IBM/Coursera Data Science Capstone Project

Segmenting and Clustering Airbnb Listings in Zurich, Switzerland

Georgios Chatzis

June 7, 2019

Contents

1 Introduction/Business Problem

Airbnb [1] is a community where millions of hosts and travellers choose to create a free account so they can list their space and book unique accommodations anywhere in the world.

Although Airbnb listings provide enough information about the shared space, there is less information about the nearby location. For example, travellers might be interested in what kind of venues (supermarkets, restaurants, public transportation, etc.) are close to the accommodation they book. In addition, travellers cannot filter Airbnb listings based on the nearby venues. In other words, each time travellers make a search for an accommodation using the Airbnb community, they may want to get direct information about the venues in the area and a list of similar Airbnb listings with same venue categories nearby.

The main objective of this project is to explore, segment and cluster Airbnb listings in Zurich, Switzerland. I will use the Foursquare API [2] to explore the areas around the Airbnb listings in Zurich. I will use the explore function to get the most common venue categories for each Airbnb listing, and then use this feature together with the prices to group the listings into clusters. I will use the k-means clustering algorithm to complete this task. Finally, I will use the Folium library to visualize the listings in Zurich and their emerging clusters.

2 Data

Based on the problem definition, I will leverage the data from the following data sources to solve the problem:

1. Airbnb Data Collection [3]: Here is the data provided for each Airbnb listing. Each link downloads a zip file of the data for a named city or region; in my case this is Zurich, Switzerland. The zip file holds one or more csv files. Each csv file represents a single "survey" or "scrape" of the Airbnb web site for that city. The data is collected from the public Airbnb web site and the code used is available on GitHub [4]. Each csv file contains the attributes as follows:

   (a) room_id: A unique number identifying an Airbnb listing.

   (b) host_id: A unique number identifying an Airbnb host.

   (c) room_type: One of "Entire home/apt", "Private room", or "Shared room" borough: A sub-region of the city or search area for which the survey is carried out. The borough is taken from a shapefile of the city that is obtained independently of the Airbnb web site. For some cities, there is no borough information; for others the borough may be a number.

   (d) neighbourhood: As with borough: a sub-region of the city or search area for which the survey is carried out. For cities that have both,

a neighbourhood is smaller than a borough. For some cities there is no neighbourhood information.

(e) reviews: The number of reviews that a listing has received. Airbnb has said that 70% of visits end up with a review, so the number of reviews can be used to estimate the number of visits. Note that such an estimate will not be reliable for an individual listing (especially as reviews occasionally vanish from the site), but over a city as a whole it should be a useful metric of traffic.

(f) overall_satisfaction: The average rating (out of five) that the listing has received from those visitors who left a review. accommodates: The number of guests a listing can accommodate. bedrooms: The number of bedrooms a listing offers. price: The price for a night stay. In early surveys, there may be some values that were recorded by month.

(g) minstay: The minimum stay for a visit, as posted by the host.

(h) latitude and longitude: The latitude and longitude of the listing as posted on the Airbnb site: this may be off by a few hundred meters.

(i) last_modified: the date and time that the values were read from the Airbnb web site.

2. Foursquare API: to get the most common venues of given Airbnb listing.

Airbnb data is used to get the coordinates (latitude and longitude), neighbourhood and price for each listing in Zurich, Switzerland. Having this information, I can leverage Foursquare API to explore the areas around the Airbnb listings and get the most common venue categories for each listing. Venue categories together with the price are used to segment the listings into similar clusters.

3 Methodology

3.1 Data Wrangling: Clean and Transform

I scrape the Airbnb Data Collection website in order to obtain the Airbnb data in Zurich and to transform them into a `pandas` dataframe. The result is a dataframe with 7985 rows and 19 columns. The Airbnb dataset contains the same listing more than one with different last modified dates. In other words, the listing in the dataset has been modified several times. We group the listings by **room_id** and keep the latest of the **last_modified**. The result is a `pandas` dataframe with 3359 rows, where each row corresponds to a unique **room_id**. Although the dataset contains listings from years 2016 and 2017, for visualization reasons, I used only Airbnb listings from 2016. This reduces the dataset to 383 listings. Finally, I drop the empty column **name** and remove rows with `NaN` from the dataset. The final dataset has 207 rows and 18 columns.

3.2 Explore and Visualize Airbnb Data

The values in the columns **neighbourhood** and **room_type** are repetitive, meaning that they are categorical attributes. For each category per attribute I count the number of occurrences and plot the results.

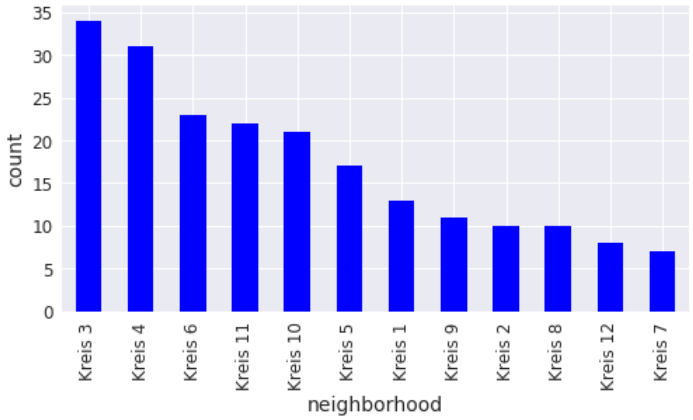

Figure 1: Airbnb listings per neighbourhood in Zurich

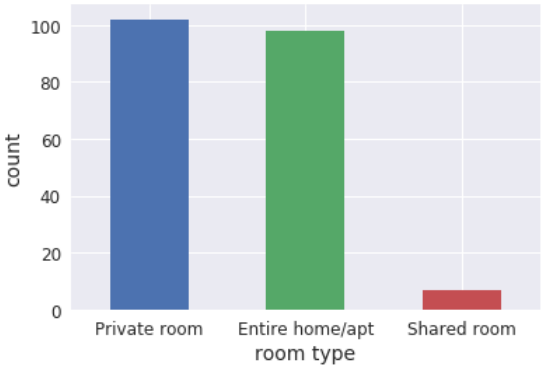

Figure 2: Airbnb listings per room type in Zurich

In addition, I explore the features **price** and **overall_satisfaction**. These two features, besides venues from Foursquare API, will be used to cluster the Airbnb listings. I plot the histograms of both features to take a look into their distributions.

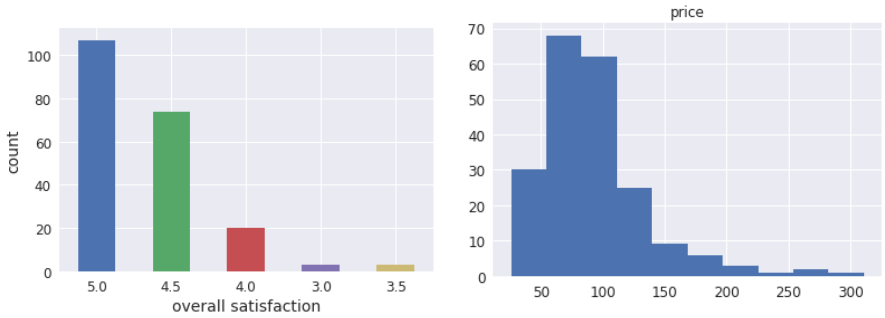

Figure 3: Explore overall satisfaction and price features

Price feature is tail heavy, i.e. it extends much farther to the right of the median than to the left. This may make it a bit harder for some ML algorithms to detect patterns. I will try transforming price attribute to a more bell-shaped distribution.

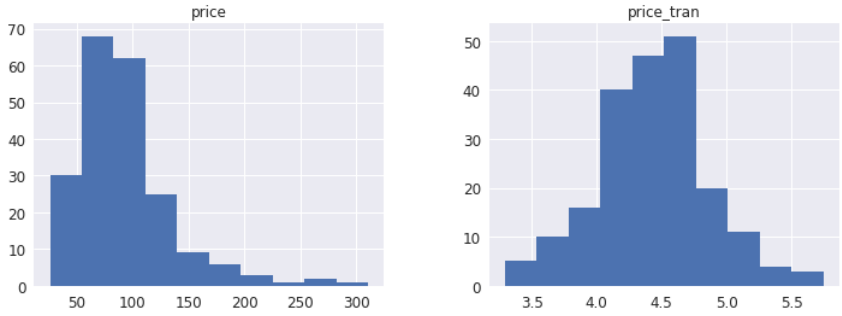

Figure 4: Airbnb listings price transformation

Finally, I drop redundant features from the Airbnb dataset. The final dataframe consists of 6 columns, i.e. **room_id**, **room_type**, **neighbourhood**, **latitude**, **longitude** and **price_tran**.

I create also a map for the different Airbnb listing in Zurich. To do that, I work with `Folium`, a `Python` visualization library, solely developed for visualizing geospatial data. I use `Folium` library to visualize geographic details of Zurich and the Airbnb listings and I create a map of Zurich with Airbnb listings superimposed on top. I use latitude and longitude values to get the map below:

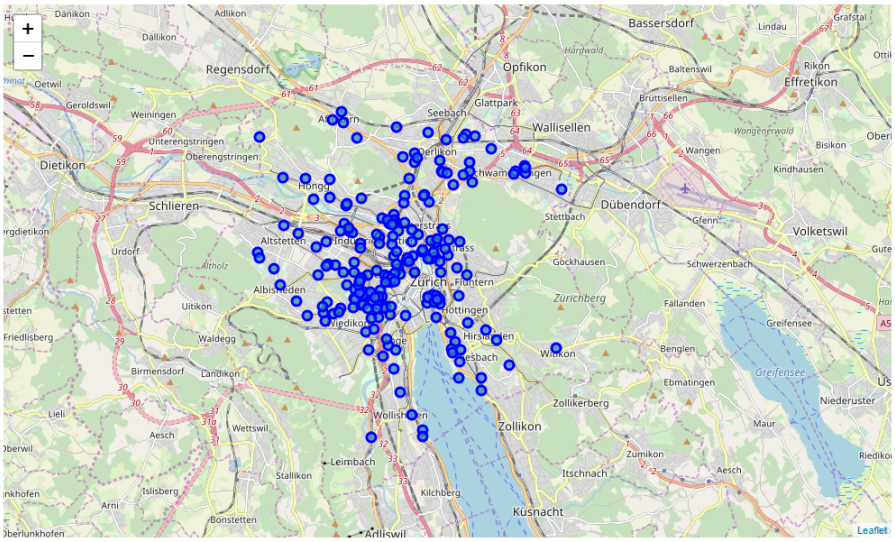

Figure 5: Airbnb listings in Zurich

3.3 Analyse Each Airbnb Listing

I decided to work only with Airbnb listings in the neighbourhoods around Zurich lake, i.e. Kreis 1, 2 and 8, for illustration reasons. The filtered dataframe has 33 listings.

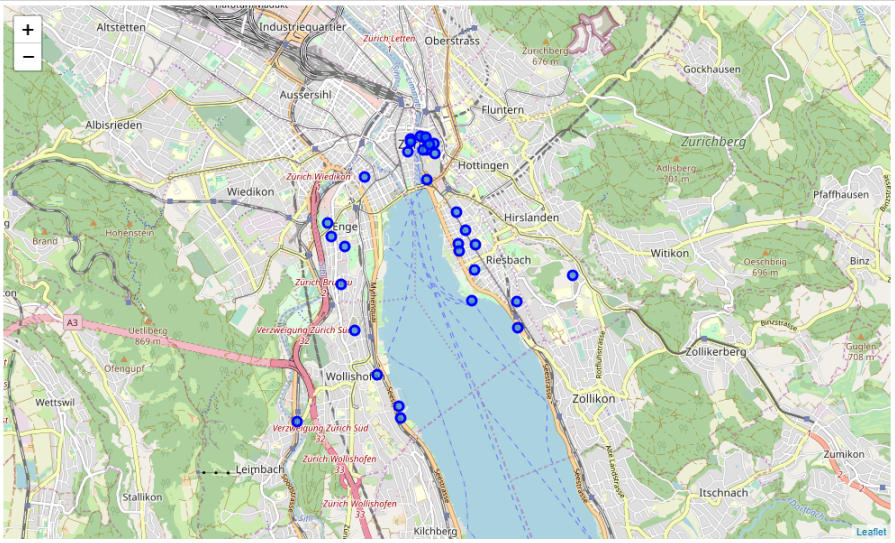

Figure 6: Airbnb listings around Zurich Lake

I utilize the Foursquare API to explore the Airbnb listings and segment them. I set the limit to 100 venues and the radius to 500 meter for each listing from their given latitude and longitude. The response of the Foursquare API is transformed to a dataframe, which later I merge with the Airbnb listings. The result is a dataframe as follows:

	room_id	room_latitude	room_longitude	venue	venue_latitude	venue_longitude	venue_category
0	1034594	47.373248	8.543547	Lindenhof	47.373005	8.540883	Pedestrian Plaza
1	1034594	47.373248	8.543547	Café Schober	47.371400	8.544149	Café
2	1034594	47.373248	8.543547	Schwarzenbach Kolonialwaren	47.371444	8.544091	Gourmet Shop
3	1034594	47.373248	8.543547	Old Crow	47.372092	8.541024	Cocktail Bar
4	1034594	47.373248	8.543547	Neumarkt 17 AG	47.372868	8.546121	Furniture / Home Store

Figure 7: Airbnb listings and venues

Although the number of venues is limited to 100, for some Airbnb listings I got less than 100. I count the number of venues per listing and plot the results.

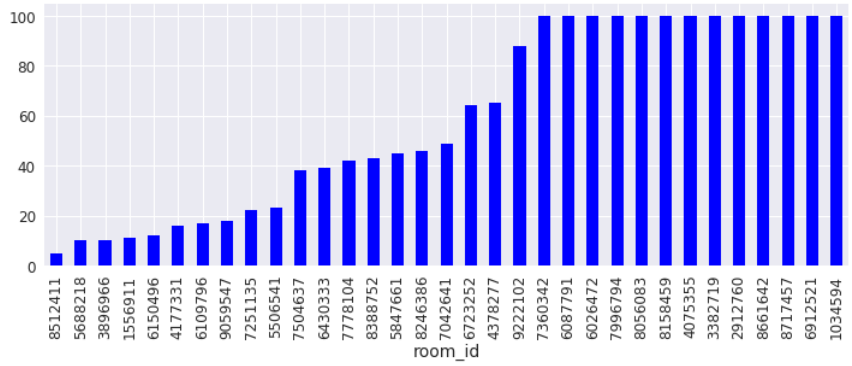

Figure 8: Venues per Airbnb listing

There are 131 unique venues categories. Next, I one-hot encode the feature **venue_category** to convert the categories from text to numerical, and I group rows by **room_id**. By taking the mean I got the frequency of occurrence of each category.

	room_id	American Restaurant	Argentinian Restaurant	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Australian Restaurant	Automotive Shop	BBQ Joint	Bakery	Bar	Bath House	Bathing Area	Beac
0	1034594	0.0	0.01	0.01	0.02	0.01	0.0	0.0	0.00	0.0	0.010000	0.06	0.0	0.0	0.0
1	1556911	0.0	0.00	0.00	0.00	0.00	0.0	0.0	0.00	0.0	0.181818	0.00	0.0	0.0	0.0
2	2912760	0.0	0.00	0.01	0.02	0.00	0.0	0.0	0.00	0.0	0.010000	0.06	0.0	0.0	0.0
3	3382719	0.0	0.02	0.00	0.02	0.01	0.0	0.0	0.01	0.0	0.010000	0.04	0.0	0.0	0.0
4	3896966	0.0	0.00	0.00	0.00	0.00	0.0	0.0	0.00	0.0	0.000000	0.00	0.0	0.0	0.0

Figure 9: Frequency of occurrence of each category

Finally, I create a new dataframe and display the top 10 venues for each Aribnb listing.

	room_id	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	1034594	Swiss Restaurant	Café	Bar	French Restaurant	Italian Restaurant	Hotel	Restaurant	Lounge	Cocktail Bar	Diner
1	1556911	Bakery	Swiss Restaurant	Wine Shop	Café	Modern European Restaurant	Grocery Store	Medical Center	Museum	Tram Station	Fast Food Restaurant
2	2912760	Swiss Restaurant	Café	French Restaurant	Bar	Hotel	Italian Restaurant	Restaurant	Cocktail Bar	Plaza	Lounge
3	3382719	Swiss Restaurant	Café	Boutique	Restaurant	Cocktail Bar	Hotel	French Restaurant	Italian Restaurant	Bar	Jewelry Store
4	3896966	Bus Station	Harbor / Marina	Swiss Restaurant	Campground	Pool	Restaurant	Cupcake Shop	Department Store	Dessert Shop	Diner

Figure 10: Listings with top 10 categories

3.4 Prepare Data for Clustering

I merge the frequency of occurrence of each category dataframe with the attributes **price_tran** and **overall_satisfaction** by **room_id**, and I get the final dataset, where I apply later the `KMeans()` algorithm.

	room_id	price_tran	overall_satisfaction	American Restaurant	Argentinian Restaurant	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Australian Restaurant	Automotive Shop	BBQ Joint	Bak
0	1034594	4.890349	4.0	0.0	0.01	0.01	0.02	0.01	0.0	0.0	0.00	0.0	0.010
1	1556911	4.430817	4.5	0.0	0.00	0.00	0.00	0.00	0.0	0.0	0.00	0.0	0.181
2	2912760	5.337538	4.5	0.0	0.00	0.01	0.02	0.00	0.0	0.0	0.00	0.0	0.010
3	3382719	5.105945	4.5	0.0	0.02	0.00	0.02	0.01	0.0	0.0	0.01	0.0	0.010
4	3896966	4.543295	5.0	0.0	0.00	0.00	0.00	0.00	0.0	0.0	0.00	0.0	0.000

Figure 11: Final Dataset

Finally, I normalize the dataset. Normalization is a statistical method that

helps mathematical-based algorithms interpret features with different magnitudes and distributions equally. I use `StandardScaler()` from `Scikit-Learn` [5] to normalize the dataset.

3.5 Cluster Airbnb Listings

I run `KMeans()` from `Scikit-Learn` to cluster the listings into clusters. I cannot simply take the value of k that minimizes the inertia, since it keeps getting lower as we increase k. Indeed, the more clusters there are, the closer each instance will be to its closest centroid, and therefore the lower the inertia will be. However, we can plot the inertia as a function of k and analyse the resulting curve:

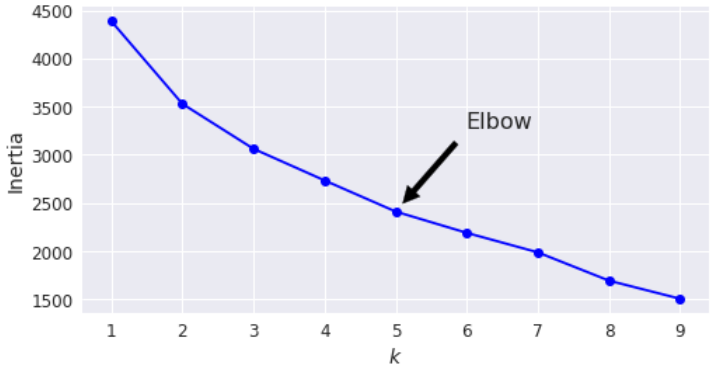

Figure 12: Inertia per number of clusters

As you can see, after $k = 5$ the decrease in the inertia is less steep as for smaller values of k. So $k = 5$ is a pretty good choice.

4 Results

I add cluster labels to my final dataset and create a map of Zurich with the clusters.

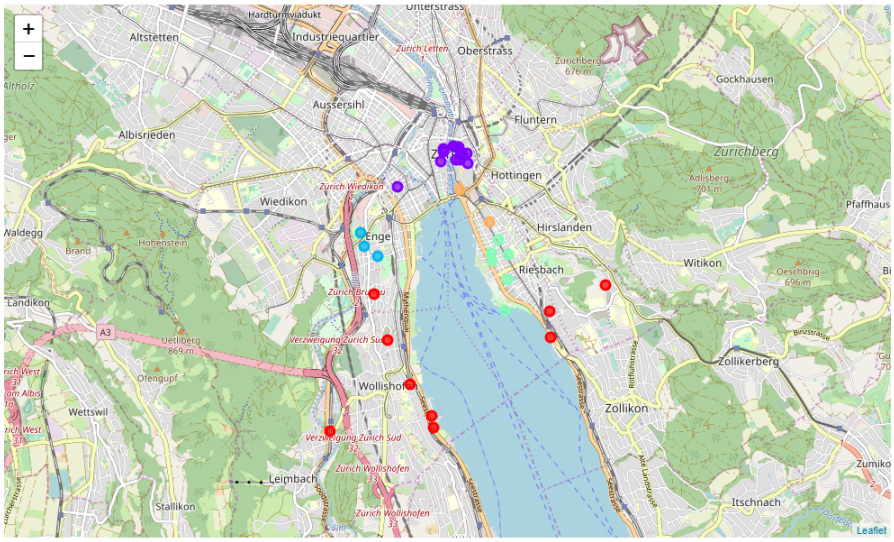

Figure 13: Clusters

The number of listings in each cluster differs.

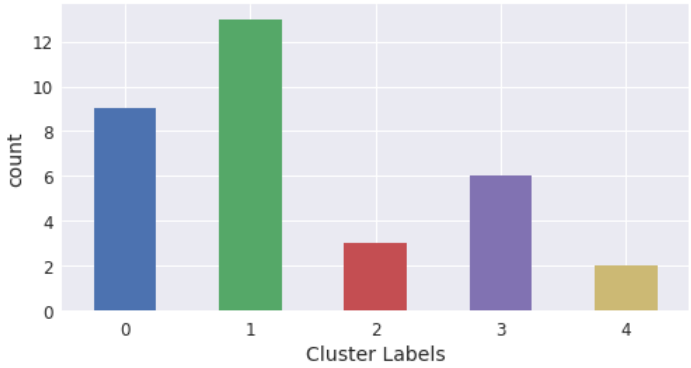

Figure 14: Number of listings in each cluster

5 Discussion and Recommendations

k-means partitioned the Airbnb listings into 5 groups since we specified the algorithm to generate 5 clusters. The Airbnb listings in each cluster are similar to each other in terms of the features included in the dataset.

I check the centroids values by averaging the features and get the top most common venues in each cluster.

Table 1: Clusters with top 5 venues

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
price	medium	medium	low	medium	high
satisfaction	4.67/5	4.69/5	5/5	4.5/5	4.5/5
venue 1	Bus Station	Swiss Rst	Italian Rst	Italian Rst	Hotel
venue 2	Swiss Rst	Cafe	Hotel	Other Rst	Italian Rst
venue 3	Other Rst	Bar	Other Rst	Hotel	Cafe
venue 4	Tram	French Rst	Tram	Cafe	Swiss Rst
venue 5	Harbor	Hotel	Swiss Rst	Bakery	Bar

Now we can create a profile for each group, considering the common characteristics of each cluster. For example, the 5 clusters can be:

1. **CLUSTER 0**: Medium price rooms, overall satisfaction 4.67/5 with Bus station, Swiss Restaurant, Restaurant, Tram Station and Harbor/Marina nearby
2. **CLUSTER 1**: Medium price rooms, overall satisfaction 4.69/5 with Swiss Restaurant, Cafe, Bar, French Restaurant and Hotel nearby
3. **CLUSTER 2**: Low price rooms, overall satisfaction 5/5 with Italian Restaurant, Hotel, Restaurant, Tram Station, Swiss Restaurant nearby
4. **CLUSTER 3**: Medium price rooms, overall satisfaction 4.5/5 with Italian Restaurant, Restaurant, Hotel, Cafe and Bakery nearby
5. **CLUSTER 4**: High price rooms, overall satisfaction 4.5/5 with Hotel, Italian Restaurant, Cafe, Swiss Restaurant and Bar nearby

6 Conclusion

I have combined Airbnb listings and Foursquare data to provide useful information to travelers in Zurich about the location and the most common venues they can visit in an area of 500 meters around their accommodation.

I have grouped the Airbnb listings around Zurich lake into 5 clusters based on similar venues, prices and overall satisfaction. Travelers could leverage the clusters to filter listings according to their price preferences and the most common venues. In other words, travelers could search Airbnb listings according to location or venues they would like to visit, close to their accommodation.

The project is available on GitHub [6]

References

[1] https://www.airbnb.com/

[2] https://developer.foursquare.com/

[3] http://tomslee.net/airbnb-data-collection-get-the-data

[4] https://github.com/tomslee/airbnb-data-collection

[5] https://scikit-learn.org/stable/

[6] https://github.com/georchat/ibm-coursera-capstone-project