Restaurant Delivery Clusters in Berlin

Ali Gündüz
June, 2020

Introduction

Background

With rapid urbanization on a global scale and an increased labor participation of women has led to a greater demand for delivery of restaurant food services all around the world. Different markets are approaching this challenge in different ways. Whereas restaurants in some countries such as Turkey, due to their lower labor costs, mostly offer home delivery via their own delivery personnel, the developed countries have seen the arrival of food delivery companies such as takeaway.com, Uber Eats, GrubHub that offer delivery services between restaurants and consumers. There are also delivery-focused restaurant chains like Domino's that offer their own delivery service even in developed countries.

Business Problem

The COVID-19 shutdowns have revealed the problematic side of restaurants' dependence on food delivery apps. They can charge fees that seem exorbitant to restaurant owners. Depending on an large corporation also means that the restaurant doesn't have control over a significant part of the food ordering experience of the customer such as delivery speed and service quality.

On the other hand, employing their own delivery personnel is not economically feasible for most independent restaurants in the developed countries where labor costs are considerable.

One possible solution we would like to explore is to have local (think neighborhood or district) food delivery contractors or cooperatives where the end consumers order directly from the restaurant and the restaurant uses a local network of riders for quick delivery. To try out such a delivery service, we would need a location with a dense concentration of restaurants which we could approach for cooperation. Failing to find such a suitable location would mean that our riders wouldn't have enough orders to service throughout the day or that it would take a long time for our riders to deliver the orders. Once we find a suitable cluster of restaurants, we would ideally have a small servicing/waiting station for riders in the cluster's middle.

In our example, we will look for such a location candidate around the city center of Berlin, Germany.

We believe such an experiment would be interesting for independent restauranteurs, city planners and entrepreneurs as possible stakeholders.

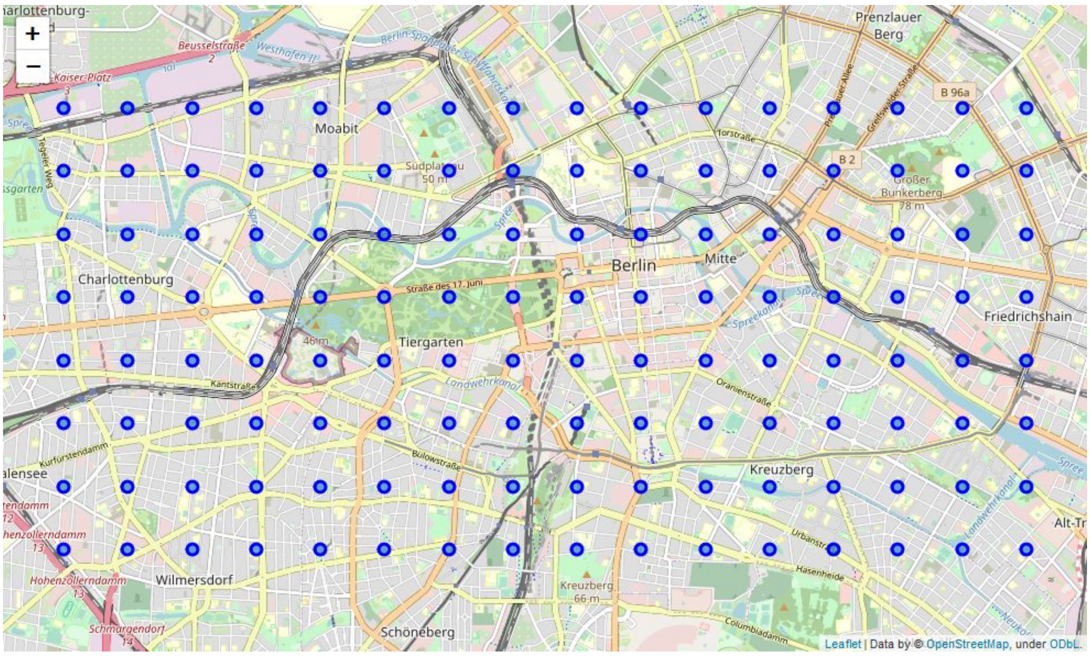
Data

We will use the **Foursquare API** to get local restaurant information in Berlin. In particular, Foursquare API's `explore` functionality returns us a venue list for a given coordinate and radius. We will download venue data for a large section of Berlin's city center and save the **name**, **category**, **id number** and **coordinates** (latitude and longitude) for each restaurant.

We will also use the **OpenStreetMap Nominatim** geolocation service via the geopy library to reverse search street addresses for the cluster center coordinates.

Methodology

When exploring recommended venues near a given location, Foursquare API limits the number of results. Since we would like to explore a large part of Berlin's city center, simply exploring the city from one location will not give us enough data points. As a solution we can create a grid of locations and explore each location separately to get a comprehensive view of the city. By setting the south-western and north-western edges of the grid and the stride distances between each point we want, we set up a grid of 128 location points over the central Berlin. We can use `Folium` to visualize a map to make sure we are covering the right area.



We will use Foursquare API's `explore` endpoint on the location coordinates to list most popular `food` venues within 500 metres of each point. Beware that Foursquare's results change depending on the time of the day we run the query. So, we use the `time` parameter so that the results correspond to any time of the day.

As we proceed with our exploration of each location point in our grid, we will receive venue name, category, id number and coordinates.

```
Exploring: (52.526, 13.44)
Exploring: (52.526, 13.45)
Exploring: (52.532, 13.3)
Exploring: (52.532, 13.31)
Exploring: (52.532, 13.32)
Exploring: (52.532, 13.33)
Exploring: (52.532, 13.34)
Exploring: (52.532, 13.35)
Exploring: (52.532, 13.36)
Exploring: (52.532, 13.37)
Exploring: (52.532, 13.38)
Exploring: (52.532, 13.39)
Exploring: (52.532, 13.4)
Exploring: (52.532, 13.41)
Exploring: (52.532, 13.42)
Exploring: (52.532, 13.43)
Exploring: (52.532, 13.44)
Exploring: (52.532, 13.45)
Finished exploring!
```

Note that we have a slight overlap between coverage areas of grid points to make sure that we don't have blind spots inside the grid areas. Therefore, for each venue data we receive we check if the venue has been saved before. We save all new venues in a common `pandas` dataframe.

```
all_areas.head(10)
```

	id	name	categories	lat	lng
0	51b0b687498eb7911a3f5d5d	Viet's	Vietnamese Restaurant	52.486826	13.298148
1	50083606e4b0a467c6a67a52	Il Gusto	Italian Restaurant	52.487640	13.299763
2	515ad070e4b0e1f513fdb28b	Steinecke	Bakery	52.493167	13.305096
3	4b5c51f4f964a520dd2a29e3	Trattoria Taormina	Trattoria/Osteria	52.491707	13.301395
4	4dd3a22cb0fbf653b6413b4c	Grillhaus Wedo's	Snack Place	52.488470	13.301182
5	4e2680c93151a5765fc70e27	Wasserperle	German Restaurant	52.491629	13.299931
6	51727692e4b093e579db5b55	Das Knusperhäuschen	Bakery	52.491535	13.293843
7	4f9fadd5e4b0f9485431228c	LECKERBACK	Bakery	52.486170	13.297124
8	56670809498eba82ce7def90	Genazvale	Caucasian Restaurant	52.489803	13.312004
9	4b0919a7f964a5203d1423e3	Parkcafé Berlin	Café	52.490966	13.314369

```
all_areas.shape
```

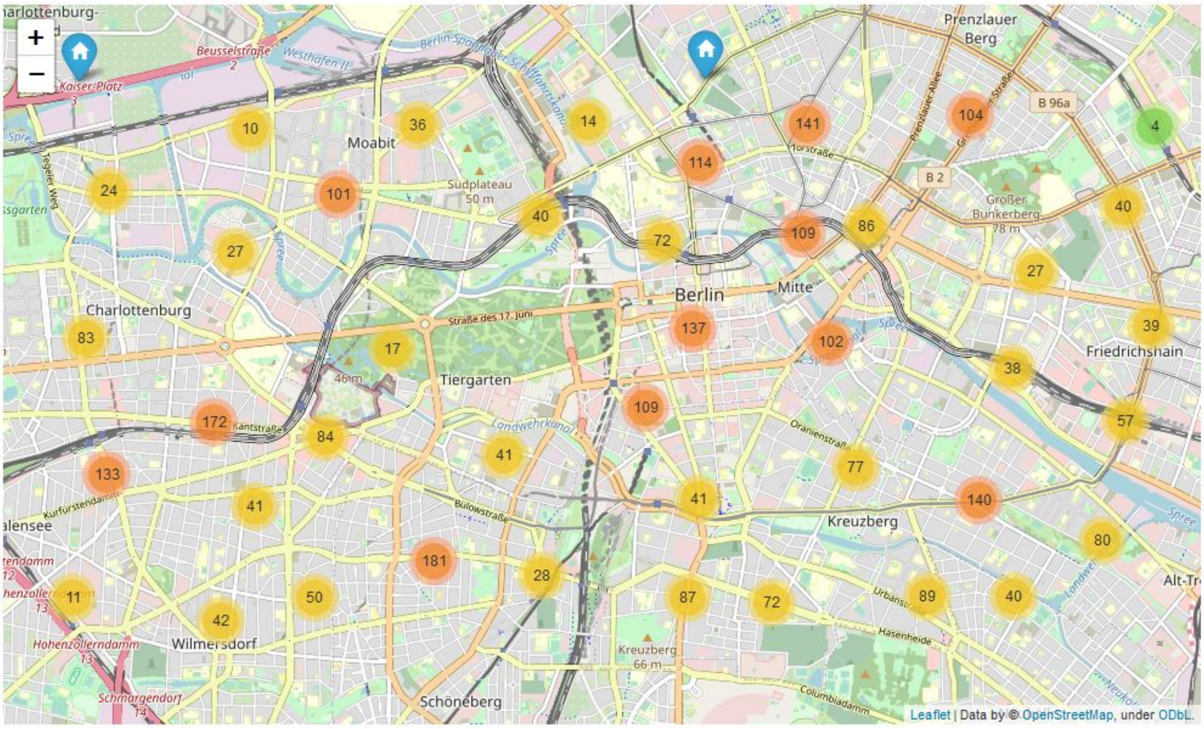
```
(2942, 5)
```

```
print('There are {} unique restaurant categories.'.format(len(all_areas['categories'].unique())))
```

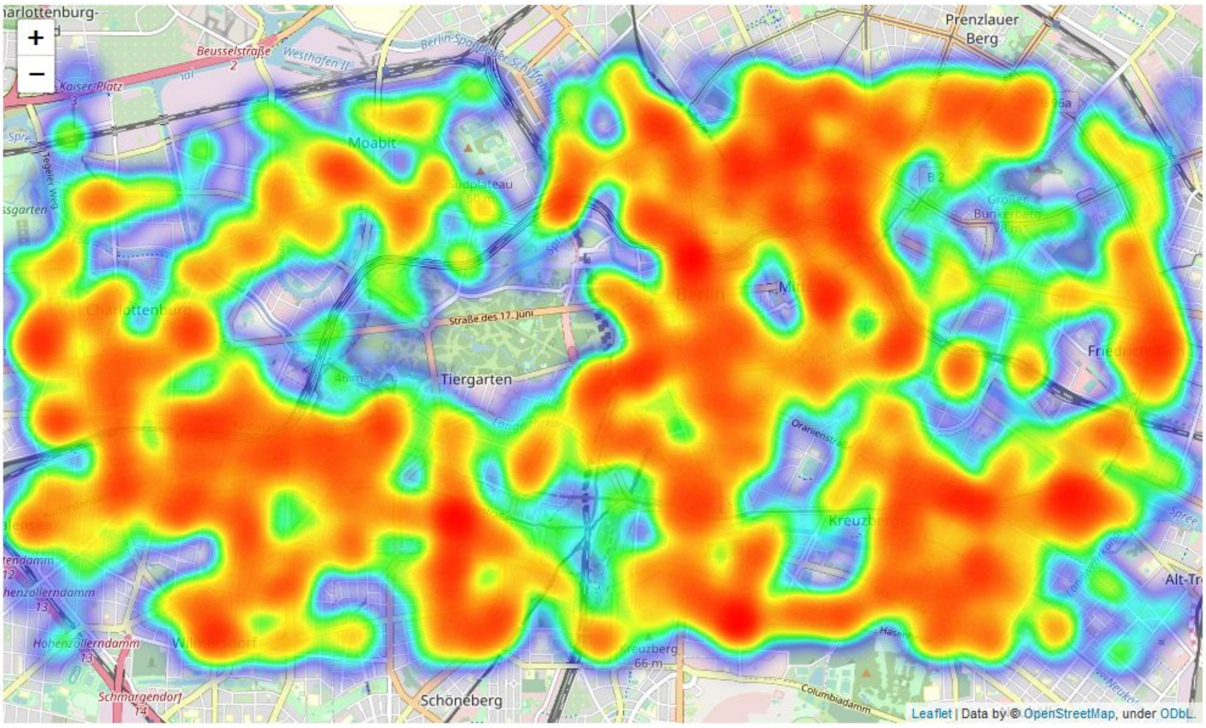
```
There are 120 unique restaurant categories.
```

As a result of this grid search, we collect almost 3000 restaurants in 120 categories in our dataframe.

Before running machine learning to get viable restaurant clusters, we can visualize the restaurant locations on a map to get a sense of the results. Since there are too many locations to make a usual map with individual markers for each location, we utilize `FastMarkerCluster` which interactively clusters markers depending on the zoom level.



As an additional map type, it would be interesting to utilize a `HeatMap`, as this map type would give us a visual way to imagine the density clusters of restaurants.



From this map, we can already see that restaurant concentration is low in parks and high in places near large train stations and shopping districts.

Now that we have an idea of restaurant locations in the city center, we need to determine if and where there are dense clusters of venues. Our goal is to determine candidates for an area with an enough density of restaurants where we can offer a network of locally available riders as a home delivery solution for restaurants.

For this task, we don't choose traditional clustering methods such as k-means, hierarchical and fuzzy clustering. Even though they are good for determining areas with enough number of venues without supervision, they are not able to separate high density clusters from low-density areas.

Instead, we will use **DBSCAN** (Density-based spatial clustering of applications with noise) algorithm which can give us regions of high location density while separating low density locations as noise without supervision. We define density as having at least 60 restaurants within our specified radius.

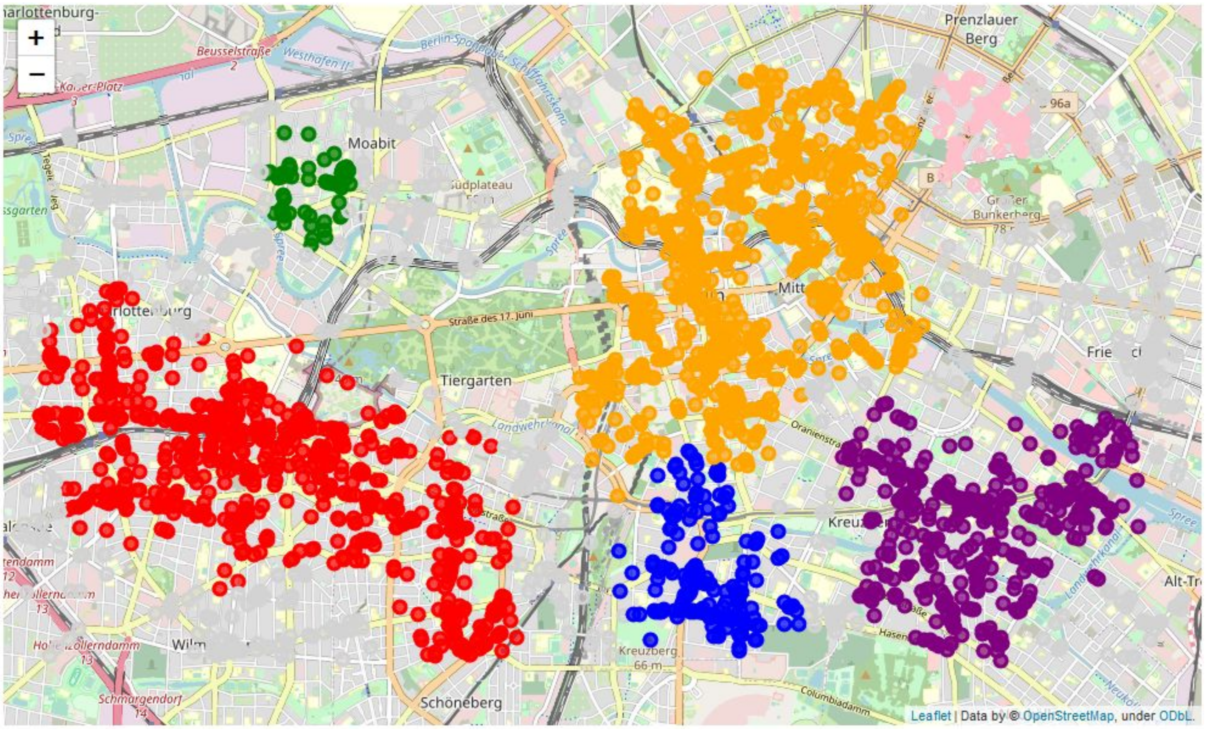
```
db = DBSCAN(eps=0.006, min_samples=60).fit(all_areas[['lat','lng']])
labels = db.labels_
unique_labels = set(labels)

print(unique_labels)
```

```
{0, 1, 2, 3, 4, 5, -1}
```

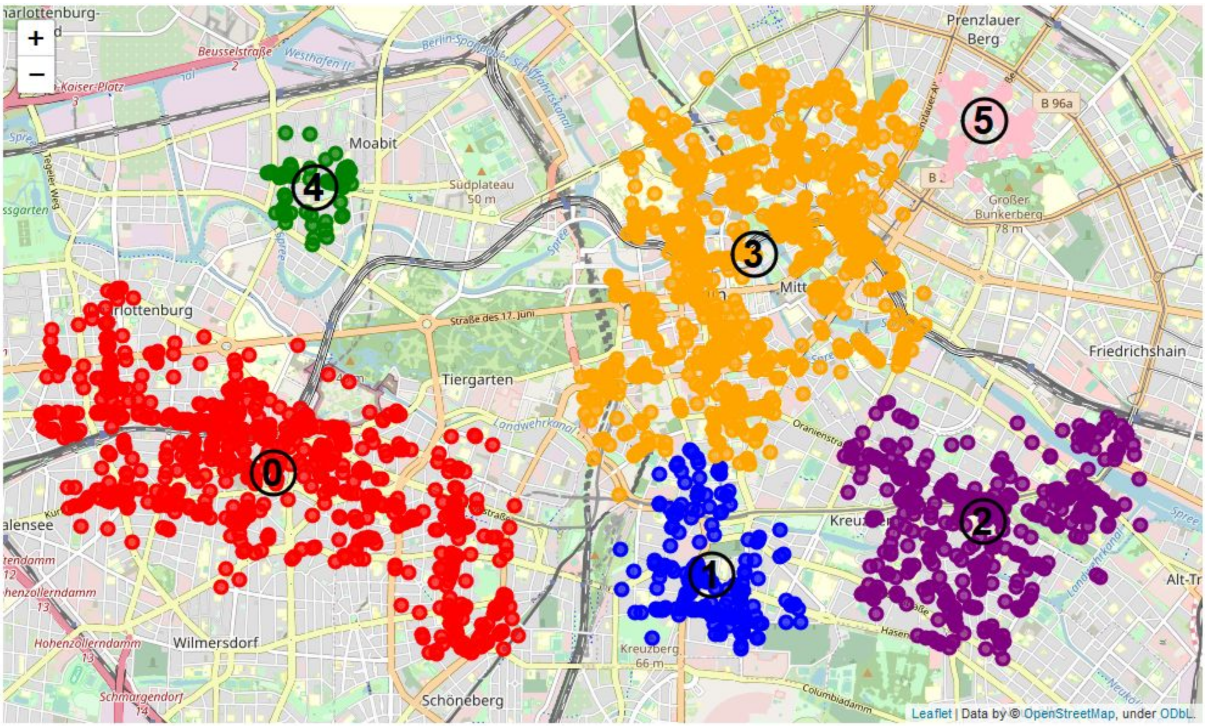
DBSCAN found **6 clusters** (Regions 0, 1, 2, 3, 4, 5) that fit our requirements. Please note that -1 label is used for locations deemed noise, i.e. members of non-dense regions. We add the region labels to corresponding restaurant rows in our dataframe

	id	name	categories	lat	lng	Region
0	51b0b687498eb7911a3f5d5d	Viet's	Vietnamese Restaurant	52.486826	13.298148	-1
1	50083606e4b0a467c6a67a52	Il Gusto	Italian Restaurant	52.487640	13.299763	-1
2	515ad070e4b0e1f513fdb28b	Steinecke	Bakery	52.493167	13.305096	-1
3	4b5c51f4f964a520dd2a29e3	Trattoria Taormina	Trattoria/Osteria	52.491707	13.301395	-1
4	4dd3a22cb0fbf653b6413b4c	Grillhaus Wedo's	Snack Place	52.488470	13.301182	-1
5	4e2680c93151a5765fc70e27	Wasserperle	German Restaurant	52.491629	13.299931	-1
6	51727692e4b093e579db5b55	Das Knusperhäuschen	Bakery	52.491535	13.293843	-1
7	4f9fadd5e4b0f9485431228c	LECKERBACK	Bakery	52.486170	13.297124	-1
8	56670809498eba82ce7def90	Genazvale	Caucasian Restaurant	52.489803	13.312004	-1
9	4b0919a7f964a5203d1423e3	Parkcafé Berlin	Café	52.490966	13.314369	-1
10	500e7a42e4b0a57241d8b10f	Truc-xinh	Thai Restaurant	52.490150	13.314045	-1
11	507feb4fe4b015eaecf999ea	Schlemmermarkt Fehrbeliner Platz	Food Court	52.489933	13.314410	-1
12	519e2e97498e323cc612c0ee	Café Milch Zeit	Café	52.490984	13.316710	-1
13	58a04ec901f0772d803c209d	Bahadur	Indian Restaurant	52.487791	13.319595	-1
14	4bddb02b587b2d7f40ab5409	A Telha	Seafood Restaurant	52.490520	13.325013	-1
15	4c66e091e75ac92853cff8da	Paracas	Latin American Restaurant	52.491891	13.321121	0
16	4e628839d164ddd5e5d21463	Restaurant Zum Hax'nwirt	German Restaurant	52.491524	13.317790	-1
17	4b609427f964a5208eee29e3	Witwe Bolte	German Restaurant	52.492500	13.323716	0
18	4db1891d6e81a2637eed52fd	Les 3 veuves de Wilmersdorf	French Restaurant	52.489290	13.319580	-1

At this point, it makes sense to visualize a map of our clusters (color markers) as well the noise (gray markers).



As expected, DBSCAN separated out locations in areas it deemed not densely populated enough with other restaurants. To visualize the clusters more clearly, we remove noise and label the region numbers on the map.



Taking the average of coordinates in each cluster, we can calculate clusters' center points. Moreover, via the reverse search functionality of the geopy library, we can acquire the street addresses of these center points.

```
Center of Region 0 : 6, Meinekestraße, Charlottenburg, Charlottenburg-Wilmersdorf, 10719, Germany
Center of Region 1 : 107, Gneisenaustraße, Kreuzberg, Friedrichshain-Kreuzberg, Berlin, 10961, Germany
Center of Region 2 : 9, Lausitzer Straße, Kreuzberg, Friedrichshain-Kreuzberg, Berlin, 10999, Germany
Center of Region 3 : Pergamon Museum, 5, Am Kupfergraben, Spandauer Vorstadt, Mitte, Berlin, 10117, Germany
Center of Region 4 : Volkshochschule Mitte, Haus 3, 75, Turmstraße, Moabit, Mitte, Berlin, 10551, Germany
Center of Region 5 : Borowsky - der Ankleider, 27, Greifswalder Straße, Winsviertel, Prenzlauer Berg, Pankow, Berlin, 1040
5, Germany
```

As mentioned in the introduction, a local delivery service may consider having a servicing/waiting station for its delivery riders that is central to its service area. In such a case, the areas around these addresses would be ideal locations.

Results

Having obtained possible candidates for a local delivery service, we can now tackle the question of which of our resulting clusters would make for the best trial area for a test run.

Obviously, having as many restaurants as possible nearby to approach for cooperation would increase our chances of success. As already can be gleaned from the above map, not all clusters have the same number of restaurants in them.

Region	
0	690
1	182
2	407
3	875
4	62
5	72

From these numbers, **Regions #3** (875 restaurants) and **#0** (690 restaurants) are the most populous clusters by a long lead to other regions.

Before deciding on simply the most populous region as our best candidate, it could be interesting to compare the restaurant categories presented in the clusters, in case there is a significant discrepancy that could affect our decision.

Using One Hot Encoding, we transform category data into 0's and 1's for each category's presence in each region. This lets us form the frequency of occurrence of each category in our regions, from which we can list the top 10 most common restaurant categories in each region.

Region	1st Most Common Category	2nd Most Common Category	3rd Most Common Category	4th Most Common Category	5th Most Common Category	6th Most Common Category	7th Most Common Category	8th Most Common Category	9th Most Common Category	10th Most Common Category
0	Italian Restaurant	Café	Bakery	German Restaurant	Vietnamese Restaurant	Restaurant	Burger Joint	Chinese Restaurant	Indian Restaurant	French Restaurant
1	Café	Italian Restaurant	Bakery	Asian Restaurant	Vietnamese Restaurant	Indian Restaurant	Pizza Place	Bistro	Sushi Restaurant	German Restaurant
2	Café	Italian Restaurant	Bakery	Pizza Place	German Restaurant	Vietnamese Restaurant	Restaurant	Turkish Restaurant	Indian Restaurant	Falafel Restaurant
3	Café	Italian Restaurant	German Restaurant	Bakery	Vietnamese Restaurant	Restaurant	Sushi Restaurant	Breakfast Spot	Bistro	Vegetarian / Vegan Restaurant
4	Bakery	Café	Italian Restaurant	Pizza Place	Asian Restaurant	Burger Joint	Breakfast Spot	Vietnamese Restaurant	Fried Chicken Joint	Indian Restaurant
5	Café	Vietnamese Restaurant	Italian Restaurant	Bakery	Breakfast Spot	Deli / Bodega	German Restaurant	Pizza Place	Sushi Restaurant	Doner Restaurant

There seems to be a similarity of category distributions. To get an even more stark, we focus on the top 6 categories in our two most populous regions.

Region	1st Most Common Category	2nd Most Common Category	3rd Most Common Category	4th Most Common Category	5th Most Common Category	6th Most Common Category
0	Italian Restaurant	Café	Bakery	German Restaurant	Vietnamese Restaurant	Restaurant
3	Café	Italian Restaurant	German Restaurant	Bakery	Vietnamese Restaurant	Restaurant

We again see very similar representations of categories in these two regions. In fact, with a slight difference of order, the top 6 restaurant categories are exactly the same!

Before coming to a final decision, we can note that the discrepancies in less popular restaurant categories are also low. By finding the standard deviations of frequency of occurrence for each category. We calculate the mean standard deviation between categories in regions #0 and #3 to be around 0.0028

With a low mean of standard deviation between occurrences of categories between the two regions, we conclude that two candidate regions have a similar distribution of restaurant types.

Discussion

As we have seen, Region #3 offers us a high density restaurant cluster. There is also not a very significant discrepancy of available restaurant categories in this region in comparison to other populous regions.

With these evidence, we can conclude that our best candidate area for a local food distribution service in Berlin would be Region #3, which spans the popular “Mitte” (literally middle) district of Berlin.

In the future, it could be useful to explore specific categories' popularity for food delivery and adjust results accordingly. Also, a survey of restaurant owners in our target area could enlighten us further as to which factors we could focus our attention.

Conclusion

We believe that densely populated urban areas could offer alternative ways for food delivery techniques where independent restaurants can offer quick and high quality delivery services without being subordinate to delivery conglomerates which may not have the restaurants' best interests in mind.

Whether it be a local contractor company, a cooperative between restaurants themselves or any other delivery scenario not discussed, analysis of restaurant data will be crucial in determining the success of any effort for a new food delivery network.