IBM CAPSTONE PROJECT

THE BATTLE OF THE NEIGHBORHOODS (WEEK 1)

Background and Introduction

Problem Description

In today's society there are many ways to shop for everything. These ways include online shopping, in person brick and mortar, and there are services that businesses provide where they will go and shop for the specific items their customers want. Often times it is difficult for a personal shopper or an individual to buy things if they do not know what they specifically want. Then if you factor in an enormous amount of choices and the challenge only escalates. Even with today's recommender systems that can make recommendations based upon a profile, these systems have their weaknesses. Part of the problem is that there are so many choices and the overwhelming number of choices make it difficult for people and smart machines to decide and decide well. According to this NYTimes article https://www.nytimes.com/2010/02/27/your-money/27shortcuts.html "Research also shows that an excess of choices often leads us to be less, not more, satisfied once we actually decide. There's often that nagging feeling we could have done better."

A city like New York is an enormous city and many times when US Citizens or foreigners are looking for a hotel in NYC the choices can be overwhelming. So not only is the problem of choosing a hotel extremely difficult but finding the right area to stay in adds to the complexity. Often times many tourists that visit NYC don't know anything about the city, but they can often, and without hesitation tell you what it is they like they do. They can very easily share the types of experiences they like to have, the venues they visit the most, etc. I believe that when a customer tells you what they like to do, therein lies the solution to part of the problem of having too many choices. If you can tell me what you like to do, I can help you determine where you would best like to stay and give you options that target your likes. I feel that this solution leaves the buyer with less remorse in the long run.

What I am proposing is to use Foursquare data and code a solution that pulls venue data from Foursquare and builds a dataframe that gives you the top ten venues by neighborhood. From there we can pull the foursquare data that has all of the hotels for that area you want to stay based on your tastes in venues. This would simplify the arduous task of picking a hotel from the 669 hotels that are currently available in NYC.

Methodology

1. Collect the data
2. Inspect/Review/Understand the data
3. Prepare the data
4. Model the solution

Data Collection
The environment that I will work from for this solution will be a Jupyter Notebook. While utilizing Python as the scripting language, I will leverage the many different Python libraries that handle JSON data, Dataframes, and mapping objects. The Dataframes will mostly contain location data that is extracted through the Foursquare API. Foursquare is a repository of location data that it obtains through interacting with its partners like Snap, Twitter, Google etc. The users of those mobile applications are sending location data back to Foursquare and Foursquare become the ultimate repository of location data. The Foursquare database is rich with comprehensive location data and it will provide all of the data that we need for this evaluation.

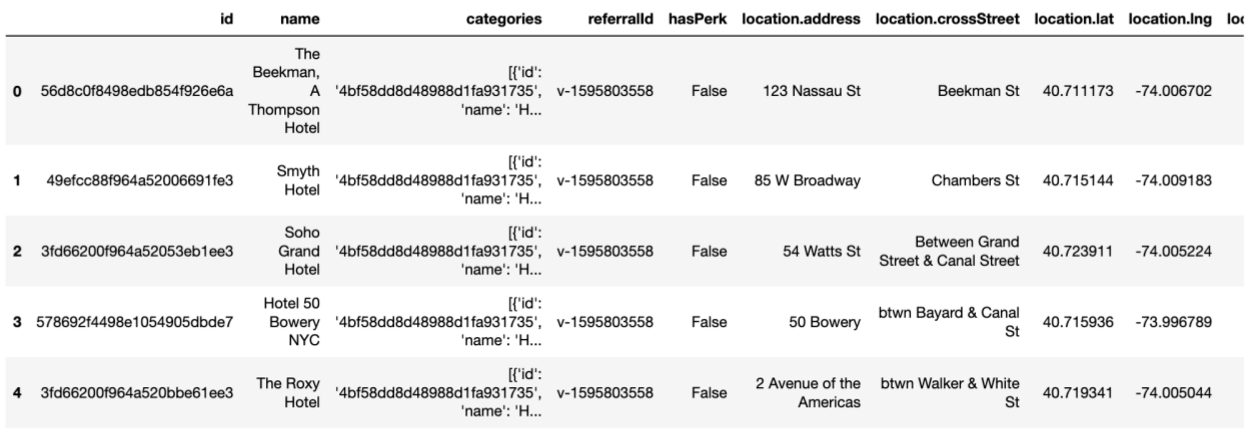
The primary method for retrieving this data will be through the Foursquare RESTful API.
An example query statement through the Foursquare API is below:

https://api.foursquare.com/v2/venues/search?client_id=CLIENT_ID&client_secret=CLIENT_SECRET&ll=LATITUDE,LONGITUDE&v=VERSION&query=QUERY&radius=RADIUS&limit=LIMIT

This single line of code above allows us to pass our credentials and parameters for the query in one contiguous line of code. With a successful response we can code a GET request to extract the JSON data from the results of the query. The results are in. JSON format and will appear as such.

```
{'meta': {'code': 200, 'requestId': '5f1e0504acc3b6486dd91655'},
 'response': {'venues': [{'id': '56d8c0f8498edb854f926e6a',
    'name': 'The Beekman, A Thompson Hotel',
    'location': {'address': '123 Nassau St',
     'crossStreet': 'Beekman St',
     'lat': 40.7111725,
     'lng': -74.0067017,
     'labeledLatLngs': [{'label': 'display',
       'lat': 40.7111725,
       'lng': -74.0067017}],
     'distance': 182,
     'postalCode': '10038',
     'cc': 'US',
     'city': 'New York',
     'state': 'NY',
     'country': 'United States',
     'formattedAddress': ['123 Nassau St (Beekman St)',
      'New York, NY 10038',
      'United States']},
    'categories': [{'id': '4bf58dd8d48988d1fa931735',
      'name': 'Hotel',
      'pluralName': 'Hotels',
      'shortName': 'Hotel',
```

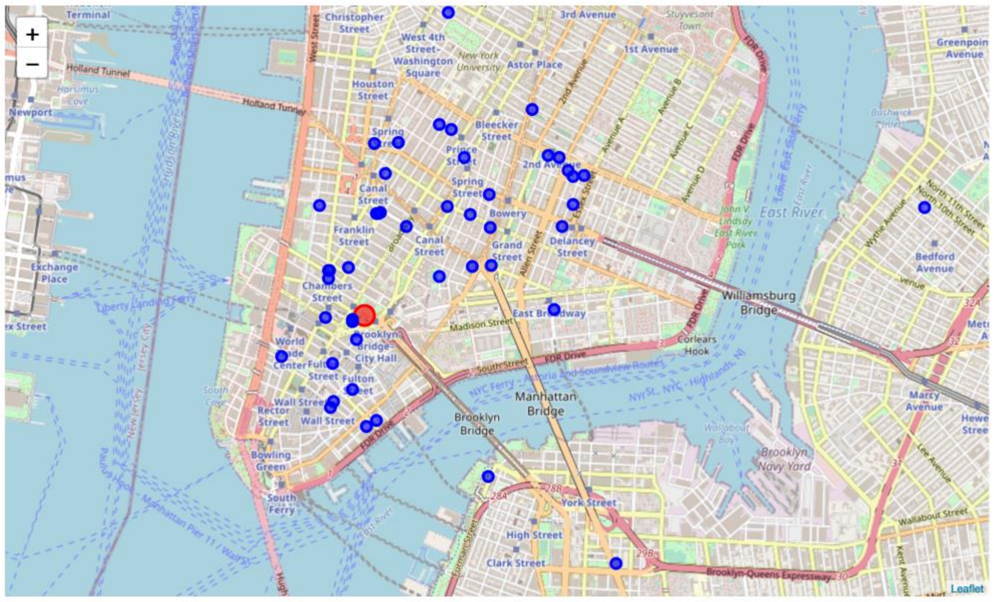
Inspect/Review/Understand the Data
The data extracted will be in numerous lines of JSON data and we can transform the data into a dataframe that would appear as the example below. The dataframe below contains all of the information we requested in the API request. Information such as the hotel name, category, location address, latitude, longitude, etc.



	id	name	categories	referralId	hasPerk	location.address	location.crossStreet	location.lat	location.lng	lo
0	56d8c0f8498edb854f926e6a	The Beekman, A Thompson Hotel	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	v-1595803558	False	123 Nassau St	Beekman St	40.711173	-74.006702	
1	49efcc88f964a52006691fe3	Smyth Hotel	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	v-1595803558	False	85 W Broadway	Chambers St	40.715144	-74.009183	
2	3fd66200f964a52053eb1ee3	Soho Grand Hotel	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	v-1595803558	False	54 Watts St	Between Grand Street & Canal Street	40.723911	-74.005224	
3	578692f4498e1054905dbde7	Hotel 50 Bowery NYC	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	v-1595803558	False	50 Bowery	btwn Bayard & Canal St	40.715936	-73.996789	
4	3fd66200f964a520bbe61ee3	The Roxy Hotel	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	v-1595803558	False	2 Avenue of the Americas	btwn Walker & White St	40.719341	-74.005044	

Prepare the Data
Through a series of Python statements, we can filter, arrange, delete, or append, our dataframe as we need. Ultimately, we want the data to be in a format that is useable for preparing the final solution.

Model the solution
In modeling the solution, we will use the Folium library to prepare a map of the hotels that are in an around the venues where we want to stay. Below is an example of hotels around the financial district. This is ultimately a display where we chose the venues that contain the features that we want and then displays the hotels we should likely choose from.


+
−
Holland Tunnel
Williamsburg Bridge
Manhattan Bridge
Brooklyn Bridge
East River
Chambers Street
Canal Street
Delancey Street
East Broadway
Wall Street
Brooklyn Navy Yard
Leaflet