Dinner and a Date

Jerry Ackaret

October 11, 2019

Introduction

Background

To solve a real world problem, I decided to create something that was useful. My wife volunteers for two non-profit organizations that use silent auctions to make money. Her specialty is organizing and decorating the baskets. One of the most popular we call "Dinner and a Date" baskets. They consist of tickets to a Venue and gift cards to a restaurant, as well as other 'trinkets'.

Prior to my wife's involvement, the Fun Venue and Food Venues were randomly put together, resulting in the possibility of miles between them, not too conducive to a 'date'. She started pairing the two items for being in close proximity, thereby making the basket more appealing.

Problem

The problem is: Since no thought for this arrangement was put into the 'solicitation' of the items, there was no guarantee of arriving at a suitable combination. So, my solution is to provide a list of Fun Venues, and a list of distance and regional associated Food Venues to facilitate the donation requests for what will make suitable combinations.

Audience

The audience for this type of information is the 'Donation Solicitor' of a non-profit organization that uses auctions as a fund raising operation. The proposed information would be used to increase the potential value of the item to the bidders, which in turn **increases the money for the non-profit**.

Data acquisition and manipulation

You can follow along with the project by reviewing the python code that I wrote for the project:

https://github.com/jatcat/Coursera_Capstone/blob/master/DinnerAndADate_Final.ipynb

Primarily, only Foursquare data is necessary to accomplish this project, with the exception of determining Portland, OR latitude and longitude, using geopy.geocoders. I am using both the Explore and Details Endpoint.

From the Explore endpoint, I am retrieving the Portland, OR area information for a limited set of Venues: Museums, Art Galleries, Concert Halls, Historic Sites, Stadiums, and Zoos.

The underlying idea is to limit the venues to more upper end venues, thereby increasing the perceive value of the package. A partial example of this type of data is:


```
'venue': {'id': '43ee1bd9f964a5205f2f1fe3',
 'name': 'Portland Art Museum',
 'location': {'address': '1219 SW Park Ave',
  'lat': 45.51622988093649,
  'lng': -122.68359661102295,
  'labeledLatLngs': [{'label': 'display',
    'lat': 45.51622988093649,
    'lng': -122.68359661102295}],
  'postalCode': '97205',
  'cc': 'US',
  'city': 'Portland',
  'state': 'OR',
  'country': 'United States',
  'formattedAddress': ['1219 SW Park Ave',
   'Portland, OR 97205',
   'United States']},
 'categories': [{'id': '4bf58dd8d48988d18f941735',
```


Obviously, the Foursquare API returns more information than is necessary for this analysis, so I scrubbed the data down to Name, Category, Latitude, Longitude, Address, and Venue ID and called these the Fun Venues. The id of a venue is a key item to include, as it is unique and in turn may be used to link datasets together.

Once that data is retrieved into a DataFrame, it looks like:

	name	categories	lat	lng	address	id
0	Portland Art Museum	Art Museum	45.516230	-122.683597	1219 SW Park Ave	43ee1bd9f964a5205f2f1fe3
1	Pittock Mansion	Museum	45.525262	-122.716684	3229 NW Pittock Dr	41b8e700f964a520801e1fe3
2	Oregon Museum of Science & Industry (OMSI)	Science Museum	45.508601	-122.665915	1945 SE Water Ave	415c9e00f964a520551d1fe3
3	Portland Children's Museum	Science Museum	45.508530	-122.717786	4015 SW Canyon Rd	4efdf6cea69d45461bd1cf1a
4	Oregon Historical Society	History Museum	45.515914	-122.682456	1200 SW Park Ave	4b5ce375f964a5207a4929e3

Also using the Explore Endpoint, I obtained the Food Venues that surround each Fun Venue, within a radius that is as a variable early in the program. I restricted the data to what will be necessary for the analysis, namely: Name, ID, Latitude, Longitude, Address, and Distance to Fun Venue.

Part of the retrieval process of the Food Venues involved merging the Fun and Food Venues into a single DataFrame, of which a partial view looks like this:

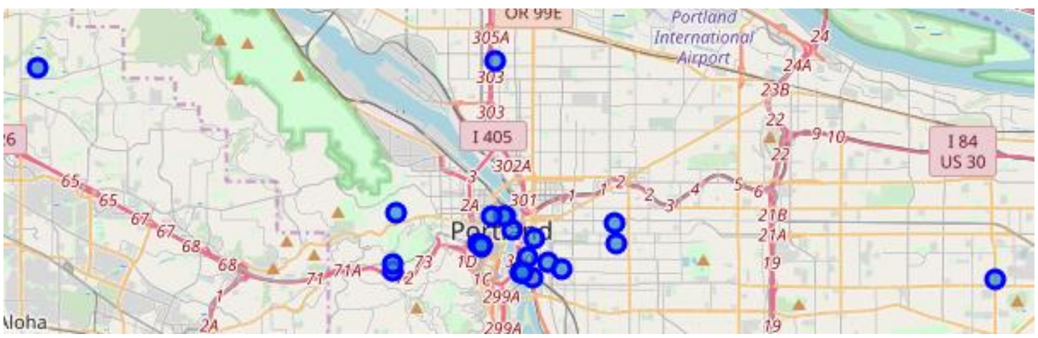
	Fun_Venue	Fun_Address	Fun_Latitude	Fun_Longitude	Fun_ID	Venue	Venue_ID	Venue_Latitude	Venue_Longitude
0	Portland Art Museum	1219 SW Park Ave	45.51623	-122.683597	43ee1bd9f964a5205f2f1fe3	Higgins Restaurant & Bar	41ddd100f964a520c51e1fe3	45.515464	-122.682074
1	Portland Art Museum	1219 SW Park Ave	45.51623	-122.683597	43ee1bd9f964a5205f2f1fe3	Behind The Museum Café	4ee11c6abe7b7e4d131503ff	45.516492	-122.684151
2	Portland Art Museum	1219 SW Park Ave	45.51623	-122.683597	43ee1bd9f964a5205f2f1fe3	Raven & Rose	50960315e4b0e34d634b48f5	45.514794	-122.682321
3	Portland Art Museum	1219 SW Park Ave	45.51623	-122.683597	43ee1bd9f964a5205f2f1fe3	Southpark Seafood & Oyster Bar	40b13b00f964a520fef51ee3	45.517909	-122.681596
4	Portland Art Museum	1219 SW Park Ave	45.51623	-122.683597	43ee1bd9f964a5205f2f1fe3	Shigezo Restaurant	4cae07f1bf70236a47fc04f9	45.517737	-122.682393

The last data that I used was from the Detail Endpoint. The Explore Endpoint does not have phone numbers or ratings, and so using just the Venue IDs from Fun and Food, I retrieved the phone numbers and ratings where they exist, and merged this data back into the master list. Here is a partial example of that DataFrame. Note the addition of the Fun_Phone and Fun_Rating.

	Fun_Venue	Fun_Address	Fun_Phone	Fun_Rating	Fun_Latitude	Fun_Longitude	Fun_ID	Venue	Venue_Category	Venue_Add
74	USS Blueback	1945 SE Water Ave	NaN	7.9	45.508422	-122.666018	4b48e783f964a520a85c26e3	Portland Spirit Cruises and Events	Restaurant	[11 Caruther Portland 97214, U
110	USS Blueback	1945 SE Water Ave	NaN	7.9	45.508422	-122.666018	4b48e783f964a520a85c26e3	Newport Floating Restaurant	Seafood Restaurant	[Portland, United St
242	World Forestry Center	4033 SW Canyon Rd	5032281367	7.5	45.510465	-122.718061	49cc4846f964a5205e591fe3	AfriCafe	Café	[at Oregon Portland 97221, U
238	World Forestry Center	4033 SW Canyon Rd	5032281367	7.5	45.510465	-122.718061	49cc4846f964a5205e591fe3	Cascade Grill	Restaurant	[SW Zoo Portland, United St
240	World Forestry Center	4033 SW Canyon Rd	5032281367	7.5	45.510465	-122.718061	49cc4846f964a5205e591fe3	Bear Walk Cafe	Café	[SW Zoo Portland, United St

Methodology

Once I had the Fun Venues, and prior to pulling the associated Food Venues, I used the mapping function folium to show the Fun Venues on a map to determine if there were any to discard.

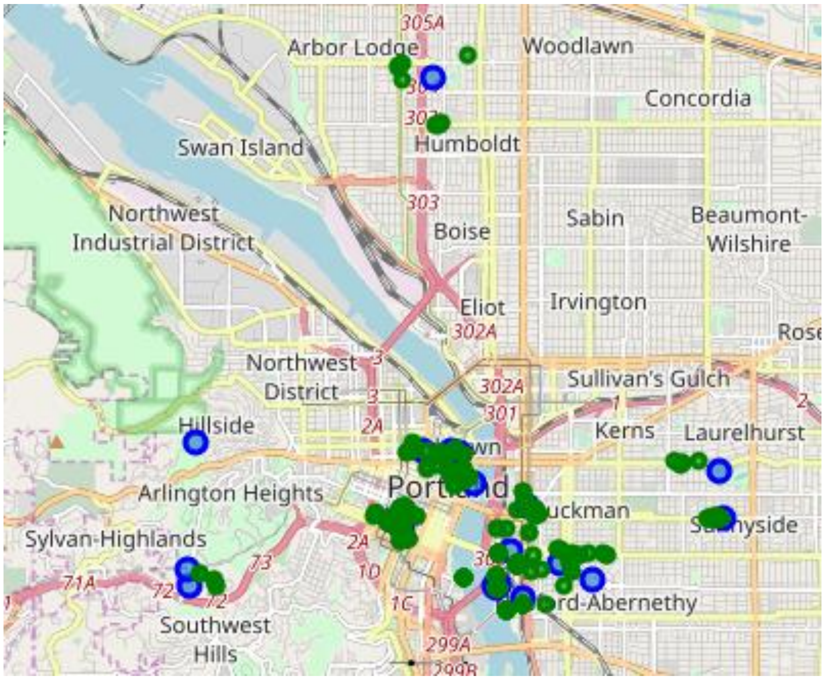


From this I determined that there were two venues that fell outside of the core part of Portland and decided to drop them as being out of the scope of the project. These are the one on the far right and one on the far left of this map. Selecting their Circles gave me their names; 'Washington County Museum', and 'Madrugada Pottery'.

When I first looked at the Food Venues, I discovered that there were a fair number that just did not meet the level that I wanted for a Dinner and a Date package. So, I reran the code with the following categories dropped: 'Food Truck', 'Donut Shop', 'Bakery', 'Bagel Shop', 'Coffee Shop', 'Dessert Shop', 'Fast Food Restaurant', 'Food Court', 'Food Stand', 'Juice Bar' and ,'Snack Place'.

Since the result of a non-limited dataset resulted in over 1,500 Food Venues around these 27 Fun Venues, and knowing that a list of that magnitude would be overwhelming as a usable output, I chose to limit the number going forward to per Fun Venue. Since part of the intent was to have Food Venues that were close to the Fun Venue, I used distance to further contract the results to the closest ones to the Fun Venues. The maximum number to retrieve was set early in the program in order for it to be simple to change if desired. I used 10 for the purposes of this report. The full list was also maintained separately, as it was necessary for cluster analysis.

A mapping of the resulting Fun Venues and their associated Food Venues looks like:



Notice that there is one by the word “Hillside” that has no Food Venues around it. While this is a very nice venue (called the Pittock Mansion, a 1914 historical house museum), it became apparent that the idea of pairing this with a close food venue was in vain. This lack is important to note, and will be readdressed in the Clusters portion.

The groupby/count function was used to determine the number of Food Venues that were associated with each Fun Venue. This information was used to adjust the radius used in the Food Venue discovery stage. I had originally used 500 meters, but found that some Fun Venues had very limited choices. I increased that radius to 750 meters and reran the program.

At this point, the process split into two.

Spreadsheet

For the output to the Donation Solicitor, I chose the Excel Spreadsheet format. Since there was a lot of information that they would not need, I dropped several fields before exporting the spreadsheet file: 'Fun_Latitude', 'Fun_Longitude', 'Fun_ID', 'Venue_Latitude', 'Venue_Longitude', and 'Venue_ID'.

Clusters

But, while the spreadsheet is the main information for the Solicitor to follow, it does not provide ALL of the information necessary to arrive at which Food Venues would be the best to contact. Even though distance and rating can be beneficial, it also is of some worth to know what types of food is most prevalent in the area, under the theory that the venues that have the best ratings of the most options will tend to be the actual best venues, I used

Clustering to determine each region of Portland's most common types of eating establishments.

The next stage was to use one hot encoding to turn the categorical variables into binary vectors. Since machine learning algorithms cannot use categorical data directly, one hot encoding is a necessary step.

I turned that data into the common types of food venues that exist around each of the Fun venues. I used k-means clustering to determine if there is any regional clustering. The Pittock Mansion had to be dropped from the dataset used in the clusters, because a NAN in the dataset resulted in a decimal for the Cluster, which was not conducive to using the Cluster number for mapping purposes.

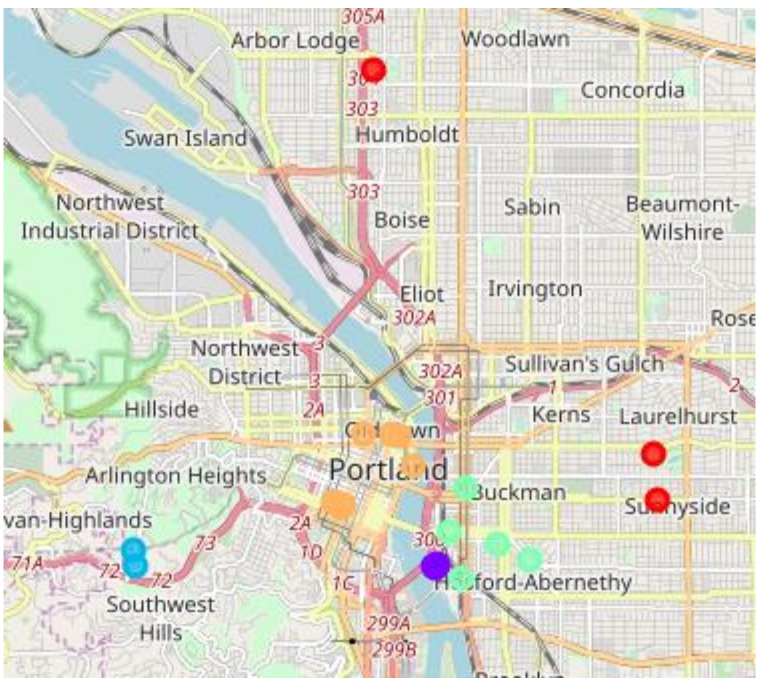
Results

The spreadsheet used for contact information of the Venues is a file called "Solicitation.xlsx, and a copy may be found at:

https://github.com/jatcat/Coursera_Capstone/blob/master/Solicitation.xlsx

This file is automatically produced and provides the appropriate columns for the 'Donation Solicitor' to use in order to first contact the Fun Venue, and if they agree to provide tickets, then find an associated Food Venue. Since there can be as many results as the solicitor wishes, the odds of finding Food Venue to combine with the Fun Venue is high.

The clustering analysis clearly showed regional propensity towards certain types of food venues.



From a Food Venue Category standpoint, Cluster 0 (Purple) on the east side of the river is primarily a Seafood and Mexican Restaurant area, Cluster 1 (Blue) in the West Hills is more Café and Restaurant, Cluster 2 (Green) on the east side of Portland, but reasonably close to the river is Mexican and Pizza shops, Cluster 3 (tan) on the west side of the river is primarily American Restaurants, and finally Cluster 4 (red) further east away from the river is more Thai.

Given this additional information, it would be beneficial for the Solicitor to consider the highest ratings of Food Venues in these categories for the individual clusters.

Discussion

I believe that the results that I obtained will be beneficial in the coming year donation solicitation. I have spoken to the chairperson of one of the non-profits that my wife volunteers for, and she is quite interested in getting a list to start working with. The underlying technique used can be also used in other ways. First, changing the parameters of the Fun Venues would be of use to determining Fun Venues that are family oriented, such as water parks, bowling alleys, etc. In addition, either of these can be used personally, by people that just want to explore finding new venues that would make for a good outing with one's significant other, or family.

Conclusion

While the nature of my job as a Reliability Engineer in the design of servers means that this is probably my only exposure to Foursquare, I was glad to find a project to use for this Capstone that was useful for someone, and not just academic. I was pleased that one of my intended audience is already excited to see the results.