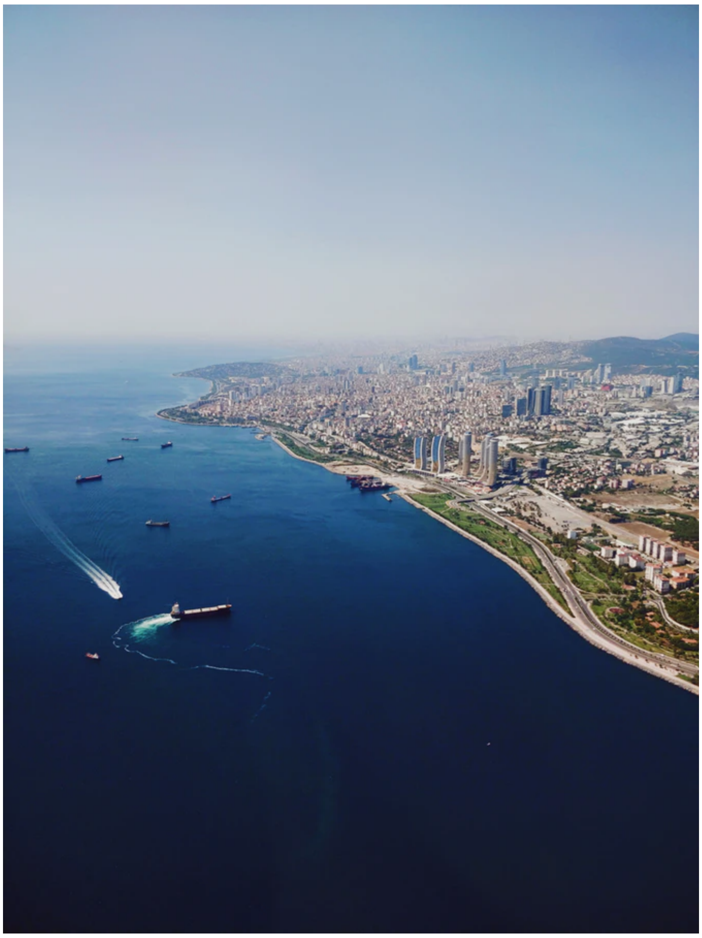
Assistance for Relocation Among Istanbul Districts



IBM Applied Data Science Capstone
Project Report

Morteza Ghorbani Kari
2019

I. Introduction

In this section, we lay out the foundations for understanding the project and what it is going to accomplish and who is going benefit from the results. Data science problems always target an audience and are meant to help a group of stakeholders solve a problem, so we explicitly describe our audience and why they would care about this problem.

I.1 Business Problem

Living in a metropolis has its challenges. And one of those is finding a suitable residence based on one's needs and financial abilities. In periods of inflation and when the cost of housing rises, people (especially those who rent) are forced to relocate to cheaper areas; So, it's an important problem to be addressed with the tools data science provides for us.

With an area of over 5000 square kilometers and a population of over 15 million people, Istanbul is among the largest metropolitan areas in the world. And with a large amount of data available about it, we can easily use data science tools to solve its problems.

People who are considering relocating from their current residence (for economic or other reasons), do not want to lose the convenience of the facilities (venues) close to them, so by giving them a chance to look for similar districts, we can lower the inconvenience of relocation for them.

To conclude, in this project, I am going to study the similarity of districts in Istanbul based on the variety of venues available in them and their housing sale price.

I.2 Target Audience

Two main groups will benefit from the results of this project:

- People/Families who are looking toward similar districts for relocation (based on nearby venues and mean housing sale prices).
- Realtors who want to offer the best options to their customers (the ones with the highest chance to be accepted by their customer).

II. Data

In this section, I describe the data I will be using to execute the idea.

The data used in this project is from the following sources:

- **List of Districts of Istanbul:** Available at the Wikipedia page: List of districts of Istanbul
- **Hurriyetemlak.com:** Where 12-month average per square meter housing sale price for each district was extracted.
- **GeoPy API:** From which coordinates for each district's center were retrieved.
- **FourSquare API:** From which information on venues in each district was retrieved.

Steps taken to collect the data:

- Using **Beautiful Soup**, I scraped the Wikipedia page.
  - Now, the data consists of district names in a DataFrame (A total of 39 districts).
- Using **GeoPy API**, I constructed a function which receives the **district Name** and returns **Latitude** and **Longitude** of the districts' center point.
  - Now, the data consists of 39 district names, Latitudes, and Longitudes in a DataFrame.
- Then, I joined **sale_price** DataFrame and **istanbul_districts.**
- Finally, you can see the first 5 rows of istanbul_districts DataFrame (which has 39 rows and 4 columns (District, Latitude, Longitude and Sale Price)):

	District	Latitude	Longitude	Sale Price
0	Adalar	40.876259	29.091027	5568
1	Arnavutköy	41.184182	28.740729	2265
2	Ataşehir	40.984749	29.106720	5512
3	Avcılar	40.980135	28.717547	2454
4	Bağcılar	41.033899	28.857898	3264

III. Methodology

The original data I will be using in this study is stored in a DataFrame with 4 columns (District, Latitude, Longitude and Sale Price).

Foursquare API needs a **coordinate**, **radius** and a **limit** for the number of returned venues in response to an exploratory query. As described in the previous section I used GeoPy API to find out the coordinate for each district's center point. And I set the **limit to 100** venues to keep the amount of data for the analysis reasonable.

Now, its time to determine the radius parameter.

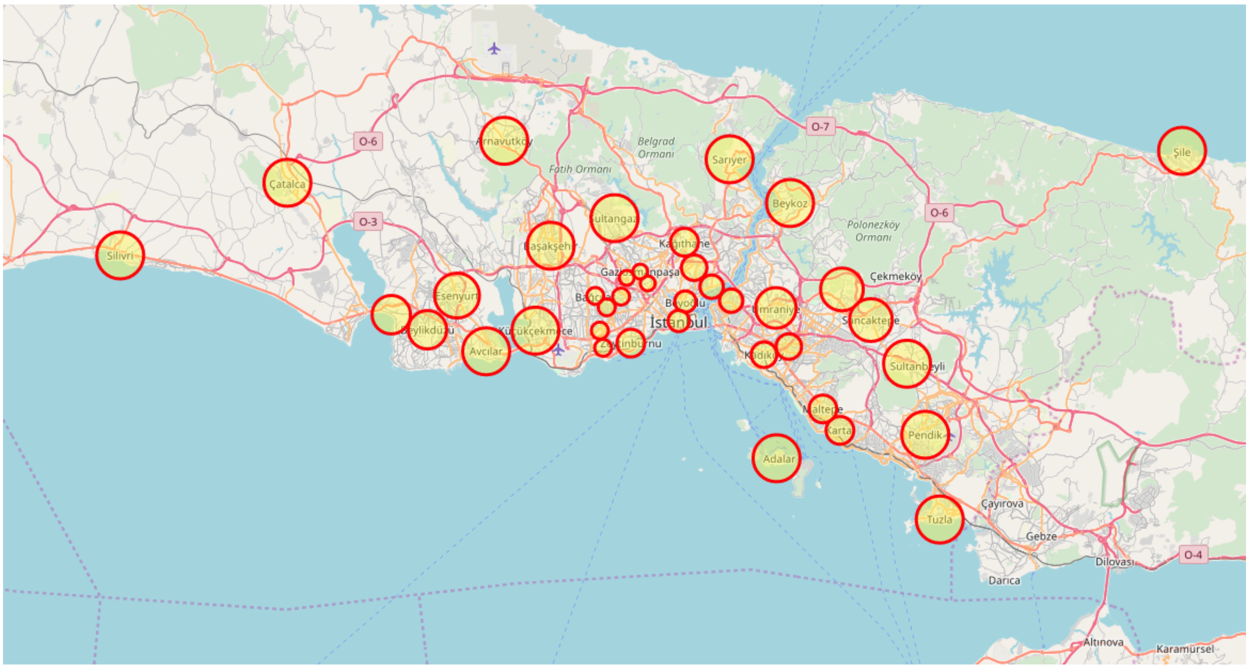
Determining Search Radius

Now that I have the coordinates for districts and have defined my desired limit for the number of venues returned, and just need to decide on the radius of the search. To be exact, I need to decide whether to use a single radius for all districts or use a different radius for each of them.

As the suburban districts are less dense and setting a small radius for them may not cover the whole area, so I decided to use a variable radius for each district.

Based on districts' center points distance from each other, I set the search radius for each district to half of the distance to the closest neighboring district. This number happened to be small for districts located in downtown Istanbul, but was much larger in suburban areas and districts farther from downtown; And because with this measure their search area was too large, I needed to cap the search radius for those districts, so I used 2500 meters as the maximum search radius.

I used python's **Folium** library to visualize the geographic data and illustrate the search area for each district's center point on a map you see below:



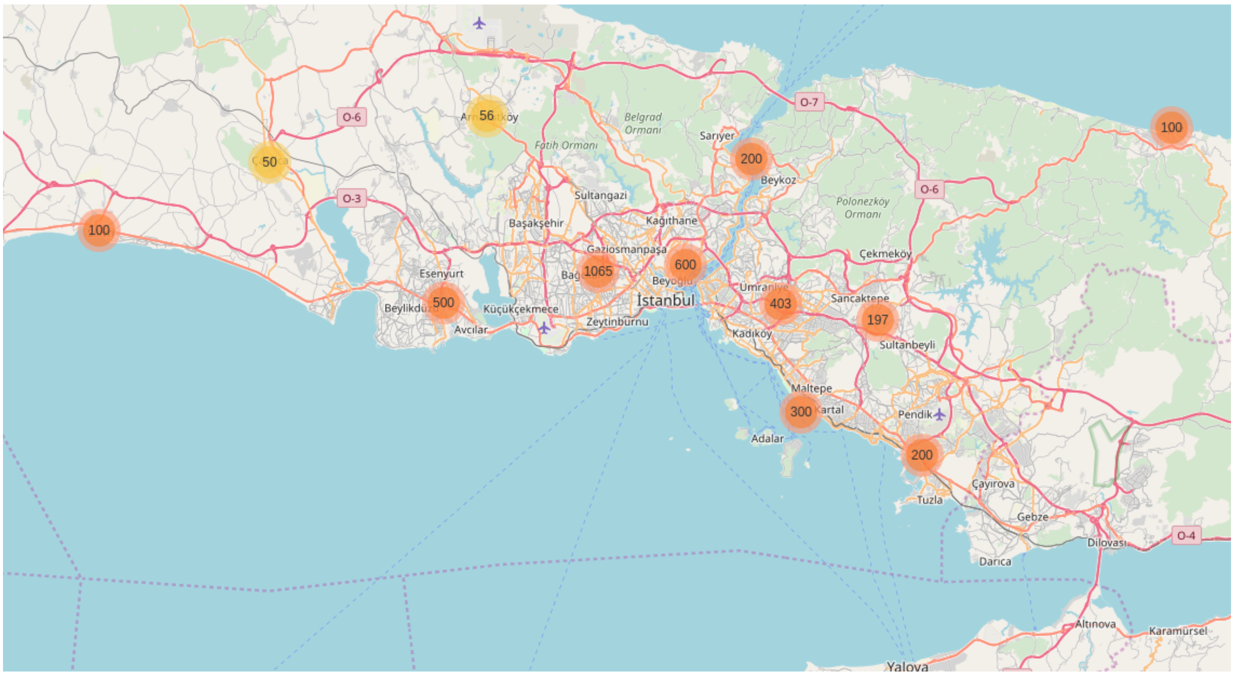
Retrieving Venue Information From FourSquare API

Using FourSquare API, I passed each district's coordinates, radius (which was calculated in the previous step) and limit and received venue information (Venue, Venue Latitude, Venue Longitude, Venue Category) and stored them in a DataFrame to be processed in later steps (Total number of venues was 3771)

Here you can see the first 10 rows of istanbul_venues DataFrame:

	District	District Latitude	District Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Adalar	40.876259	29.091027	İnönü Evi Müzesi	40.878251	29.093647	History Museum
1	Adalar	40.876259	29.091027	L'isola Guesthouse	40.877038	29.096136	Bed & Breakfast
2	Adalar	40.876259	29.091027	Merit Halki Palace Hotel	40.878802	29.090974	Hotel
3	Adalar	40.876259	29.091027	Heybeliada Şafak Askeri Gazino	40.873609	29.099478	Restaurant
4	Adalar	40.876259	29.091027	Heybeliada Su Sporları Kulübü	40.882365	29.089167	Pool
5	Adalar	40.876259	29.091027	Heybeliada Çam Limanı	40.870158	29.084727	Harbor / Marina
6	Adalar	40.876259	29.091027	Farkli Bi' Yer	40.876581	29.100965	Café
7	Adalar	40.876259	29.091027	Luz Café	40.877528	29.097877	Café
8	Adalar	40.876259	29.091027	Erguvan Evyemekleri	40.876864	29.100745	Turkish Restaurant
9	Adalar	40.876259	29.091027	Heybeliada Deniz Lisesi Kolaylık Tesisleri	40.870648	29.097261	Restaurant

Here I used Folium's FastMarkerCluster to superimpose received venues on top of a map of Istanbul:



Prepossessing for data analysis

I performed **one-hot encoding** based on venue categories and calculated the mean occurrence of each category in each district. (**There are 306 unique venue categories**.)

By ordering data of each district by the mean occurrence of categories we can see which categories are the most common in a district. You can see the first 5 of the most common categories in each district below:

	District	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Adalar	Seafood Restaurant	Beach	Café	Restaurant	Turkish Restaurant	Other Great Outdoors	Campground	Fast Food Restaurant	Pool	Ice Cream Shop
1	Arnavutköy	Café	Restaurant	Turkish Restaurant	Gym	Kofte Place	Arcade	Campground	Fish & Chips Shop	Dessert Shop	Bakery
2	Ataşehir	Restaurant	Steakhouse	Hotel	Café	Gym / Fitness Center	Kebab Restaurant	Doner Restaurant	Coffee Shop	Basketball Stadium	Basketball Court
3	Avcılar	Café	Dessert Shop	Coffee Shop	Gym / Fitness Center	Restaurant	Gym	Plaza	Bar	Pizza Place	Pub
4	Bahçelievler	Café	Dessert Shop	Turkish Restaurant	Gym	Restaurant	Ice Cream Shop	Hookah Bar	Breakfast Spot	Fast Food Restaurant	Nail Salon

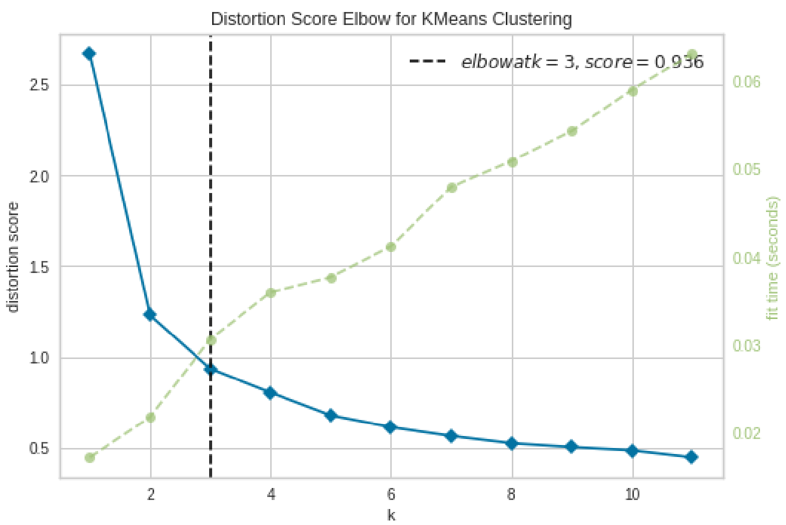
Then, I used **MinMaxScaler** from **sci-kit learn** library to scale prices and added the price column to the one-hot encoded data.

So the data which is going to be fed into the clustering algorithm consists of 39 rows for 39 districts, and 307 features (306 for venue categories and 1 for average sale price).

Clustering

As we assumed that the districts are going to be similar based on the existing venues and their housing prices, I decided to use the **unsupervised clustering algorithm K-Means** to cluster similar districts.

The elbow method was used to determine how many clusters I should use for the algorithm. **KelbowVisualizer** from the **yellowbrick** library was used to visualize this step. As a result, 3 clusters came out as the answer.



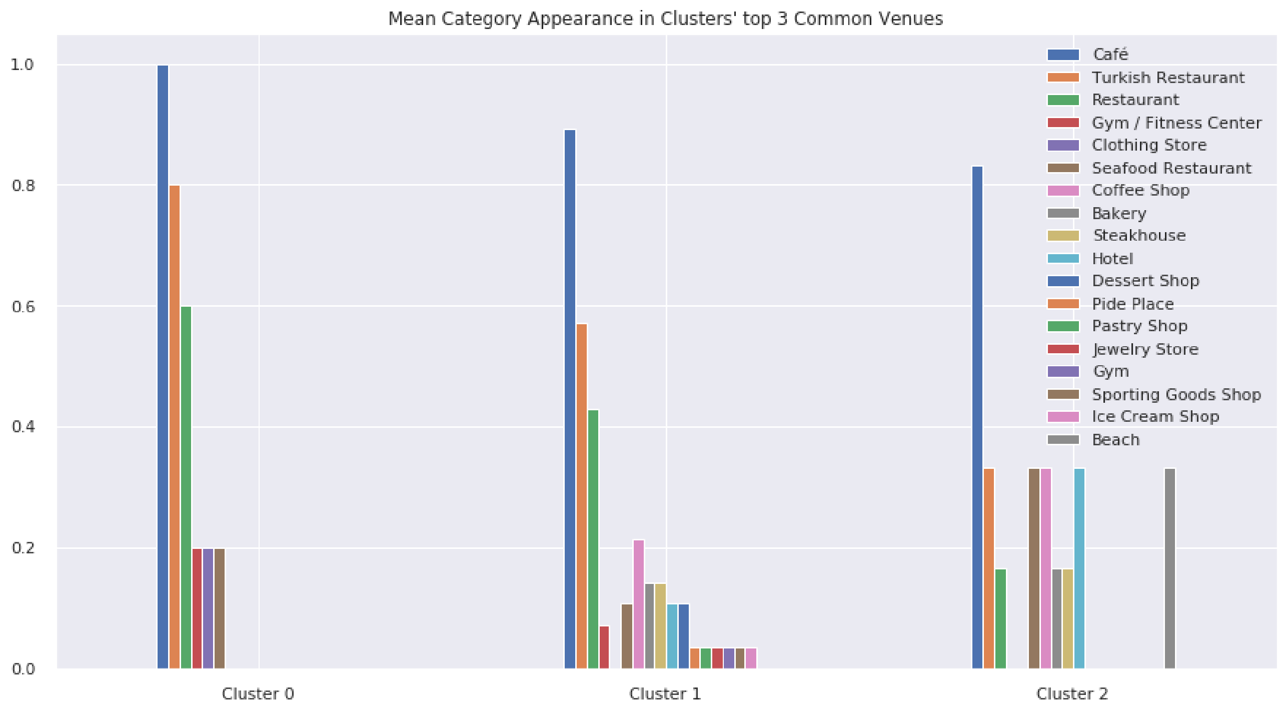
IV. Results

The cluster labels and top 3 venues in each district were appended to the original data so we can use them for visualization in the next step. Here you see the first 5 rows of the final result:

	District	Latitude	Longitude	Sale Price	Distance	Cluster Labels	Top Venues
0	Adalar	40.876259	29.091027	5568	2500.000000	2	Seafood Restaurant, Beach, Café
1	Arnavutköy	41.184182	28.740729	2265	2500.000000	1	Café, Restaurant, Turkish Restaurant
2	Ataşehir	40.984749	29.106720	5512	1409.093554	2	Restaurant, Steakhouse, Hotel
3	Avcılar	40.980135	28.717547	2454	2500.000000	1	Café, Dessert Shop, Coffee Shop
4	Bağcılar	41.033899	28.857898	3264	870.697134	1	Café, Gym, Coffee Shop

Cluster Characteristics

By calculating the mean appearance of each category in districts of each cluster we get the following chart:



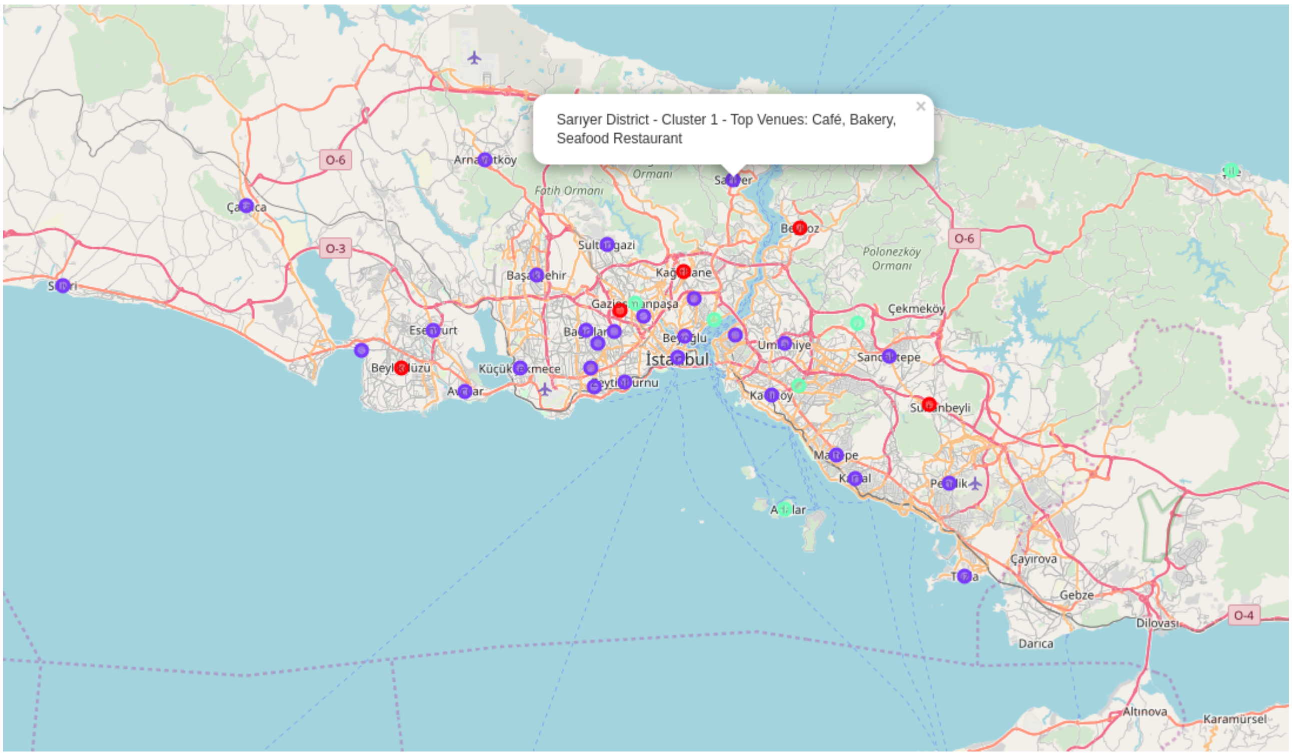
There's a noticeable presence of cafes and restaurants in all of the clusters. The second cluster is very diverse and the last cluster has a more presence of hotels and beaches.

By averaging the housing sale price we get the numbers represented in the following chart:

Now, we can Name the clusters as follows:

- **Cluster 0**: Gastronomic Cluster with Medium Housing Price (5 Districts)
  - Bayrampaşa, Beykoz, Beylikdüzü, Kağıthane, Sultanbeyli
- **Cluster 1**: Diverse Cluster with Low Housing Price (28 Districts)
  - Arnavutköy, Avcılar, Bağcılar, Bahçelievler, Bakırköy, Başakşehir, Beyoğlu, Büyükçekmece, Çatalca, Esenler, Esenyurt, Eyüp, Fatih, Güngören, Kadıköy, Kartal, Küçükçekmece, Maltepe, Pendik, Sancaktepe, Sarıyer, Silivri, Sultangazi, Şişli, Tuzla, Ümraniye, Üsküdar, Zeytinburnu
- **Cluster 2**: Recreational Cluster with High Housing Price (6 Districts)
  - Adalar, Ataşehir, Beşiktaş, Çekmeköy, Gaziosmanpaşa, Şile

Finally, I used Folium to illustrate the final results of the clustering on the map. Clusters are color-coded and the popup on each district shows its name, cluster number, and 3 most common venue categories:



V. Discussion

One of the clear observations in the results was the excessive presence of cafes, restaurants and food-related venues in Istanbul. Of course one of the reasons can be the massive effect of tourism in the recent decade, or it can be the result of data collection bias (supposing Foursquare data mostly consist of recreational and food-related venues).

The results suggest that for every resident living in a district in Istanbul, there are at least 4 other districts that are suitable (based on the parameters of our study.)

In this study, the parameters were limited to housing prices and venue category composition of each district. So, although the findings can be helpful to solve the proposed problem, we still cannot be completely sure about the accuracy of the results.

VI. Conclusion

In the end, I'd like to point to potential future works to be done on this matter, because there are numerous other parameters for assessing the qualities and characteristics of a residence, the accuracy of this study can be improved by adding more features for each district (by data collection or feature engineering) or even using more granular divisions of space (like neighborhoods) in later studies.

There remains a lot of work to be done on this problem, and there are lots of paths to be explored in the future.

Finally, Thank you for reading through the report.

Morteza Ghorbani Kari

17 December 2019