IBM Data Science Capstone
Finding a Compatible Neighborhood

Jeffrey T. Barton

April 8, 2020

1 Introduction

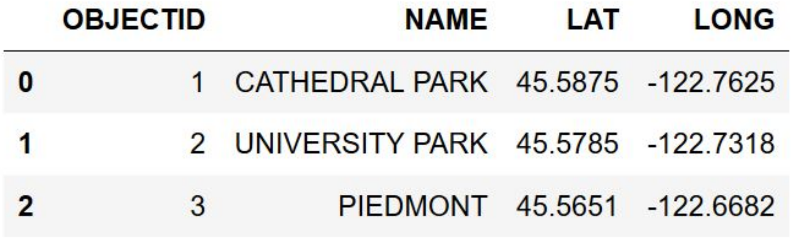
In this project we seek to provide a user who is a newcomer or thinking of moving to Portland, Oregon, with a list of promising neighborhoods in which to focus their housing search. The user will be asked to provide a list of desired amenities in a neighborhood in order of preference, and the "app" will output a list of Portland neighborhoods that are considered to be similar to the neighborhood described by the user.

2 Data

The data required will be a list of Portland neighborhoods and their corresponding latitudes and longitudes. From that location data we are able to find relevant information about the types of venues that are located in or near each location, including bars, restaurants, museums, grocery stores, etc. Though not a necessity for the project, for mapping purposes we also make use of a geojson file of the neighborhood boundaries. Both the list of neighborhoods and the geojson file can be freely downloaded at
http://gis-pdx.opendata.arcgis.com/datasets/neighborhoods-regions.

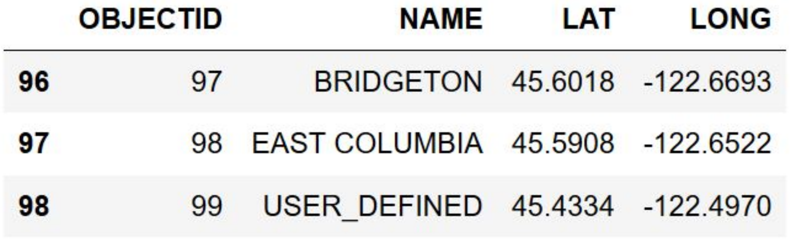
Unfortunately, latitudes and longitudes for each neighborhood were not readily available, so those were found through simple web searches and entered into a csv file by hand. Figure 1 shows the beginning of the resulting dataframe for neighborhoods.

Because we want to determine neighborhoods that are most compatible with a user's neighborhood preferences, we include in the csv file (and resulting dataframe) a row that represents a "stand in" neighborhood for the user's preferences. This row is found at the bottom of the dataframe as shown in Figure 2. The latitude and longitude are selected so that the map marker for the user's neighborhood will be an isolated point in the southeast corner of the map.



	OBJECTID	NAME	LAT	LONG
0	1	CATHEDRAL PARK	45.5875	-122.7625
1	2	UNIVERSITY PARK	45.5785	-122.7318
2	3	PIEDMONT	45.5651	-122.6682

Figure 1: Neighborhoods of Portland and Location



	OBJECTID	NAME	LAT	LONG
96	97	BRIDGETON	45.6018	-122.6693
97	98	EAST COLUMBIA	45.5908	-122.6522
98	99	USER_DEFINED	45.4334	-122.4970

Figure 2: User Neighborhood and Location

As we can see from the figures, there are 98 distinct neighborhoods in Portland that will make up our set to compare with the user's preferences.

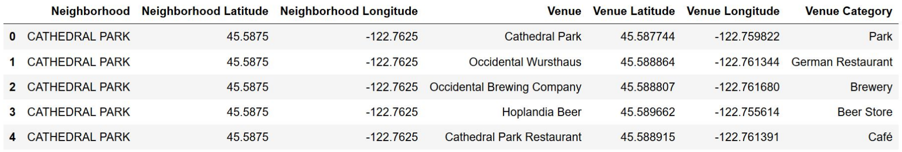
3 Methodology

Once in possession of our list of neighborhoods and their locations, we use Foursquare to retrieve relevant information about the venues available in each neighborhood. Of course we do not send the location of the fictitious user neighborhood during this stage.

3.1 Results from Foursquare

Our Foursquare query allows us to set a limit on the number of results returned for each neighborhood along with a search radius from each location. Initially we set the limit at 100 and the radius at 600 meters, though these can be experimented with later. We query Foursquare for all venues near each neighborhood's latitude and longitude within the specified limits. The figure of partial results below shows that the only information we keep is the venue, its location, and its category.

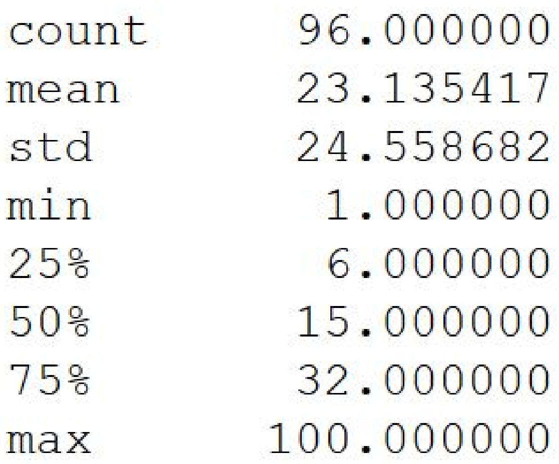
A quick check reveals that our search resulted in 293 unique venue categories. (This number will change depending on the limits set and the time at which the



	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	CATHEDRAL PARK	45.5875	-122.7625	Cathedral Park	45.587744	-122.759822	Park
1	CATHEDRAL PARK	45.5875	-122.7625	Occidental Wursthaus	45.588864	-122.761344	German Restaurant
2	CATHEDRAL PARK	45.5875	-122.7625	Occidental Brewing Company	45.588807	-122.761680	Brewery
3	CATHEDRAL PARK	45.5875	-122.7625	Hoplandia Beer	45.589662	-122.755614	Beer Store
4	CATHEDRAL PARK	45.5875	-122.7625	Cathedral Park Restaurant	45.588915	-122.761391	Café

Figure 3: Foursquare Venue Information

query is sent.) The basic descriptive statistics below give an indication of how many venues are typically present in one of our neighborhoods.

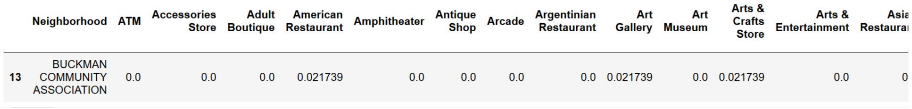


```
count     96.000000
mean      23.135417
std       24.558682
min        1.000000
25%        6.000000
50%       15.000000
75%       32.000000
max      100.000000
```

Figure 4: Statistics on Number of Venues

We note that the count being equal to 96 indicates that two neighborhoods were not returned any venues.

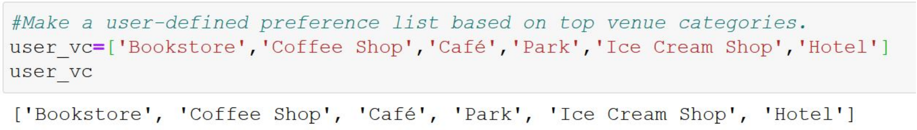
Our next step is to consolidate all venue information by neighborhood so that we get an overall picture of what is available. We do this by grouping our Foursquare output by neighborhood and calculating the mean number of each type of venue in each neighborhood. The result gives us the proportions of each type of venue that are present in each venue. The figure below shows an example result.



	Neighborhood	ATM	Accessories Store	Adult Boutique	American Restaurant	Amphitheater	Antique Shop	Arcade	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	Arts & Entertainment	Asia Restaura
13	BUCKMAN COMMUNITY ASSOCIATION	0.0	0.0	0.0	0.021739	0.0	0.0	0.0	0.0	0.021739	0.0	0.021739	0.0	0

Figure 5: Venue Proportions

From the collection of venue categories we print out a list of all categories so that the user can choose which ones to include in their list of desired amenities in a neighborhood. Our Jupyter notebook prompts the user to enter an ordered list of venue category preferences (as many as they choose). An example list is below.

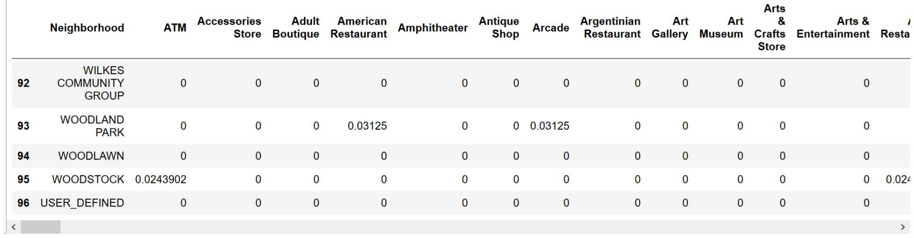


```
#Make a user-defined preference list based on top venue categories.
user_vc=['Bookstore','Coffee Shop','Café','Park','Ice Cream Shop','Hotel']
user_vc
```

```
['Bookstore', 'Coffee Shop', 'Café', 'Park', 'Ice Cream Shop', 'Hotel']
```

Figure 6: Sample User Preference List

We then use that list to generate a score for each category based on where that category fell in the user's list. The scores are then scaled so that they sum to one. In this way we create category weights, or "proportions," that are comparable to those generate by our Foursquare calls. We then append the user's neighborhood proportions to the dataframe of the Foursquare data. A partial result of the final dataframe is shown below.



	Neighborhood	ATM	Accessories Store	Adult Boutique	American Restaurant	Amphitheater	Antique Shop	Arcade	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	Arts & Entertainment	Resta
92	WILKES COMMUNITY GROUP	0	0	0	0	0	0	0	0	0	0	0	0	
93	WOODLAND PARK	0	0	0	0.03125	0	0	0.03125	0	0	0	0	0	
94	WOODLAWN	0	0	0	0	0	0	0	0	0	0	0	0	
95	WOODSTOCK	0.0243902	0	0	0	0	0	0	0	0	0	0	0	0.024
96	USER_DEFINED	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 7: Neighborhood Venue Summary Including User's

3.2 Segmenting Neighborhoods

Once we have a single dataframe that includes Portland's real neighborhoods plus our user-generated neighborhood, after standardizing the data we can send the entire dataframe to a clustering algorithm where the neighborhoods will be segmented into similar clusters. We examined both K-means clustering and DBSCAN, but K-means generated more reasonable results. Before deciding on the number of clusters, K, to specify we calculated the inertia for a range of K. The inertia is a measure of how well the algorithm divided the data set based on mean squared distances of data points to their centroids. A lower inertia score indicates better performance, though we have to be aware that increasing K *always* decreases inertia. Thus we are looking for a balance between a low inertia and a useful number of clusters. The figure below shows the inertia of the K-means algorithm as a function of the number of clusters, K.

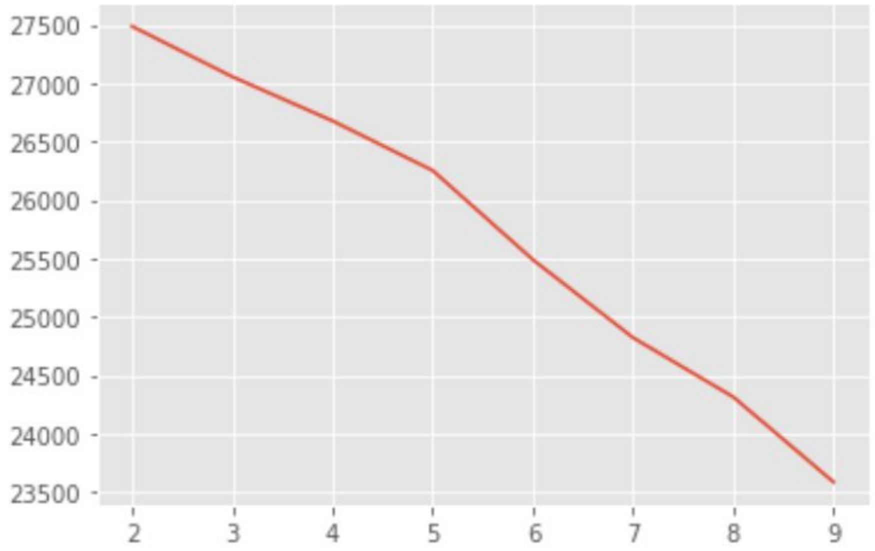


Figure 8: Inertia vs. K

The presence of an "elbow" at $K = 7$ leads us to select $K = 7$ for the number of clusters in this project. Once the clustering algorithm has finished, the cluster that contains the user-defined neighborhood will provide the user with a list of promising neighborhoods with which to begin their housing search.

4 Results

4.1 Mapping the Clusters

Once we have received the cluster labels from the clustering algorithm, we append them to our dataframe for the purposes of mapping the results. We provide the user with a map that makes it easy to locate neighborhoods with amenities similar to their preferences. Below are examples (one with the neighborhood boundary layer, and one without). The color of the user's neighborhood is shown by the marker in the southeast, or bottom-right, corner.

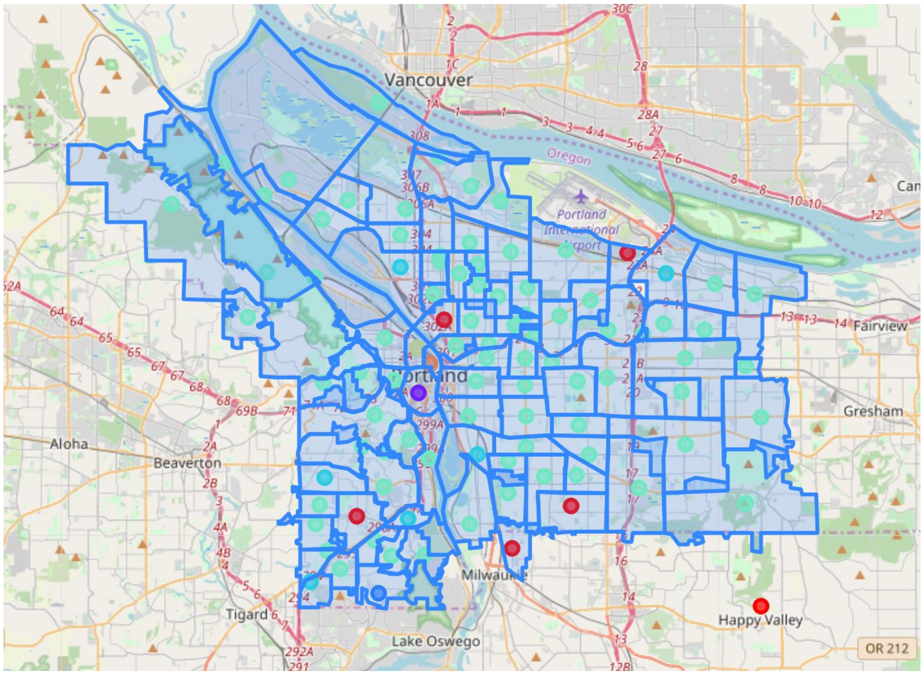


Figure 9: Neighborhood Clusters, K=5, with Boundaries

4.2 The User's Most Promising Neighborhoods

Along with the graphical output, the user is also presented with a dataframe that includes all of the neighborhoods in the cluster containing their own as well as the top ten most common venues in each. A sample of such a list is shown below.

This section outputs all of the neighborhoods in the cluster to which the user's neighborhood belongs.

5 Discussion

The mapped results are representative of a common issue when using this method – that there is often a very large cluster of neighborhoods. Of course it is of limited usefulness to get a list of "promising neighborhoods" that includes the majority of Portland's 98. This is especially problematic when even a cursory glance at the top 10 types of venue for each neighborhood seems to indicate that they are very dissimilar. In this case, the user's neighborhood was clustered with only 5 other neighborhoods, which is a useful size. However, it is often the case that the user neighborhood is placed in a much larger cluster.

One can experiment with different parameters to try to improve the results by limiting the size of the largest cluster. For example, a different search radius,

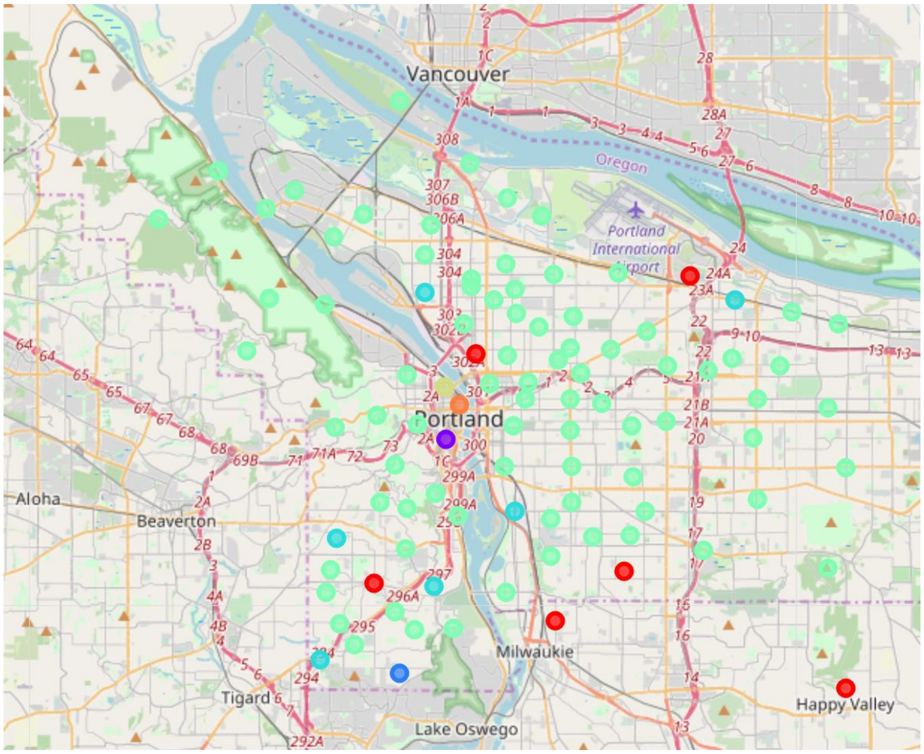
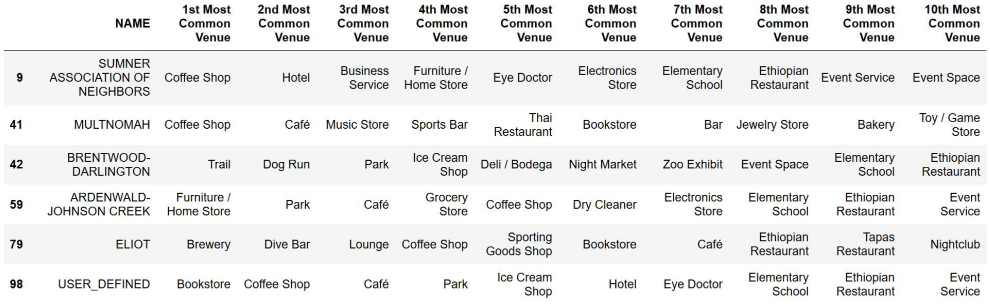


Figure 10: Neighborhood Clusters, K=5, without Boundaries



	NAME	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
9	SUMNER ASSOCIATION OF NEIGHBORS	Coffee Shop	Hotel	Business Service	Furniture / Home Store	Eye Doctor	Electronics Store	Elementary School	Ethiopian Restaurant	Event Service	Event Space
41	MULTNOMAH	Coffee Shop	Café	Music Store	Sports Bar	Thai Restaurant	Bookstore	Bar	Jewelry Store	Bakery	Toy / Game Store
42	BRENTWOOD-DARLINGTON	Trail	Dog Run	Park	Ice Cream Shop	Deli / Bodega	Night Market	Zoo Exhibit	Event Space	Elementary School	Ethiopian Restaurant
59	ARDENWALD-JOHNSON CREEK	Furniture / Home Store	Park	Café	Grocery Store	Coffee Shop	Dry Cleaner	Electronics Store	Elementary School	Ethiopian Restaurant	Event Service
79	ELIOT	Brewery	Dive Bar	Lounge	Coffee Shop	Sporting Goods Shop	Bookstore	Café	Ethiopian Restaurant	Tapas Restaurant	Nightclub
98	USER_DEFINED	Bookstore	Coffee Shop	Café	Park	Ice Cream Shop	Hotel	Eye Doctor	Elementary School	Ethiopian Restaurant	Event Service

Figure 11: Promising Neighborhoods with Top Venues

a preference list with more or fewer items, a different method for weighting preferences, or a different K for the K-means algorithm could all lead to better results.

However, a more fundamental issue is likely the sheer number of unique categories. With 293 different categories, and each neighborhood only including an average of 23 venues, comparisons will be difficult due to having a relatively sparse dataframe to send to the clustering algorithm.

6 Conclusion

The process in this project sometimes generates useful results, i.e. a small cluster of similar neighborhoods that includes the user's. However it is too often the case that the user's neighborhood lands in a cluster that includes nearly all neighborhoods. On the other hand, occasionally the user's neighborhood ends up in a category by itself, which is also problematic. Future work could include narrowing down the number of unique venue categories by consolidating them into broader ones, and the expectation is that this would be a more fruitful avenue that simply tinkering with different parameters.