World City Clustering

1 Introduction

I am interested in a classification of the big cities in the world (with population at least $100,000$) based on the food venues that are available. This seems to be an interesting thing on its own to see how cities differ in various parts of the world, but it can also be of practical interest.

For instance one might want to open a restaurant in a certain city. Then we can look in which class this particular city falls and compare it to the average city in this class. If some food venue, say "Italian Restaurant", is underrepresented in the chosen city, this might indicate that it could be a good idea to open such a restaurant.

2 Data

The first task was to get a list of cities with a population of at least $100,000$. I could find such a list in the demographic yearbook of the United Nations. It lists the cities for many countries, but some bigger ones like China, Canada and Australia are missing. Nevertheless it is a good starting point and sufficient for our purposes.

For the cities we will then need to find the corresponding latitude and longitude coordinates. This is done with the OpenCage Geocoder API. Some cities in the list couldn't be found due to wrongly spelled names in the UN report. These city names were corrected by hand.

Given the coordinates we can then use the foursquare API to obtain food venues in each city. Cities for which the API didn't find any or only very few (< 5) venues are treated as outliers and are removed. Also the not very specific food category "Restaurant" is removed and only the more specific categories like "Italian Restaurant", "Spanish Restaurant" etc. are kept. The data then looks like

	Camacari	Ampang	Cleveland
City Latitude	-12.6998	3.1896	41.5052
City Longitude	-38.3261	101.6994	-81.6934
Venue	Casa de Taipa	Mail Mee Rebus	Subway
Venue Latitude	-12.6951	3.1891	41.5001
Venue Longitude	-38.3284	101.6968	-81.6975
Venue Category	Brazilian Rest.	Noodle House	Sandwich Place

3 Methodology

Since there are only a limited amount of calls possible for the APIs we use we restrict ourselfs to a randomly chosen subset of 500 cities from the original dataset. For a visualization of the cities we can use the Folium package:

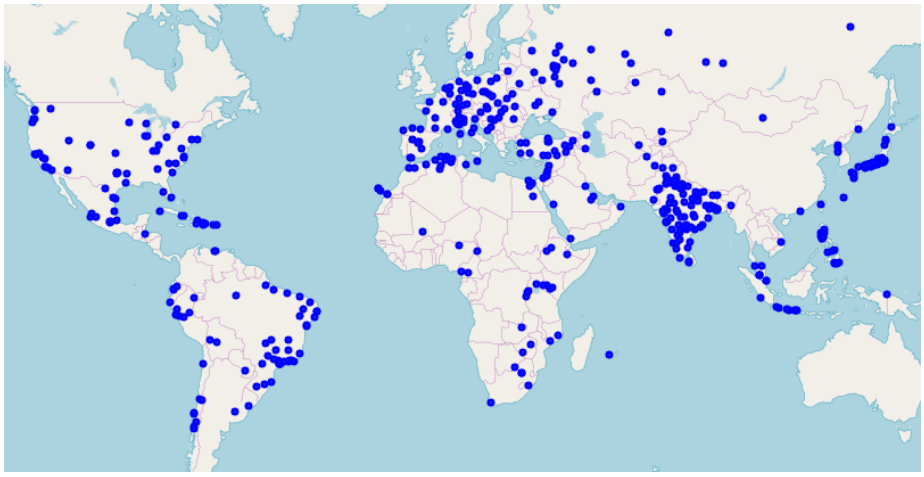

Figure 1: World Map

Given the cities we can now calculate how many different food venues exist in the cities. We find a value of 201 unique food venue categories. With this category variable we use one hot encoding meaning that each of the 201 categories gets it's own column. Furthermore we would like to know how often each category appears in each city percentagewise. So for all the cities we obtain data vectors that looks like

	Aachen	Abohar
Acai House	0.0000	0.0000
Afghan Restaurant	0.0000	0.0000
African Restaurant	0.0000	0.1429
Alsatian Restaurant	0.0000	0.0000
American Restaurant	0.0100	0.0000
⋮	⋮	⋮

This means that $14,29\%$ of the food venues in Abohar are African restaurants, 1% of the food venues in Aachen are American restaurants etc.

Based on these percentages we want to train our k-means classifier. Clearly we need to know into how many clusters we want to partition our dataset. One way to approach this problems is to use an appropriate metric to evaluate how well the data is clustered for a given classifier.

Here we use the so-called Silhouette score: For each datapoint i we calculate two values. The first one $a(i)$ is the average (euklidean) distance to all the datapoints

in the same cluster. The smaller $a(i)$, the better i fits into the cluster. Then we calculate the average distance of i to all the datapoints in a clusters that doesn't contain i. We do this for all the other clusters and call the smallest of those calculated values $b(i)$. The larger $b(i)$ is, the clearer i does not belong to any other cluster. Now we define the Silhouette score for this single point i to be

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}.$$

By definition $s(i)$ is between -1 and 1. Higher values indicate better fits. The overall Silhouette score is then just the average of the Silhouette scores for each data point.

With our data we obtain the following.

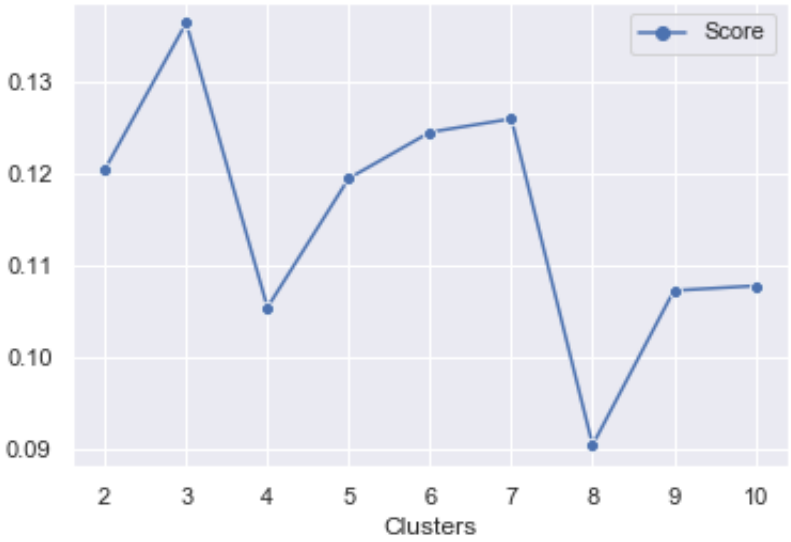

Figure 2: Silhouette Score

The highest score is at $k = 3$ clusters. But with only that few clusters we might be underfitting so I decided to go for the next local maximum at $k = 7$ clusters and train the k-means algorithm with 7 clusters.

4 Results

A nice way to show the clusters is again by taking a picture of the world map and use a different color for each cluster.

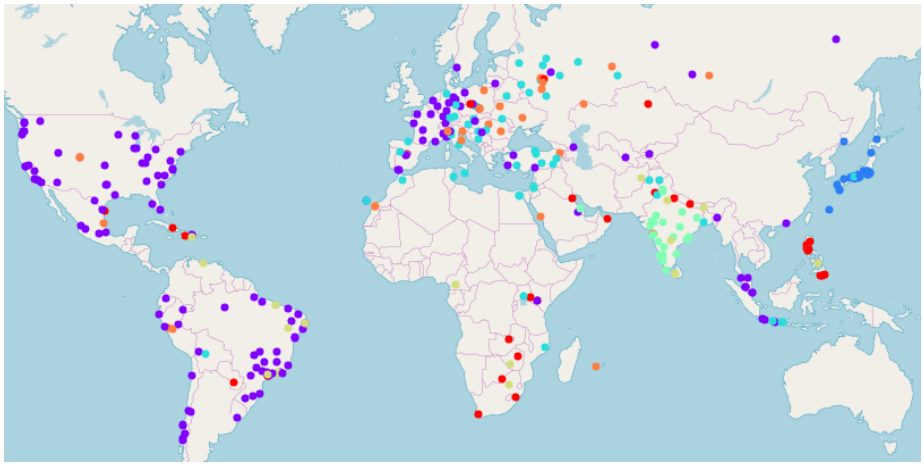

Figure 3: Clusters

It is already visible that most of the cities in America and western Europe are in the same cluster. Also both India and Japan seem to have more or less their own clusters. Eastern Europe and Russia are more mixed. The cities in southern Africa also mostly tend to belong to the same cluster.

It is of course interesting to see what food venues we can find in the different clusters. We just look at the top 5 most common venues in each cluster, more precicely on the percentage that each of those 5 food categories appears. The following table summarizes these things, where each column corresponds to one of the clusters. The entries in the table are rounded and should be read as percentages.

	0	1	2	3	4	5	6
Pizza Place	7	5	0	5	4	8	24
Italian Restaurant	0	4	4	4	0	0	4
Ramen Restaurant	0	0	12	0	0	0	0
Indian Restaurant	0	0	0	0	32	2	0
Bakery	2	4	0	4	4	28	3
Japanese Restaurant	0	0	14	0	0	0	0
Sandwich Place	0	4	0	0	2	0	0
Asian Restaurant	3	0	0	2	6	0	0
BBQ Joint	7	0	4	0	0	3	0
Café	8	7	8	33	6	3	15
Snack Place	0	0	0	0	8	6	0
Burger Joint	3	4	0	1	0	0	4
Chinese Restaurant	3	0	5	0	0	4	0
Fast Food Restaurant	3	3	5	3	3	7	5

In order to compare these results with the colored clusters in the map it is nice to show the data from the table in form of a bar chart using the same colors as in the world map.

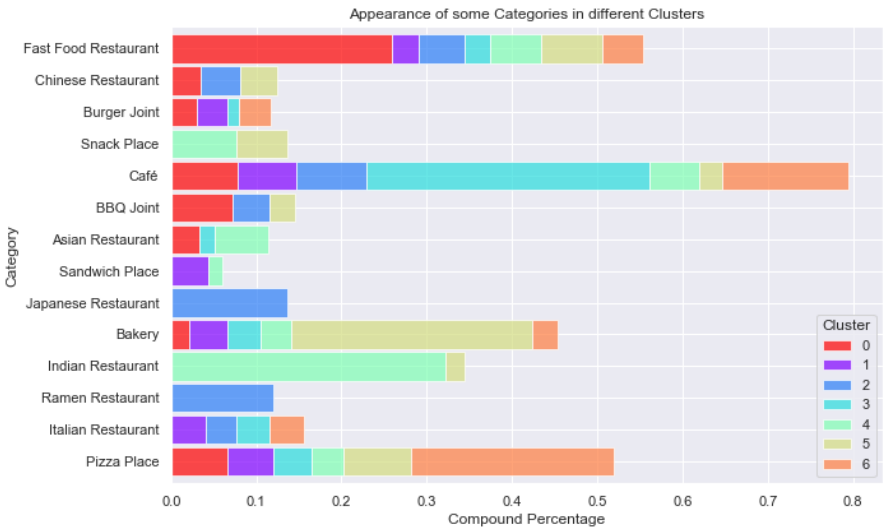

Figure 4: Venue Distribution

What we can see is that the darker blue cluster has rather high values for ramen and Japanese restaurants. Clearly almost all cities in Japan belong to this cluster. Similarly the cities in the "Indian cluster" have Indian restaurants as the dominating food venue.

In the purple cluster 1 where most American and western European cities belong to are more of a mixture of everything without one or two kinds of venues dominating.

The orange cluster seems to be rather Italian with mostly cafes and pizza places and some cities in Italy do indeed fall into this category, but more places in eastern Europe belong to this class. More prominent in Italy is cluster 4 where cafes play the biggest role.

The red "fast food cluster" is mostly found in southern Africa or the Philippenes. Lastly the ocherish cluster (many bakeries) isn't found very often at all. Mostly in India or parts of South America.

5 Discussion

It is interesting to see that even though the clustering algorithm didn't at all take into account where the cities lie, the clusters are still somewhat well separated locationwise as we can see on the world map. Particularly interesting seems to be Japan where all but two cities fall in the same class and no other city outside of Japan falls into the same cluster. This might be because Japan

has a very homogeneous population and does not undergo a lot of foreign influence. So it could be interesting to check the influence that certain demographics have on the results of the clustering.

Given that we have the trained k-means classifier, we can now also use it for predictions. Let's for instance see which cluster is predicted for New York (which is not in the training set). As one might expect the classifier predicts the purple cluster 1. We can check how much New York differs from the average city in the whole cluster (the percentage values are averaged). This can again be nicely shown with a bar chart.

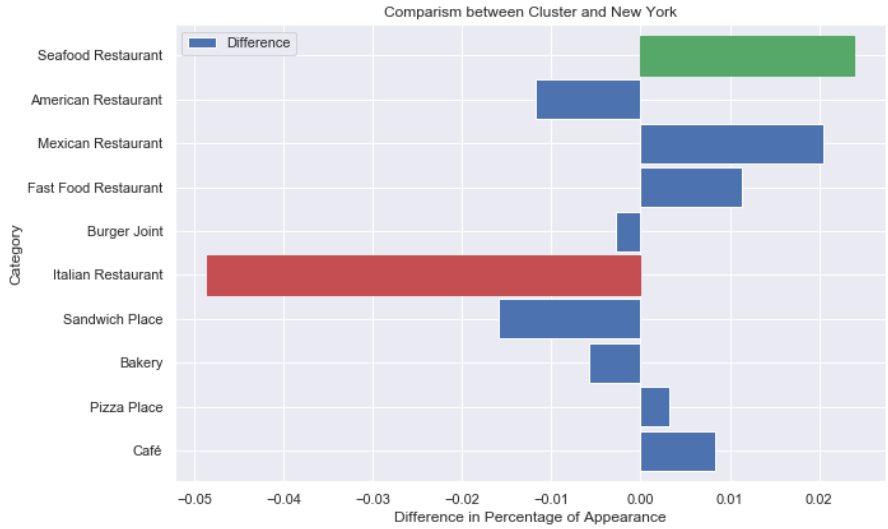

Figure 5: New York Comparison

The red bar shows the most overrepresented food venue in New York and the green bar the most underrepresented one. So if one were to open a food place in New York, it could be recommended to open a Seafood restaurant and most likely not an Italian restaurant.

6 Conclusion

It is nice to see how the clustering algorithm separates the cities into multiple clusters which can be used to give recommendations about which type of food venue might be reasonable choice to open in a certain city. But of course for better recommendations one should use a lot more features like "purchasing power", "revenue of the food venues" etc.