Up and coming: venue location scouting in the neighbourhoods of Berlin

Frederik Laubisch

November 2019

1 Introduction

1.1 Background

Berlin is the capital and the biggest city of Germany. It consists of 96 neighbourhoods, many of which are well known to both locals and tourists alike to have a plethora of interesting venues such as food places and entertainment. In the past many neighbourhoods hat were considered to be working-class and low-income areas went trough a process of gentrification and were turned into more fancy and high-class neighbourhoods. Today many of these areas are currently in the very middle of this kind of transformation.
In the most high-end and well-known areas of Berlin a saturation of venues has been reached, and starting a new business and surviving can be hard because of high rents and plenty of established competition. Therefore, it would be beneficial to find areas that are less sated with venues, but still have similar enough characteristics so that the right target audience is reached. Setting up a business in a low-income area that targets citizens of the higher-income areas would surely fail.

1.2 Problem

Based on data about the number and types of venues, as well as population and area size of the neighbourhood will be used to find areas that fall into this gap between high-end and low-end.

1.3 Interests

Owners of businesses that which to expand as well as those who wish to open their own, new business are both likely to benefit from this kind of analysis.

2 Data

2.1 Data Sources

First data about Berlins neighbourhoods needs to be acquired. This data is readily available on this wikipedia page. The table in question lists each neighbourhood with population, area size and population density. The python library *BeutifulSoup* will be used for this purpose.
Information about venues will be acquired using the FourSquare Developer API. For this purpose the coordinates for each neighbourhood will be determined using the *geopy* library for python. In order to get a rough estimate of how many venues are actually within the borders of the area, we are going to calculate an individual radius for each neighbourhood based on its area, calculated as $r = \sqrt{\frac{a}{\pi}}$, with a being the area and r the radius. This provides a rough estimate since the neighbourhoods are not circle shape.

2.2 Feature Extraction

We are going to use the relative frequency of venue types in each neighbourhood. We are only interested in venues that are businesses that provide some kind of service. This notably excludes public buildings and attractions such as parks, rivers and monuments. We want to capture the makeup of each of the neighbourhoods in terms of service businesses.
Additionally, the FourSqaure API returns venues such as railway and bus stations, as well as governmental and public service buildings such as post offices and administrative buildings.
One of the problems of the FourSquare API is that it returns a maximum of 100 venues per address only. This means we get 100 venues, then remove a number of them as explained above. The result is that neighbourhoods with more than 100 venues will have fewer than 100 venues after removing the one we are not interested in. However, there might be more venues that are of interest in the neighbourhood, that the API simply did not return because of the limit. If there was a way to filter the API call beforehand this could be fixed. But since there is no obvius way, we just have to keep this in mind for later.

2.3 Data Cleaning

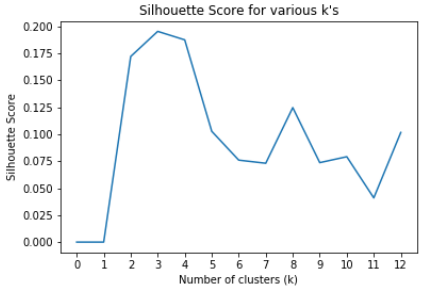
After extracting the venues which are not of the excluded categories, some data cleaning is necessary, since there are many venue categories left, that are generally of no use. Examples include categories such as 'Tree', 'Building', 'Forest' or 'Boat'.
Additionally the population for one of the neighbourhoods (Malchow) was wrong in the DataFrame. This was because the population was written with a comma as a thousands separator. BeautifulSoup interpreted this as a decimal separator and thus listed all population numbers as thousands. One neighbourhood

had fewer than a thousand and thus no thousand separator and was therefore interpreted as having 1000 time the number of inhabitants. This was easily fixed.

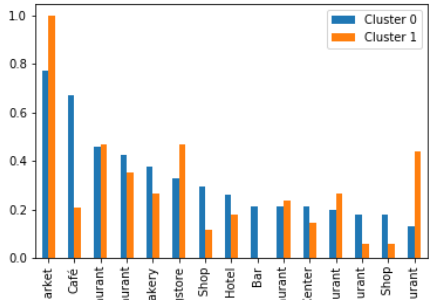
3 Methodology

3.1 Explorative Data Analysis

Using the prepared data, the neighbourhoods were clustered using the k-means algorithm with the relative frequency of the venue types as features.
The optimal number of k was determined by iterating over multiple k*'s* and evaluating them using the *Silhouette Score*. The resulting plot is shown below.

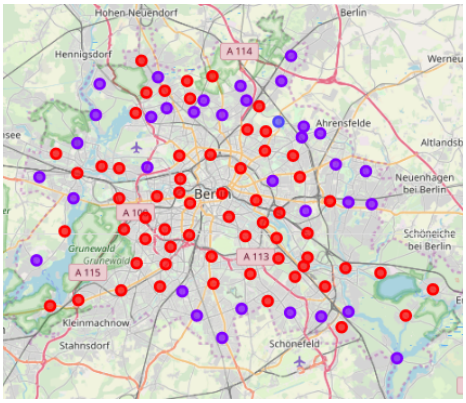


The optimal value $k = 3$ was then used to cluster the neighbourhoods. Of the resulting three clusters one consisted only of the small neighbourhood of Malchow. This small outskirt settlement is only nominally part of Berlin and it stands to reason that it would be clustered away from the rest.
The other two clusters were of more interest. Using the top 10 venue categories of each neighbourhood, we compare them by calculating the relative frequency of the venue types in the top 10. The result is shown in the figure below.



We can see that the neighbourhoods in cluster 1 are dominated by supermarkets,

drug-stores and fast-food restaurants. This suggest that they are of the low-end variety, and not the ones we are interested.
Cluster 0 consists of neighbourhoods with a higher relative number of cafe's, bars and coffee shops. So these are the ones we care about.
We can also create a map of the clusters using the *Folium* library. The results are in the figure below. We can observe that the neighbourhoods in cluster 0 tend to be the ones in the city center, while the ones in cluster 1 are more on the outskirts.



With the cluster of interesting neighbourhoods identified, we can now try to answer our initial problem. For this purpose we calculate the venue density of each neighbourhood. The venue density in this context is the number of venues per 1000 inhabitants. The results and insights of this will be reported in the next section.

4 Results

In order to answer the initial problem, we look at the neighbourhoods in the cluster of interest and order them by venue density. The to of the resulting table is shown in the next figure.

Here it is important to come back to the issue mentioned in the feature engineering part. The neighbourhoods with venue numbers close to 100 all had more than 100 venues originally. Due to the limitations of the FourSquare API and the feature selection process, they are now listed to have fewer than 100 venues, even though the number is likely much higher in reality. Many of them have over 100k inhabitants, and thus the venue density would be much higher if the correct number f venues was available.
Another issue that is revealed at this point, is that there are some neighbourhoods with very few venues. Notably the neighbourhoods *Falkenhagener Feld* and *Blankenburg* only have 8 and four venues, respectively. One of them is also

	Neighbourhood	Population	N_venues	Venue Density
35	Falkenhagener Feld	38.592	8	0.207297
85	Reinickendorf	83.447	25	0.299591
94	Märkisches Viertel	40.258	23	0.571315
10	Blankenburg	6.865	4	0.582666
8	Prenzlauer Berg	164.593	96	0.583257
50	Neukölln	166.126	98	0.589914
7	Kreuzberg	154.862	96	0.619907
54	Gropiusstadt	37.450	24	0.640854
71	Biesdorf	27.723	18	0.649280
6	Friedrichshain	134.900	99	0.733877
44	Schöneberg	123.680	93	0.751940
21	Charlottenburg	130.223	98	0.752555
0	Mitte	101.932	88	0.863321
77	Lichtenberg	41.112	36	0.875657
47	Mariendorf	52.954	48	0.906447
82	Alt-Hohenschönhausen	48.728	45	0.923494

very small, with only 6000 inhabitants. The other may have likely had more venues, but a lot of them may have been excluded during feature selection. These kind of outliers should in retrospect probably been excluded from the data set.
The implications of the results will be discussed in the next section.

5 Discussion

We managed to identify the group of neighbourhoods that are of interest to us, and found areas with lower venue densities. Lets take out all neighbourhoods with more than 80 venues, since they probably had more than 100 before feature selection. The table is shown below, with only the neighbourhoods with a density of below 1 venue per 1000 inhabitants, as well as excluding the outliers discussed in the last section.

Based on this we can make some recommendations. The neighbourhoods in the table belong to the same cluster as the famous upscale neighbourhoods with many cafes and bars that are attractive to tourists and locals alike. Additionally they have a relatively small number of relevant venues per 1000 inhabitants. This means they are less saturated with said venues. Our recommendation is to open up new businesses like bars, cafes and artisinal restaurants in these neighbourhoods.
A next good step would be to get historical data about venue count, density

	Neighbourhood	Population	N_venues	Venue Density
85	Reinickendorf	83.447	25	0.299591
94	Märkisches Viertel	40.258	23	0.571315
54	Gropiusstadt	37.450	24	0.640854
71	Biesdorf	27.723	18	0.649280
77	Lichtenberg	41.112	36	0.875657
47	Mariendorf	52.954	48	0.906447
82	Alt-Hohenschönhausen	48.728	45	0.923494
9	Weißensee	53.737	50	0.930458
76	Karlshorst	27.522	27	0.981033
51	Britz	42.640	42	0.984991

as well as population in order to create a predictive model. This way we could distinguish between neighbourhoods that are up-and-coming, those that are likely to be stable, and those that are actually declining. Unfortunately this kind of data is not available trough teh FourSquare API.

6 Conclusion

We looked at the neighbourhoods of Berlin and managed to group them based on the makeup of service business venues. We identified the group of neighbourhoods that contains the well known and successful neighbourhoods, but also smaller, up-and -coming neighbourhoods. Based on density of venues of interest we identified neighbourhoods that still have room for more service businesses. In conjunction with domain knowledge about Berlin we can provide a list of neighbourhoods that are likely candidates for more venues and have the potential to be the next big thing.