Determining the best city to open a Pizza Delivery service

November 3, 2020

1. Introduction

1.1 Business Problem

A small business owner wants to open up a pizza delivery service to cater only to the surrounding pizza restaurants. It will speed up the time it takes to get pizza from restaurant to customer by only delivery from pizza restaurants. The business owner wants to make sure there are enough surrounding restaurants in the city that they open in, to ensure successful business. They want to open in a busy multicultural city in Pennsylvania.

1.2 Interest

The interest in this data will be this entrepreneur who wants to open in one of these two cities. Or anyone else wanting to open a pizza delivery service within Pennsylvania.

2. Data acquisition and cleaning

2.1 Data sources

All the necessary data for this project will be sourced from using the *Foursquare API*. By querying 'Pizza' restaurants within one city and building a DataFrame – and then querying the same but in the second city, we will be able to form our basis for evaluation.

2.2 Data cleaning

The data is a little messy when first querying from the *Foursquare API*. First, we'll want to view the raw json and we can better understand the data we are working with. Upon reviewing the json, we determine that we'll want to add the venues into a DataFrame and leave out the other headers.

```
venues = results['response']['venues']

dataframe = pd.json_normalize(venues)
```

The above code will be a great starting point in cleaning our data. Next, we will make sure to filter only the columns we want to view – everything dealing with "location", as well as the "name", "categories", and "id". Once we filter the columns, we can set this as a variable called "phil_df" -- short for Philadelphia DataFrame.

Now we can do the exact same process, but first querying Pittsburgh with the *Foursquare API*. Once we have cleaned this DataFrame, we can set the variable to be named “pitt_df” -- short for Pittsburgh DataFrame.

3. Exploratory Data Analysis

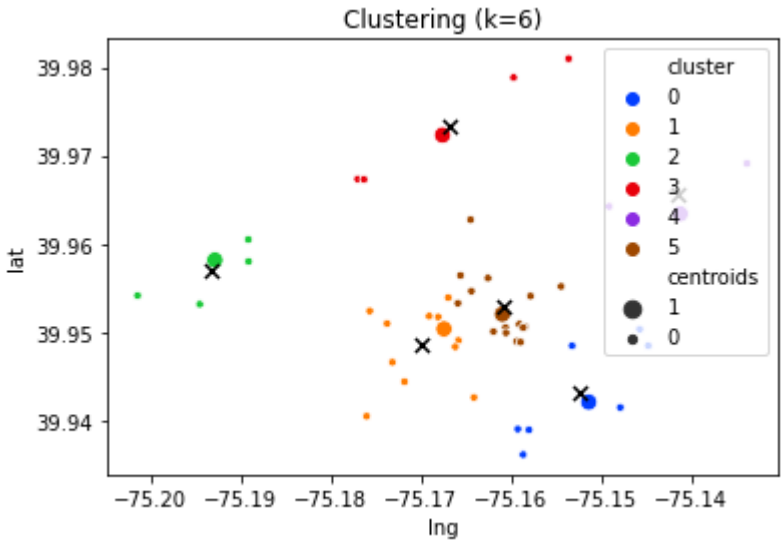
3.1 Plotting venues on map

Once we have both cities with their own separate DataFrame, we can plot these points on a map to better visualize where these venues exist. Using the folium library, and adding CircleMarkers to the map, we can see all our queried venues on the generated map. But still, we want to explore our data further.

3.2 Using k-Means cluster

Let’s use k-Means cluster to separate our venues into groups. First, we will need to determine what would be the best k to use in this situation. Building a *for* loop and then generating the results onto a graph, we can find out what would be the best k to use. The best k will be the one with the lowest derivative. For our first city, the graph shows that the best k would be *k = 6*.

Next, we input that k into the algorithm which groups our points into 6 clusters – and then it determines which is the centroid of each of those clusters. We’ll add this information to our existing DataFrame. Then we can plot these points in their individual clusters on a graph to better visualize where they are.

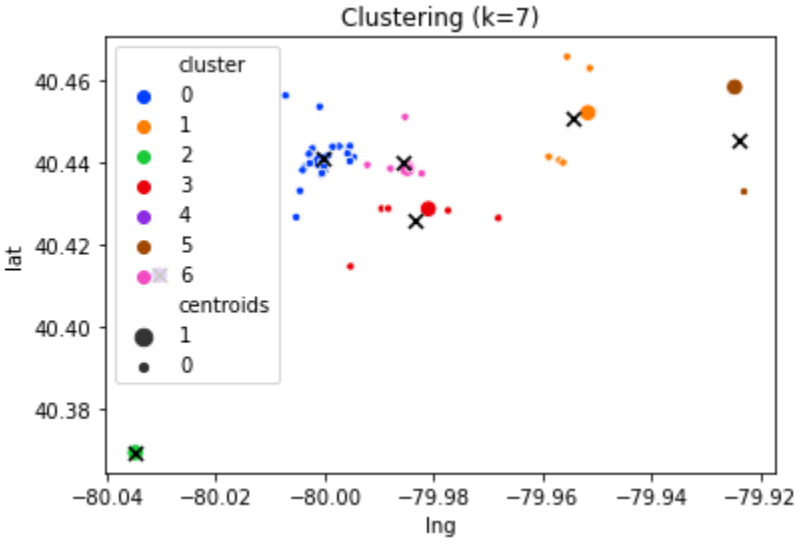


*[Graph of **Philadelphia** venues in 6 clusters]*

We can see that these clusters are evenly grouped, not by the number of points in each cluster but by the size of each cluster. So, we might want to revisit the value of *k* depending

on what our next cities k-Means cluster algorithm returns. The largest cluster is Cluster #5 (brown) with a total number of 16 venues in it.

Let's run this process again, this time using the data from Pittsburgh's venues. The new graph is suggesting to use *k = 7* for this set of data. We can now put this value into the k-Means clusters algorithm and plot the points onto a graph.

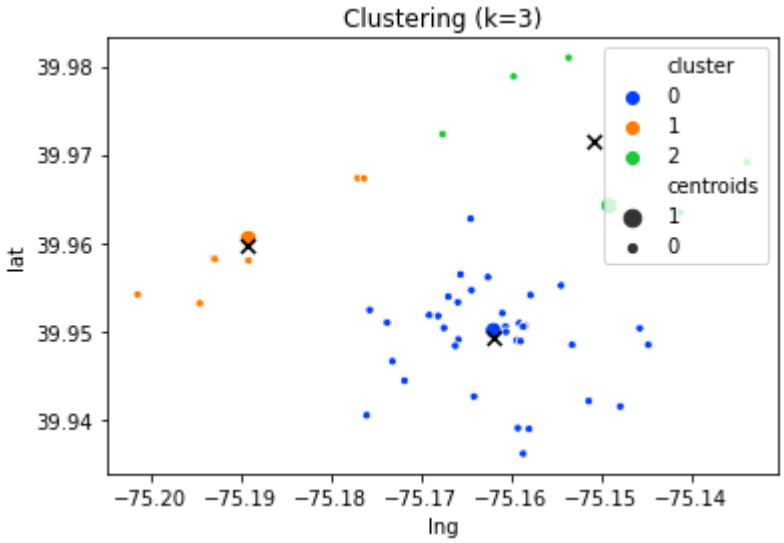


*[Graph of **Pittsburgh** venues in 7 clusters]*

The largest cluster is Cluster #0 (blue) with a total number of 24 venues in it. Since we are wanting to deliver to the largest group of venues, this would be a good choice. But let's try changing the value of k in both of these sets of data to try to service more than 24 venues in a cluster.

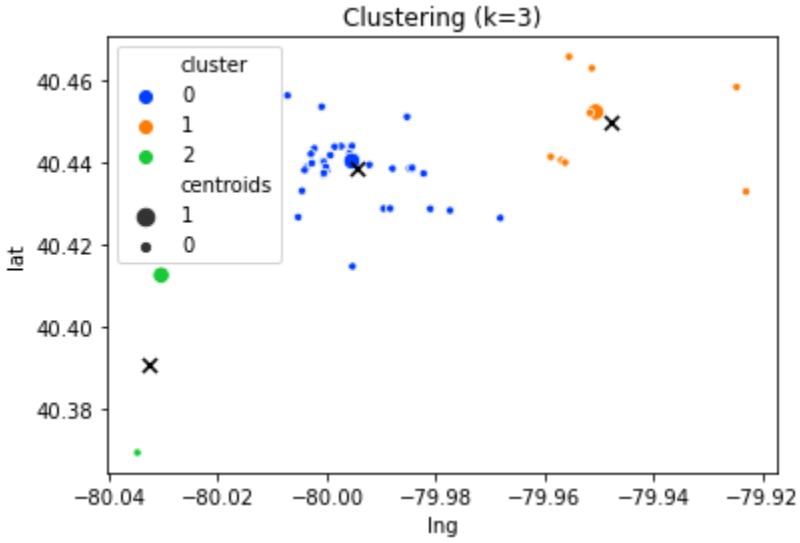
3.3 Refining the algorithm

We can change the value of k to 3, and see what the results are.



*[Graph of **Philadelphia** venues with 3 clusters]*

Philadelphia (above): The largest cluster here is Cluster #0 (blue) with a total number of 36 venues.



*[Graph of **Pittsburgh** venues with 3 clusters]*

Pittsburgh (above): The largest cluster here is Cluster #0 (blue) with a total number of 39 venues.

3.4 Determining the best point

Pittsburgh seems to be the choice city we want to choose to start our pizza delivery service. Let's find the centroid of this cluster.

	name	categories	address	lat	lng	labeledLatLngs	distance	postalCode	cc	city	state	country
32	brother's pizza	None	Benner pike	40.440394	-79.995415	[{'label': 'display', 'lat': 40.44039429610388...	474	NaN	US	Pittsburgh	PA	United States

Brother's Pizza in Pittsburgh would be the closest restaurant that we would want to center our delivery service around. There are 39 restaurants in close proximity to this restaurant. We will be able to offer delivery service of those restaurants to our customers.

4. Conclusion

Upon analysis, we discovered that opening our Pizza Delivery service in Pittsburgh would be the best choice. After grabbing nearly 50 venues in the vicinity of both cities (Philadelphia and Pittsburgh, PA), we used an algorithm to group these venues into separate clusters. The k-Means clustering algorithm helped to evenly group these venues. Once we adjusted the value of *k* to best fit our needs, we were able to come to the conclusion where the best location to start-up would be. Centering our business around the latitude and longitude of *(40.440394, -79.995415)* – in Pittsburgh, we will be able to service at least 39 pizza restaurants in a close proximity.