What kind of business to start in Houston, Texas

Michael Mehlman

Introduction

For this case study I am trying to identify businesses that are underrepresented relative to similar areas in Houston, Texas.

This would help someone who is looking to strike out and be their own boss identify potential opportunities.

I am centering the search on the approximate location of my brother's home, since he is recently unemployed and looking to potentially work for himself (if there is enough demand!).

He loves to cook, so in the end, I will pay special attention to restaurants.

Data

All data for this project will come from the foursquare API. Each region under investigation will be queried for the top venues of any type within a given radius

Methodology

Region Clustering

I tried to segment Houston into regions three different ways:

1. By neighborhood as defined in https://en.wikipedia.org/wiki/List_of_Houston_neighborhoods
2. By zip code (Houston zip codes all start in 770XX)
3. By lat/long grid

Neighborhood

Though detail about each neighborhood can be found online, I could not find any geojson or other maps that could be used to identify neighborhood centers.

Zip Code

All of Houston's regions (with some minor exceptions) are contained within zip codes starting in 770XX. I initially ran through my analysis using this approach, but realized that The zip codes are far too large (and irregular) to provide meaningful data.

Lat/Long Grid

Ultimately, I chose to center the search on my brother's address and generate rectangular regions of a given size.

This approach has the following advantages:

- Easy to scale to different sized grid or different resolution
- Easily transferable to new cities (simply enter a new center point)
- Allows centering around a point of interest (in this case, my brother's home)
- No bias in data from different region sizes

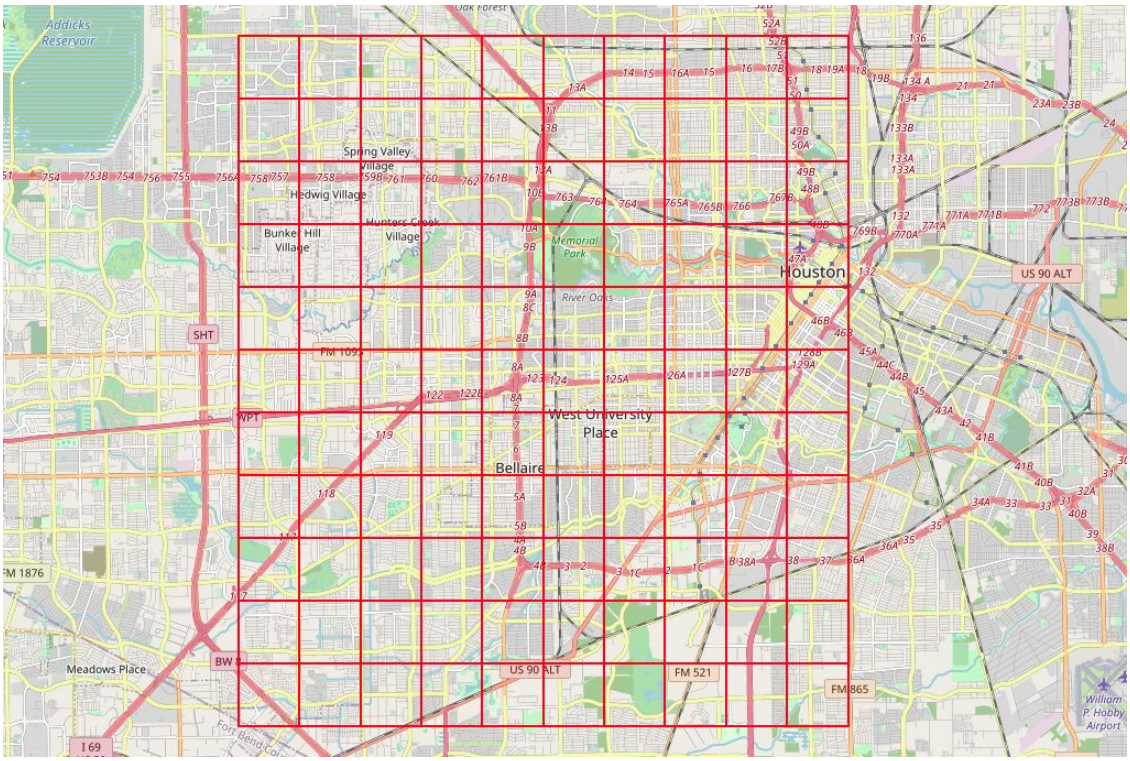
The algorithm takes a center point in lat and long and desired search radius. First, lat/long deltas are incremented until the desired search radius is met from the center point using geopy.distance (note, this would not work correctly at a massive scale since the relationship between lat/long and distance is not consistent across the globe). Then a grid is established using the center point and the lat/long deltas.

After exploring the data, I ultimately chose a desired search radius of 900 m (so rectangles of approximately 1800x1800m).

I chose to generate a 10x10 grid to cover the area within a reasonable commute of my brother's home (hence the final grid is approximately 18x18km centered on his home).

```
clat, clon = 29.72924406953473, -95.44219156375291
lat_num, lon_num = 10, 10
desired_radius = 900.# units in meters
lat_spacing, lon_spacing = 0., 0.
while geopy.distance.distance((clat, clon), (clat + lat_spacing, clon)).m < 2*desired_radius:
  lat_spacing += .001
while geopy.distance.distance((clat, clon), (clat, clon + lon_spacing)).m < 2*desired_radius:
  lon_spacing += .001
latlist = np.arange(clat-lat_num*lat_spacing/2., clat+lat_num*lat_spacing/2., lat_spacing)
lonlist = np.arange(clon-lon_num*lon_spacing/2., clon+lon_num*lon_spacing/2., lon_spacing)
regs, lats, lons = [], [], []
i = 0
for lat in latlist:
  for lon in lonlist:
    regs.append(i)
    lats.append(lat)
    lons.append(lon)
    i += 1
```

This code yielded the resulting grid:



Venue Data

Once the grid was established, I pulled Foursquare data for the center of each grid point, with a radius of 1.5x the desired search radius. Though there is some overlap, this makes sure I am not missing the corners of each grid point. I used the Foursquare API as described in the lab (with some additional error checking):

```
[6] LIMIT = 100
    def getNearbyVenues(regs, lats, lons, radius=1000):
        venues_list=[]
        for reg, lat, lon in zip(regs, lats, lons):
            url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.f
            try:
              results = requests.get(url).json()["response"]['groups'][0]['items']
              venues_list.append([(reg, lat, lon, v['venue']['name'], v['venue']['location']['lat'], v['venue']['location']['lng']
            except:
              print('Error getting data')
              venues_list.append([(reg, lat, lon, None, None, None, None)])
        nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
        nearby_venues.columns = ['reg', 'lat', 'lon', 'venue', 'venue_lat', 'venue_lon', 'venue_category']
        return(nearby_venues)
```

```
[7] venues = getNearbyVenues(regs=df.index.tolist(), lats=df['lat'], lons=df['lon'], radius=radius)
```

Clustering

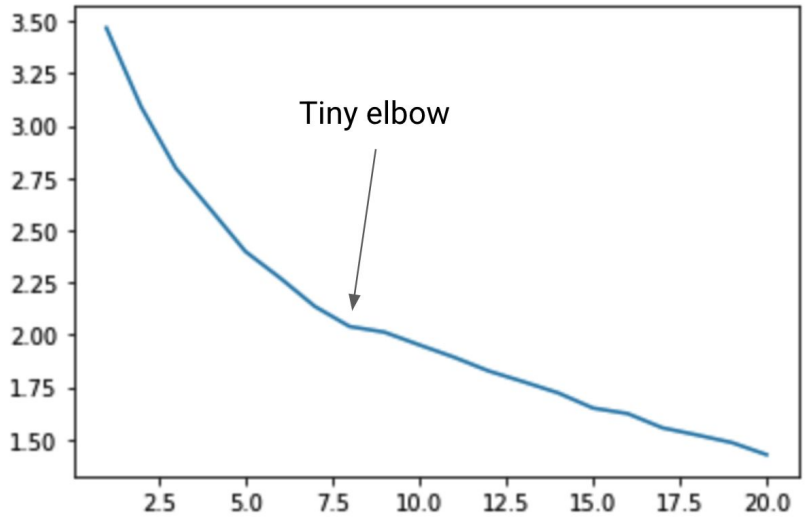
I considered using kmeans and DBSCAN algorithms to cluster.

DBSCAN

I could not get DBSCAN to converge even after generating a for loop over epsilon and minimum number of points. The algorithm would either return a single cluster or place all regions in the -1 cluster, so I switched to the tried and true:

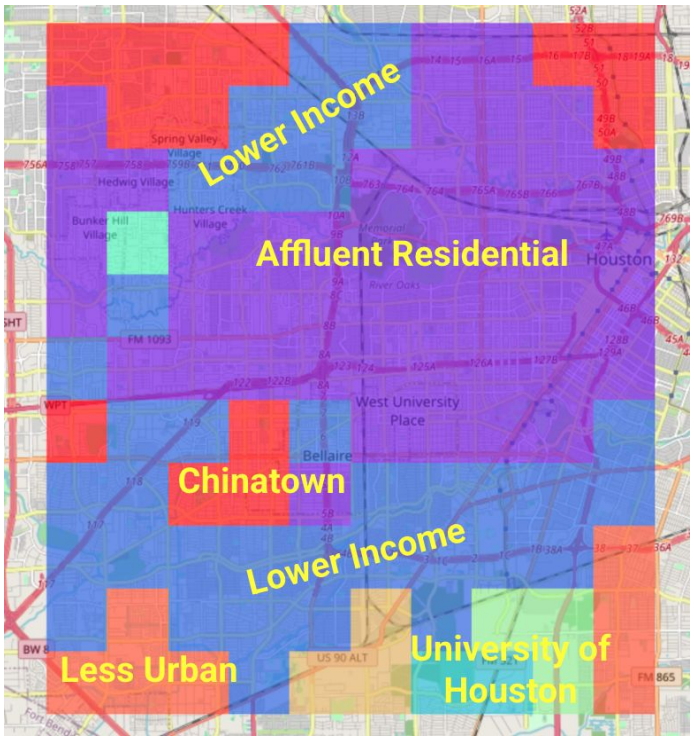
Kmeans

I plotted the kmeans.inertia as a function of number of clusters and did not observe a significant elbow. There was a small elbow at n=8, so I decided to try that in the analysis.



Clustering Results

With n=8, the results were actually fairly consistent with my experience of living in Houston. Algorithms are only useful if they return physical results, so I decided to stick with n=8.



Comparing Business Opportunities Relative to Peers

I subtracted the average occurrence rate of each venue type for the cluster from the actual occurrence rate per venue type in each region.

- Take .mean() of the grouped dataframe by cluster label over each venue type
- Subtract the .mean() from the original grouped dataframe for each venue type
- Clean up and add back in necessary columns like lat/long, etc.

```
v_rep = onehot.groupby('reg').sum().astype(float)
cols = v_rep.columns
v_rep = regs_venues_sorted[['reg','cluster_label']].join(merged[['lat', 'lon']].join(v_rep))
v_rep_mean = v_rep.groupby('cluster_label').mean().iloc[:,3:]
v_rep = v_rep.merge(v_rep_mean, on='cluster_label', suffixes=['', '_avg']).set_index('reg')
for col in cols:
  v_rep[col] = v_rep[col]-v_rep[col+'_avg']
v_rep = v_rep.drop([x+'_avg' for x in cols], axis=1)
v_rep
```

reg	cluster_label	lat	lon	ATM	Accessories Store	Afghan Restaurant	African Restaurant	American Restaurant
0	7	29.644244	-95.537192	0.00000	-0.133333	-0.066667	-0.100000	-0.500000
1	7	29.644244	-95.518192	0.00000	-0.133333	-0.066667	-0.100000	-0.500000
2	7	29.644244	-95.499192	0.00000	-0.133333	-0.066667	-0.100000	-0.500000
10	7	29.661244	-95.537192	0.00000	-0.133333	-0.066667	-0.100000	-0.500000
11	7	29.661244	-95.518192	0.00000	-0.133333	-0.066667	-0.100000	0.500000

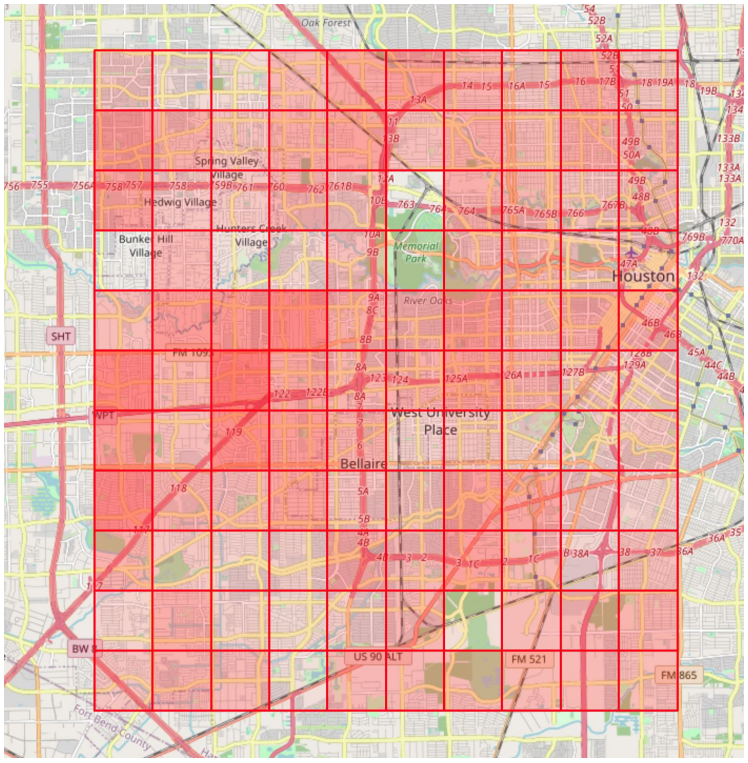
One can use this dataset directly, but in order to get better insights, I chose to look exclusively at restaurants. To do this:

- Use the relative venue representation dataframe and select only columns that include the string ‘cluster’ in the column label.
- Sum over columns (not rows)
- Add a new column using the sum and minmaxscaler so that we can plot it more easily.

```
key = 'Restaurant'
key_rep = v_rep[[col for col in v_rep.columns if (key in col) | ('cluster_label' in col) | ('lat' in col) | ('lon' in col)]]
key_sum = pd.DataFrame(key_rep.iloc[:,3:].sum(axis=1), columns=[key])
key_rep = key_rep.iloc[:,:3].join(key_sum)
key_rep[key+'_scaled'] = MinMaxScaler().fit_transform(key_rep[key].values.reshape(-1, 1)) #need the reshape for a single colu
key_rep.sort_values(key, ascending=False).head()
```

reg	cluster_label	lat	lon	Restaurant	Restaurant_scaled
90	2	29.797244	-95.537192	23.379310	1.000000
52	1	29.729244	-95.499192	23.285714	0.998094
42	7	29.712244	-95.499192	21.266667	0.956967
40	7	29.712244	-95.537192	18.266667	0.895859
63	1	29.746244	-95.480192	17.285714	0.875878

I then plotted the regions and selected region shading to be a maxminscaled value corresponding to restaurant representation relative to peers. A dark cell means the region has more restaurants than peers, a light cell means less (= a potential opportunity for a new business).



The light regions correspond to good potential areas of restaurant under representation relative to peers.

Other Analyses

I also generally examined the resulting grouped dataset to see if there was anything else of interest primarily using the .describe function on the clustered and peer average subtracted dataframe. For each min or max of interest, I dug deeper into that column to try to interpret why.

Since ethnic neighborhoods can change quickly, I think it is safer to exclude regional cuisine restaurants from results.

```
Mexican Restaurant     -3.622222
Coffee Shop            -3.222222
Pizza Place            -3.153846
Sandwich Place         -2.861111
Fast Food Restaurant   -2.583333
Bar                    -2.488889
Grocery Store          -2.384615
American Restaurant    -2.355556
Italian Restaurant     -2.111111
Burger Joint           -1.977778
Name: min, dtype: float64
```

In figure above, Coffee Shop was the most underrepresented non-ethnic venue.

reg	lat	lon	Coffee Shop
85	29.780244	-95.442192	-3.222222
66	29.746244	-95.423192	-3.222222
73	29.763244	-95.480192	-2.222222
96	29.797244	-95.423192	-2.222222
90	29.797244	-95.537192	-2.222222

Above figure shows the regions of Houston (as defined in this report) with least coffee shops relative to their peer groups.

Results

- The region around Memorial Park has few restaurants relative to its peers.
- Bunker Hill Village has surprisingly few restaurants compared to its peers.
- There is a region in south Houston on US90 that also has few restaurants.
- There were a few regions with a dearth of coffee shops and grocery stores as well.

However, these results may not be statistically significant, since, for example, there were only 216 total coffee shops in the entire dataset.

Discussion

I recommend starting to investigate opening a restaurant in the Bunker Hill Village region of Houston. This area seems conspicuously underrepresented in restaurants that make Foursquare top 100 venues in the area. If there are many restaurants here, it is possible that they are not well-reviewed enough to make Foursquare top 100 list, which still means this is a business opportunity worth considering.

Future work may include pulling more venue data from Foursquare in order to better fit the kmeans algorithm and have more certainty when comparing slight deviations from the mean between representation of venues among peer group.

Conclusion

Foursquare data was used to identify potential business opportunities in different regions of Houston, Texas. First, a search area was defined in a regular grid. These areas were clustered into peer groups. Restaurant representation was examined for each region relative to the mean restaurant representation for each region's peer group.