From Berlin to Munich

Robert Schewski

April 2021

Contents

1 Introduction

Imagine you are living in Berlin and now you have to change the job. Let the best job you can find be in Munich. So the question is, can one find at least some neighborhood which is comparable to my actual one. In the following this answer will be tried to be solved by analyzing the local structure of the city, by means of venues in a neighborhood, extracted by the Foursquare API. From this, similarities in the neighborhood in Berlin and Munich will be analyzed. This will be done by extracting the structures of the cities from public accessible sources and and than a comparison of the different features (namely the venues) of the cities. By clustering of similar neighborhood possible recommendations can be given, for neighborhood's, which are in some respect similar in Berlin and Munich.

2 Methods

Here the main methods will me described.

2.1 Python Packages

1. **Pandas:** Pandas is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. In this work Pandas is used through the whole analysis process as main data structure. More on Pandas can be found at: Pandas
2. **Requests:** Requests a Python HTTP library, enabling HTTP requests on a simpler and more human-friendly way. In this work Request is used to access HTTP data from the web. More on Request can be found at: Requests
3. **BeatifulSoup4:** Beautiful Soup is a library that makes it easy to scrape information from web pages. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree. More on BeatifulSoup can be found at: BeatifulSoup4
4. **docx:** python-docx is a Python library for creating and updating Microsoft Word (.docx) files More on docx can be found at: hrefhttps://python-docx.readthedocs.io/en/latest/docx
5. **Requests:** This module provides a portable way of using operating system dependent functionality. More on os can be found at: Requests
6. **Geocoder:** Geocoder is a simple and consistent geocoding library written in Python. In the frame of this work it is used to get Geo Coordinates for a given location based on the name. There are a huge amount of different providers for this information like Google, Bing, OSM. In this work the Geo-information of ARCGis are used. More on geocoder can be found at: Geocoder
7. **Foursquares :** Foursquares is a customer based app for Geo-information based on recommendations of local venues. It provides an API for developers to get local information based on there venues database. In this work it is used to get local venues based on neighborhoods in a city to compare this to other neighborhoods in other cities. More on Foursquare API can be found at: Foursquares
8. **Folium** Folium is a geomapping framework to map data on interactive maps. it builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the leaflet.js library. More on folium can be found at: Folium
9. **Sklear** Sklear is a free software libary for machine learning in Python. It provides different classifications, regression and clustering algorithm. More on sklearn can be found at: Sklear
10. **Matplotlib:** Matplotlib is a plotting library in Python. It is used to plot different data. More on Matplotlib can be found at: Matplotlib
11. **GeoPandas:** GeoPandas is an open source project for working with geospatial data in python. It extends the data types used by pandas to allow spatial operations on geometric types. Here it is used for plotting the neighborhoods of Berlin. More on Geopandas can be found at: GeoPandas

2.2 Machine Learning Algorithm

1. **sklearn kmeans:** The k-means Algorithm is a method for vector quantisation, also used for cluster analysis. A given Quantity of similar objects is divided into a given amount of groups k by similarity. Mathematically this means that the distance of the sum of the squared displacement from the cluster centroid μ_i has to minimized. This means the optimization of the function:

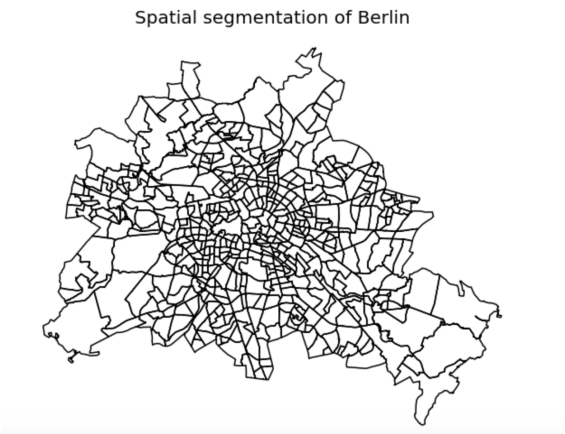
$$J = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \mu_i\|^2$$

   where x_j denotes the data points and μ_i the centroid of the cluster S_i. In case of the sklearn implementation the k-means problem is solved using either Lloyd's or Elkan's algorithm.

2. **scipy.cluster.hierarchy.linkage:** The linkage methods are used to compute the distance $d(s, t)$ between two clusters s and t. The algorithm begins with a forest of clusters. When two clusters s and t from this forest are combined into a single cluster u, s and t are removed from the forest, and u is added to the forest. When only one cluster remains in the forest, the algorithm stops, and this cluster becomes the root. At each iteration a distance matrix d is maintained, with $d(i, j)$ corresponding to the distance between cluster i and j. In this study the distance is derived by the uses the Ward variance minimization algorithm.

3. **scipy.cluster.hierarchy.dendrogramm:** The dendrogram illustrates how each cluster is composed by drawing a U-shaped link between a non-singleton cluster and its children. The top of the U-link indicates a cluster merge. The two legs of the U-link indicate which clusters were merged. The length of the two legs of the U-link represents the distance between the child clusters. It is also the distance between original observations in the two children clusters.

(a) Borough of Berlin



(b) Lokal segmentation as LOR (see text)

Figure 1: Segmentation of Berlin

3 Data

Here a short description of the available data will be presented.

3.1 Berlin

Lebensweltlich orientierte Räume (LOR) (living environment oriented spaces)
The city of Berlin is divided into of 12 boroughs (see fig.: 1a) Since 2006, Berlin is officially divided into "living environment oriented spaces" (LOR). The segmentation of Berlin into so called LOR's is shown in fig.1b These are used as "spatial basis for planning, forecasting and monitoring demographic and social developments in Berlin". Due to the dynamic (population) development of Berlin an adjustment of the LOR was necessary. With the beginning of 2021 a new definition of the LOR was made and can be downloaded from the above shown link. The data can downloaded as a table embedded into a Word docx file. The downloaded .docx file can also be found in the GitHub repository of this work. Following the above link one can obtain also obtain shapefiles of the segmentation of Berlin.

3.1.1 Data cleaning

The data for the spatial segmentation of Berlin can be found in form of a table embedded in a Microsoft Word file. From this file the data was extracted by the use of the docx package.

	PGR \nID	PGR Name	BZR ID	BZR Name	PLR ID	PLR Name	EW 31.12.19
0	01 Mitte	01 Mitte	01 Mitte	01 Mitte	01 Mitte	01 Mitte	01 Mitte
1	0110	Zentrum	011001	Tiergarten Süd	01100101	Stülerstraße	3.445
2	0110	Zentrum	011001	Tiergarten Süd	01100102	Großer Tiergarten	1.755
3	0110	Zentrum	011001	Tiergarten Süd	01100103	Lützowstraße	5.159
4	0110	Zentrum	011001	Tiergarten Süd	01100104	Körnerstraße	4.626

Figure 2: extracted data frame from the word document

The resulting data frame (see figure 2) was consisting of rows containing sub-headers which needed to be removed. A borough column was added and the columns *BZR ID* , *PGR nID*, *PLR ID* and *EW 31.12.19* were removed. The table headers where acronyms used by government of Berlin and have been translated to English. The resulting table is shown in fig 3

	Region	Borough	Sub-Borough	Neighborhood
1	Zentrum	Mitte	Tiergarten Süd	Stülerstraße
2	Zentrum	Mitte	Tiergarten Süd	Großer Tiergarten
3	Zentrum	Mitte	Tiergarten Süd	Lützowstraße
4	Zentrum	Mitte	Tiergarten Süd	Körnerstraße
6	Zentrum	Mitte	Regierungsviertel	Wilhelmstraße

Figure 3: final dataframe for Berlin

3.2 Munich

Figure 4: Segmentation of Munich

Stadtbezirke München

The municipal districts of Munich represent an administrative division of the urban area of the Bavarian state capital Munich. The data for the individual Neighborhoods can be extracted from a table present on the wikipedia (german) page *Stadtbezirke Münchens*. The page can be obtained by the link shown above.

3.2.1 Data cleaning

The data for the spacial segmentation was obtained from the Wikipedia side. To scrape the table the BeatifulSoup package was used.

4 Results

To analyse the above described data, the two data frames for Munich and Berlin were merged. For the resulting data frame the spacial coordinates for each neighborhood were evaluated using the geocoder function with the flag for the ArcGis database, and added to the data frame. This data Frame was used to obtain nearby venues to all locations in the data frame by using the Foursquares API. As parameters 100

results within a radius 600m for each area was used. For Clustering analysis the categorical values were substituted by the hot encoding function. To find a suitable amount of cluster by the kmeans algorithm a dendrogramm was constructed. The result is shown in fig 5. From this a choice of 6 cluster was made since this seems to reproduce the features of the cluster best.

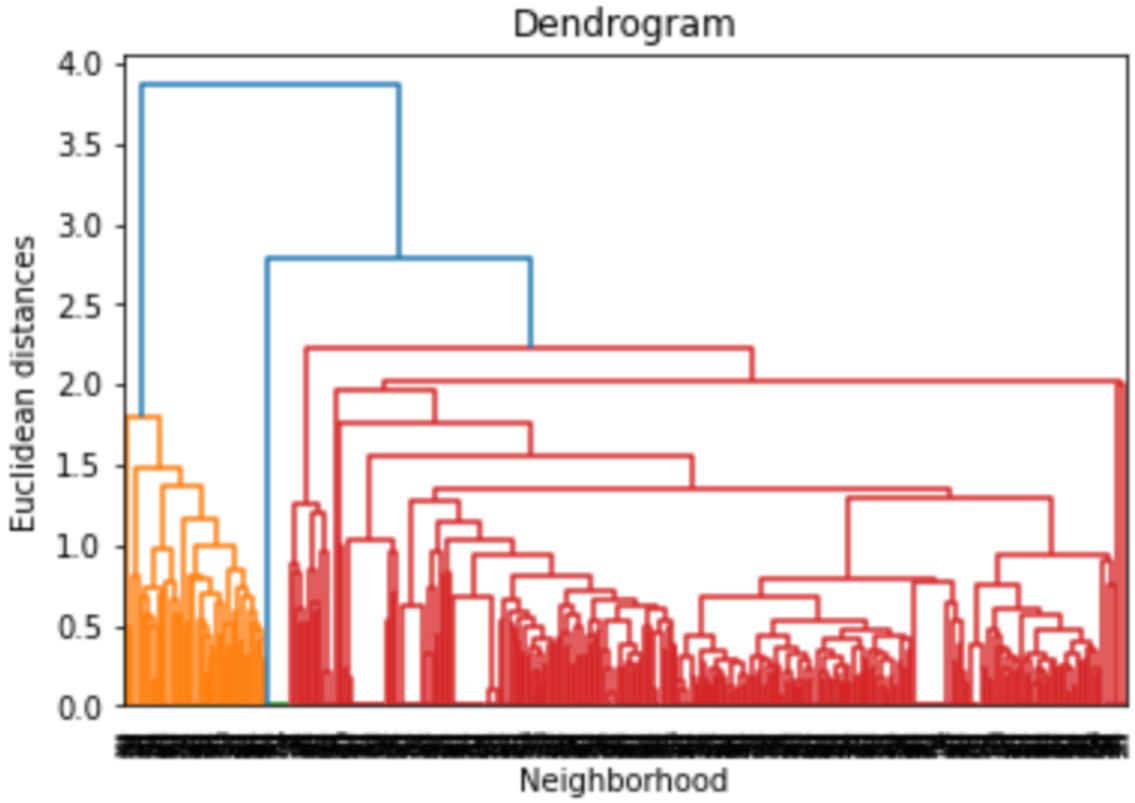


Figure 5: Dendrogram showing the hierarchical clustering of the neighborhoods of Berlin and Munich.

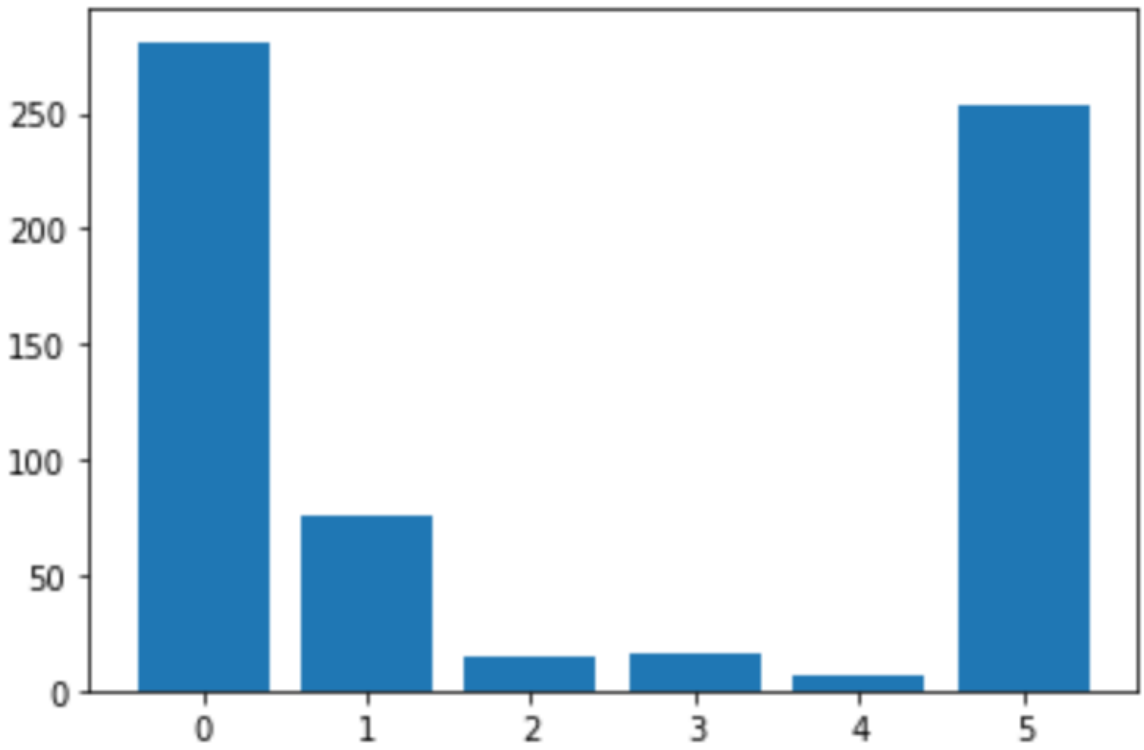


Figure 6: Histogram showing the distribution of different neighborhoods to the different cluster, as evaluated by kmeans algorithm

In the next step the data was clustered into 6 clusters by the kmeans algorithm. The amount of group member in the clusters can be plotted as an histogram to show the distribution and indicate the main cluster. The histogram is shown in fig 6. Cluster 0 has the most group member followed by the cluster 5. The rest of the cluster has a much smaller amount of group members.

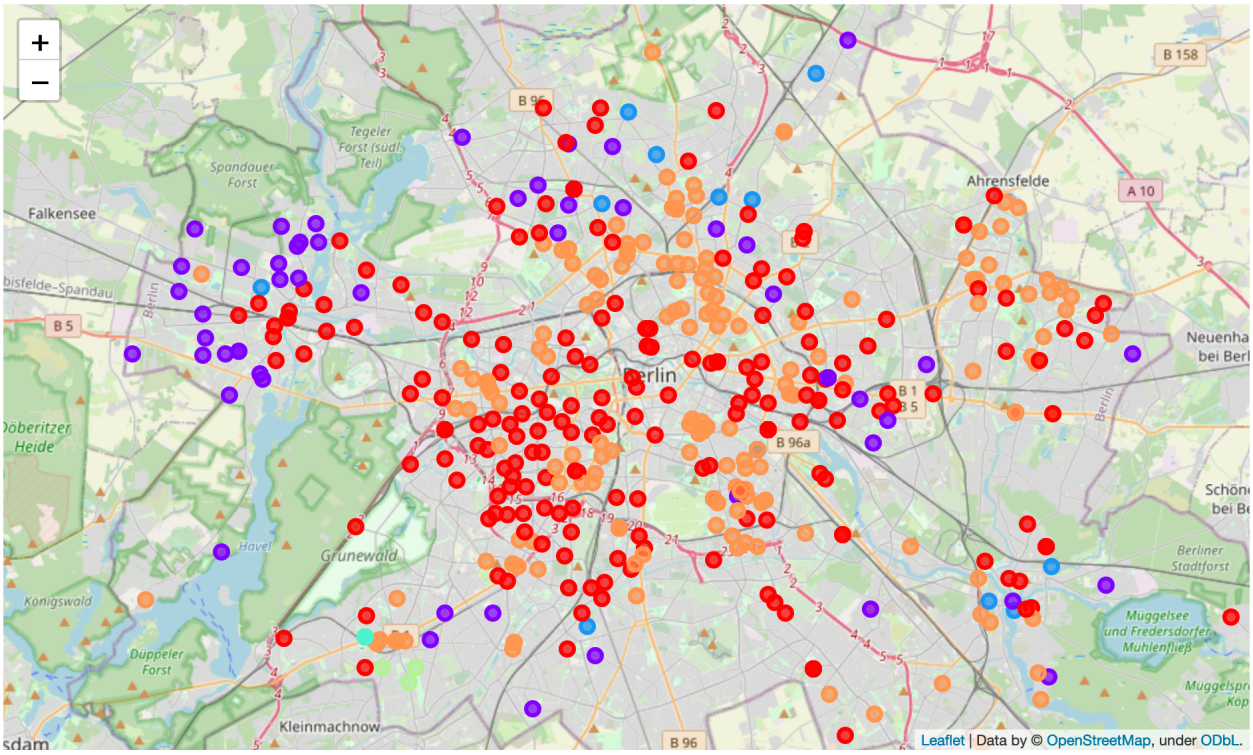


Figure 7: A map of Berlin showing the spacial distribution of the different clustered neighborhoods

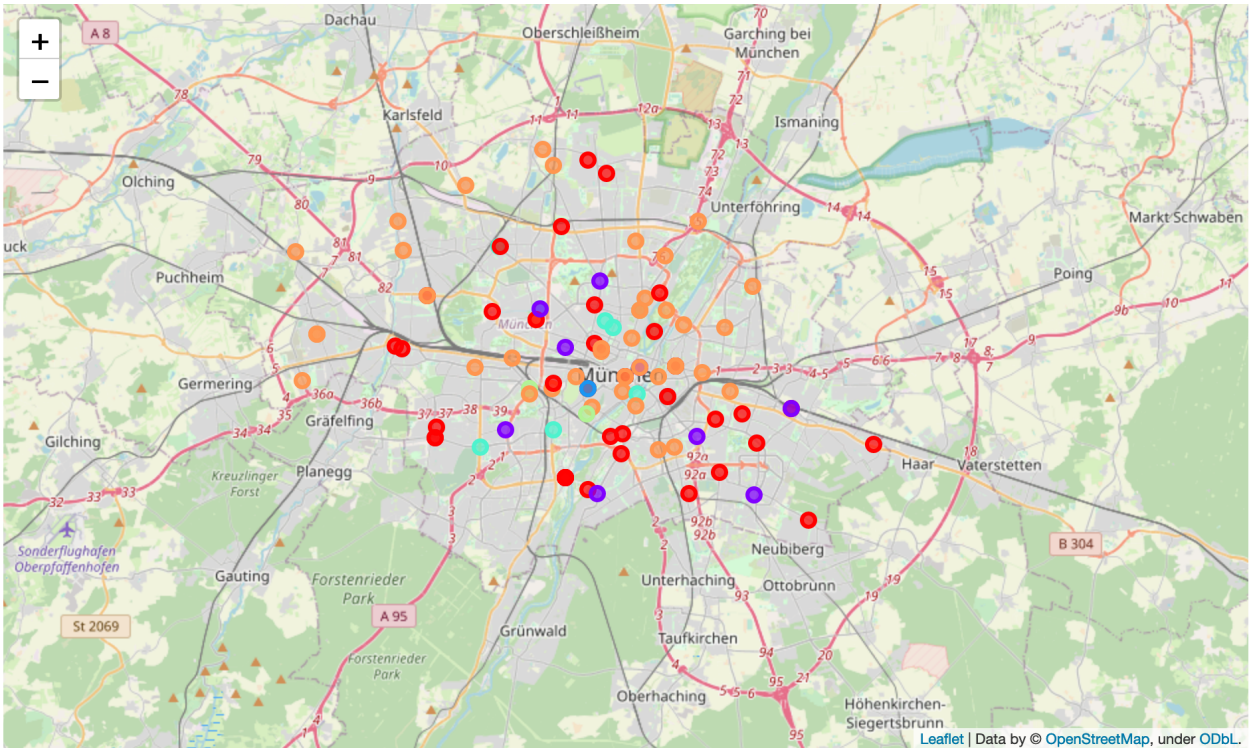


Figure 8: A map of Munich showing the spacial distribution of the different clustered neighborhoods

To illustrate the spacial distribution of the different clusters, they where plotted onto a folium map. Figure

7 and Figure 8 showing the spacial distribution of the different cluster for Berlin and Munich respectively.

	Borough	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
count	281	281	281	281	281	281	281	281	281	281	281
unique	32	44	53	56	63	66	64	67	80	75	71
top	Charlottenburg-Wilmersdorf	Supermarket	Hotel	Supermarket	Bar	Pub	Café	Gym / Fitness Center	Rock Club	Nightclub	Hotel
freq	35	43	30	27	24	26	34	38	22	22	30

Figure 9: Main attributes of the cluster 0

	Borough	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
count	255	255	255	255	255	255	255	255	255	255	255
unique	30	19	35	36	43	50	50	58	66	63	66
top	Pankow	Café	Café	Drugstore	Italian Restaurant	Bakery	Ice Cream Shop	Supermarket	Sushi Restaurant	Clothing Store	Dance Studio
freq	38	161	45	29	35	34	30	33	26	26	24

Figure 10: Main attributes of the cluster 5

	Borough	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
count	76	76	76	76	76	76	76	76	76	76	76
unique	19	8	11	26	29	25	25	21	18	16	12
top	Spandau	Bus Stop	Bus Stop	Drugstore	Drugstore	Pet Store	Pet Café	Pet Café	Peruvian Restaurant	Persian Restaurant	Pet Café
freq	24	33	31	8	6	10	10	12	12	12	15

Figure 11: Main attributes of the cluster 1

Figure 9, 10 and 11 showing the main features of the cluster 0, 5 and 1 respectively, which are the most important cluster. While for cluster 0 the most important features are supermarkets hotels and bars, cluster 5 is manly characterized by cafe drugstores and restaurants. Cluster 1 is characterised by a lower shop and restaurant density since bus stops are the most pronounced features

5 Discussion

To compare the to cities one can say that in the case of Munich the cities neighbourhoods a very scattered, with almost all kinds of cluster present. Berlin is in this sense much more structured with clusters the inner city are mostly part of type 0 and 5. The other cluster types are more located in the outskirts of the cities.

6 Summary and Conclusions

In summary it was shown that by clustering Berlin and Munich based on local venues, it is possible to categorize the neighborhood in both cities. Based on these categories it is possible to give recommendations for finding similar neighbourhoods in both cities.