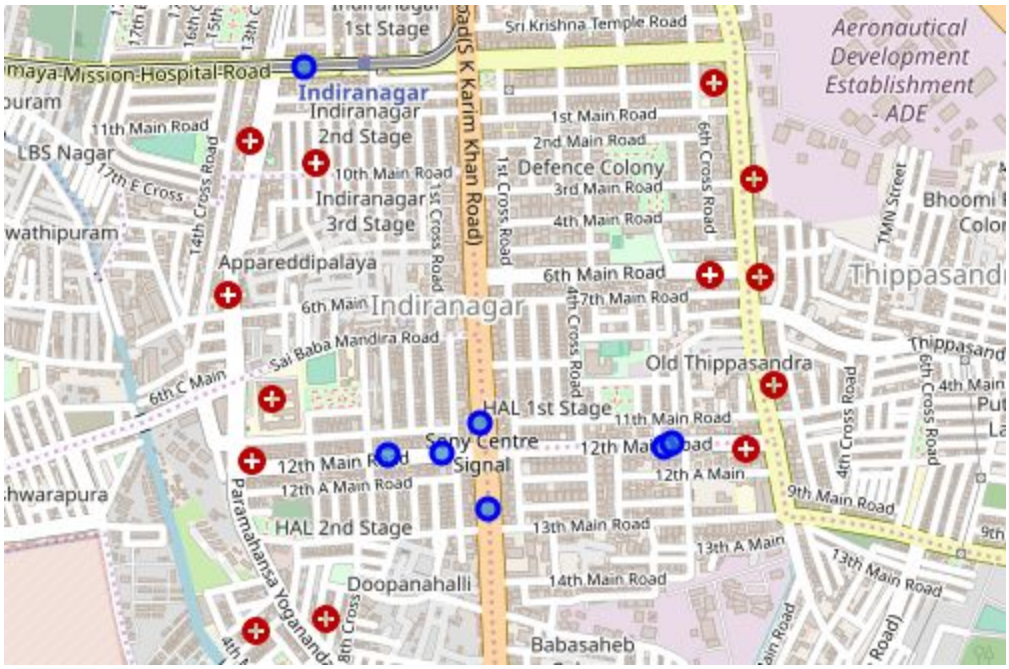
```
#Let us check the top rated venues
merge_df['rating'].sort_values().value_counts()
```

```
7.6    2
7.4    1
6.1    1
7.9    1
8.4    1
8.0    1
7.5    1
5.5    1
Name: rating, dtype: int64
```

As can be seen many of the venues do not have data about popular timings or rating. these could be because of the following reasons

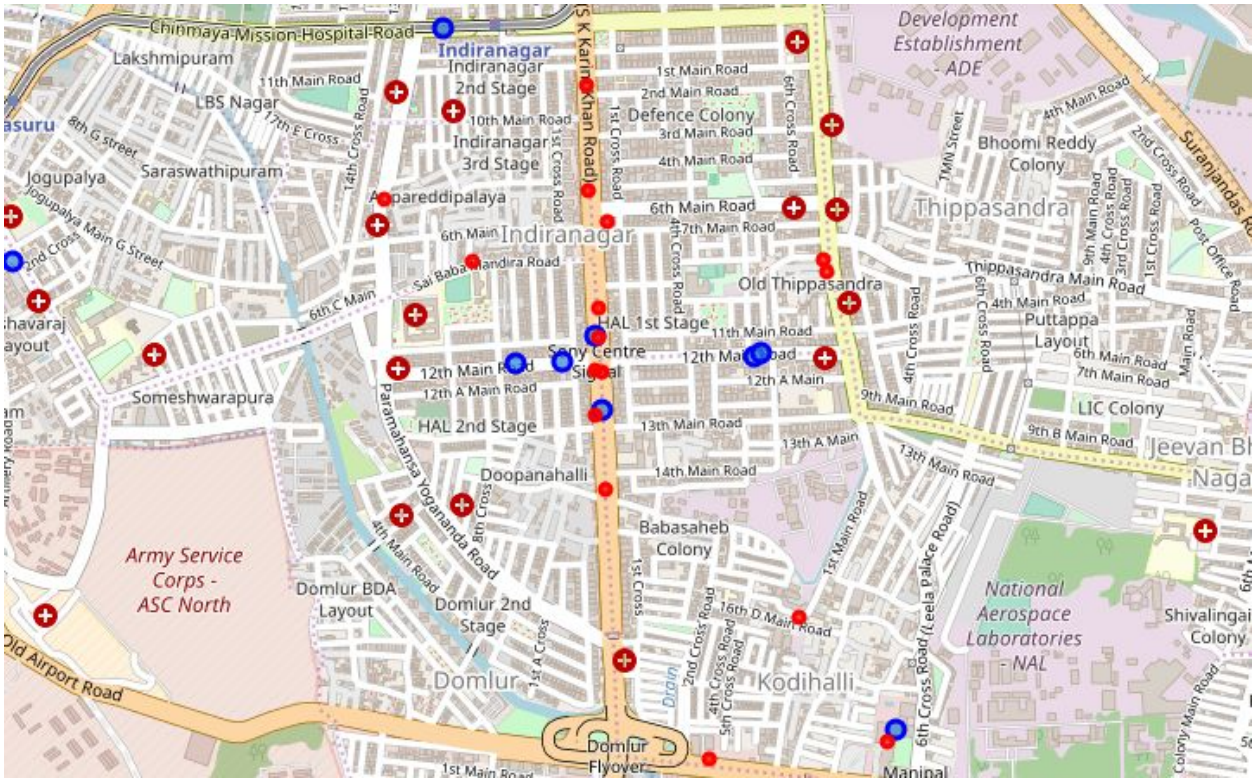
- Foursquare API is not yet popular with all the visitors or folks in Bangalore
- The venues are not that popular/happening

Also most of the venues that have rating seem to have a rating of 7 and above except for one. We will look at the outlier a little later. But for now let us plot the venues with rating. These venues are marked in blue



Most of these venues seem to be centered around **Sony Center Signal.**

Now let us visualize all the venues without any rating information. These venues are marked as red dots without plus mark.



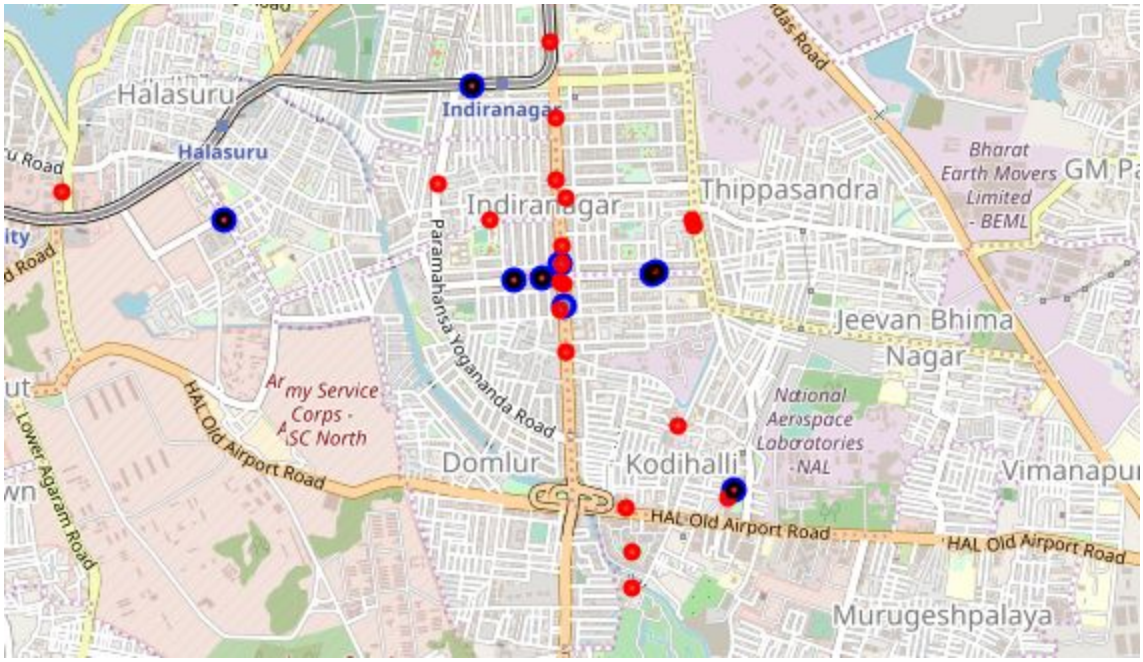
Most of these locations are also centered around the same spot. Let us look at their Categories to get a better understanding as to why some of them are not rated as good as the others.

	name	categories	rating
0	Hotel Shalimar Bar and Restaurant	[{'id': '4bf58dd8d48988d116941735', 'name': 'B...	7.5
3	Monkey Bar	[{'id': '4bf58dd8d48988d116941735', 'name': 'B...	7.4
9	Retro Lounge Bar	[{'id': '4bf58dd8d48988d116941735', 'name': 'B...	8.0
10	Chianti Ristorante & Wine Bar	[{'id': '4bf58dd8d48988d110941735', 'name': 'I...	7.6
11	The Humming Tree - Music Venue and Bar	[{'id': '4bf58dd8d48988d1e5931735', 'name': 'M...	7.9
13	Library Bar	[{'id': '4bf58dd8d48988d116941735', 'name': 'B...	7.6
15	Monkey Bar	[{'id': '4bf58dd8d48988d11b941735', 'name': 'P...	8.4

Interesting, most of the venues seems to be top of the line bars with the only exception being an Italian restaurant.

I searched Google for more information about the other venues. It appears that most of the venues are food and family kinda places. Few of the venues are also local bars where the local working class population would like to visit these venues usually do not have music or good ambience to relax/party. The working class population are not usually tech savvy and may not use applications like Foursquare hence we do not have much information about these venues.

Before we conclude, let us look at the venue with rating greater than 7. These venues are marked in blue with a black ring.



Conclusion

From the above visuals, searches and analysis we can conclude that folks in Bangalore would like to party where there is alcohol served along with music and a good ambience. Most of these venues seems to be centered around 'Sony Center Signal'.

Assumptions and Challenges

1. We are mostly considering a crowd who would like to party, hangout or chill as the target customer segment.
2. Was unable to automate the process of fetching venues details data due to restrictions in the sandbox environment. We could arrive at a better conclusion if we could work on a larger data set.

Enhancements

Analyze the time when the venues are most active, will provide additional information about the customer behaviour and the best time to visit the location on any given day of the week. The data is available in the ‘popular’ column in the data set. This information can be used to add further value to the business in terms of ideal operating hours/days. We can also extend this analysis to other regions of Bangalore to understand the trends across city based on the ethnicity of the folks living in the area. We can extend this to other countries to know how different people are spending their time at the kind locations.