IBM Data Science Capstone Project

Oliver Schackmann

Introduction

In this project, I will try to contribute to solving a widespread problem in my hometown, Munich. Since two of Germany's best universities (LMU and TUM) are located in Munich, a lot of young people move there. Unfortunately, Munich is known for its high rents. My goal is to help students to find the least expensive districts. Additionally, with the help of machine learning, the neighborhoods will be clustered into a few groups based on their venues related to free-time activities. The characteristics of each of these groups will be analyzed, so interested students can make a more informed choice on where to rent an apartment.

Data

To achieve this, we have to use multiple data sources. Rental prices, sorted by districts, can be found in an article by a local newspaper: https://www.tz.de/leben/wohnen/uebersicht-muenchner-mieten-preise-nach-postleitzahlen-tz-6133643.html. With the help of Foursquare, we will gather information about venues in each area. Using an API request URL, we will receive a json file, which will be used to obtain the name, location, and category of each venue (example below). Furthermore, we will need the coordinates of each district and its shape so that we can create a good looking map. The required geojson file can be downloaded here: https://www.suche-postleitzahl.org/plz-karte-erstellen.

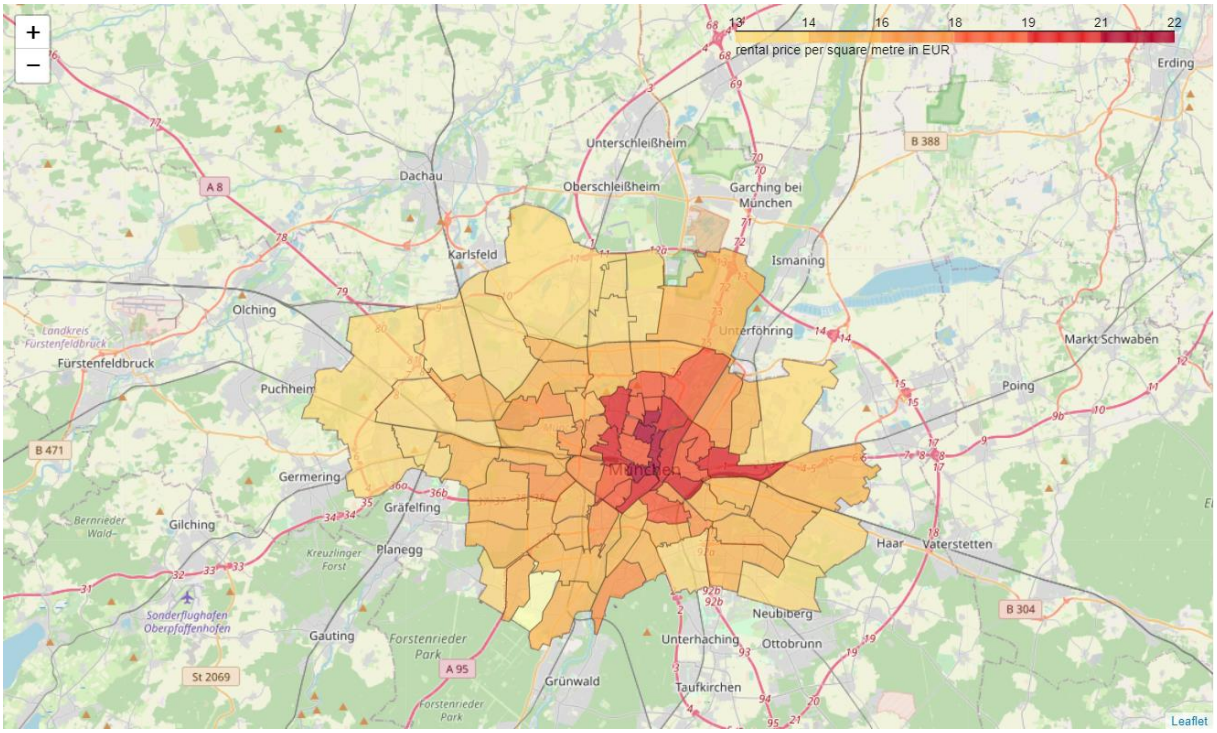
	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	80331	48.135276	11.570982	Ringlers	48.134097	11.568302	Sandwich Place

Methodology

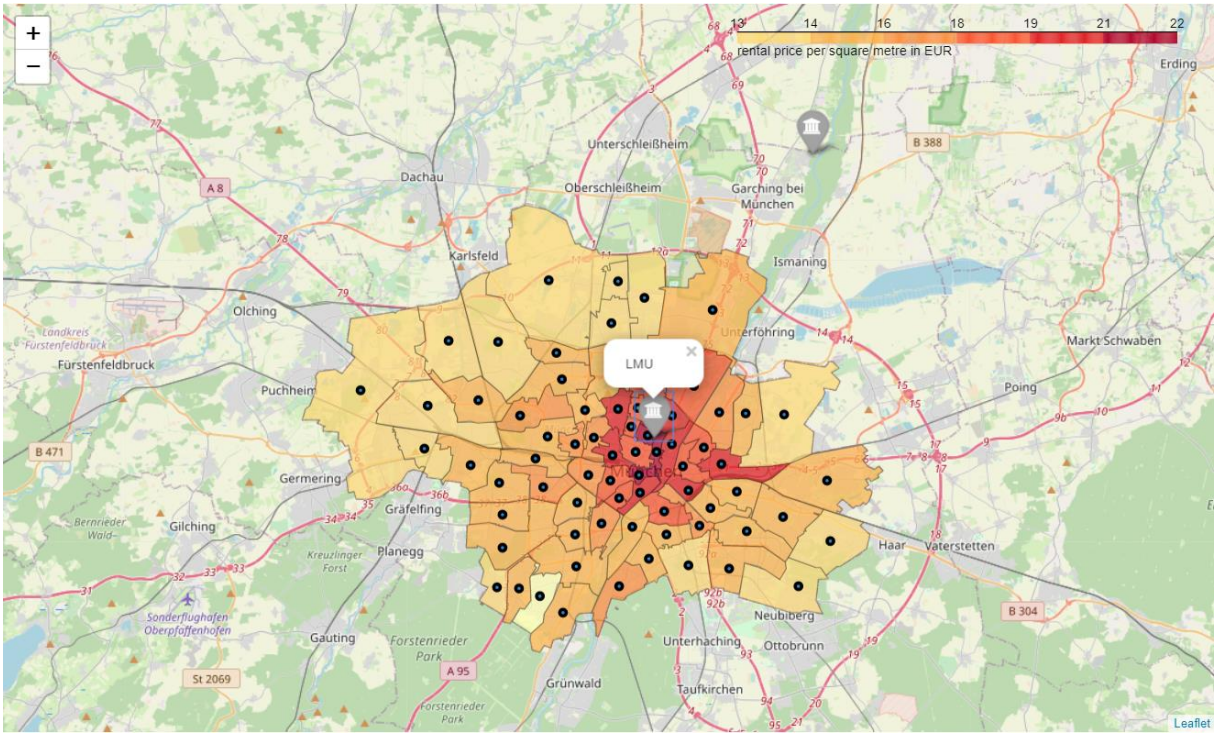
The first thing we have to do is downloading the rental price data. Since the table has a weird format, it has to be read in 3 parts. After that, we have do make a few corrections of wrongly input data, drop duplicate rows (caused by some districts being in multiple boroughs) and merge the three tables. Our resulting dataframe contains the columns "plz "(German abbreviation for "postal code ") and "rent "and looks like this:

	plz	rent
0	80995	14.10
1	80997	13.25
2	80999	13.05
3	81247	14.55
4	81249	13.25

Next, we will visualize the rental prices using python's folium module. For our purpose, a choropleth map is best suited. At https://www.suche-postleitzahl.org/plz-karte-erstellen, the required geojson file can be found. With its help, folium knows the shapes of each district and can generate this interactive map:



Another geojson file tells us the coordinates of each district's center. We will add markers for each center and two of the biggest campuses of LMU and TUM:



In the next step, we will utilize Foursquare's (https://www.foursquare.com/) location data. We create an API request URL and receive a json file containing the information described in the "data "section of the report. Looking through the returned venue categories, I noticed that some of them are not suitable for clustering because they exist in all districts. These include 'Supermarket', 'Bus Stop', etc. Therefore, I deleted them manually from the list of categories that Foursquare uses to find venues. Here are the first 5 of 1201 venues total:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	80331	48.135276	11.570982	Ringlers	48.134097	11.568302	Sandwich Place
1	80331	48.135276	11.570982	The High	48.133101	11.572939	Cocktail Bar
2	80331	48.135276	11.570982	TeeGschwendner	48.135398	11.569455	Tea Room
3	80331	48.135276	11.570982	Kleinschmecker	48.134659	11.573565	German Restaurant
4	80331	48.135276	11.570982	landersdorfer & innerhofer - restaurant	48.136237	11.569917	German Restaurant

We have to use one-hot encoding before we can start to cluster the districts. One hot encoding is a process by which categorical variables are converted into a useful form for the k-means algorithm. This is what the dataframe looks like after grouping the dataframe by 'Neighborhood 'and calculating the mean values:

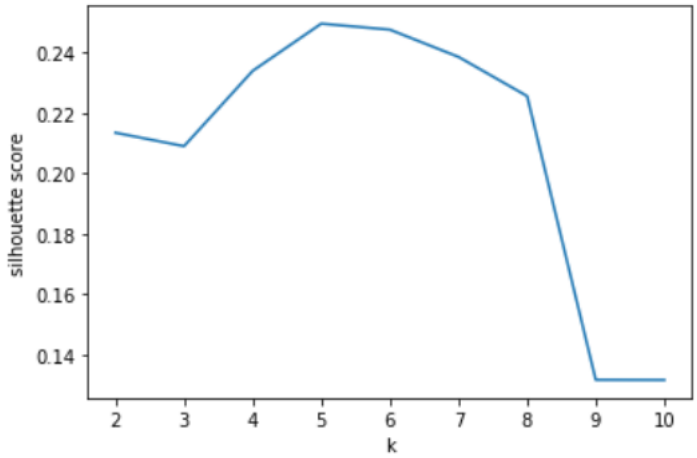
	Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Arcade	Art Gallery	Art Museum	Asian Restaurant	Athletics & Sports	Austrian Restaurant	...	Trattoria/Osteria	Turkish Restaurant	Vegetarian / Vegan Restaurant	Vietnamese Restaurant	Waterfall	Wine Bar	Wine Shop
0	80331	0.02381	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.00	0.0	0.011905	0.011905	0.0	0.000000	0.011905
1	80333	0.00000	0.000000	0.0	0.0	0.017857	0.053571	0.017857	0.000000	0.000000	...	0.00	0.0	0.000000	0.000000	0.0	0.000000	0.000000
2	80335	0.04000	0.000000	0.0	0.0	0.000000	0.000000	0.080000	0.000000	0.000000	...	0.04	0.0	0.000000	0.000000	0.0	0.000000	0.000000
3	80336	0.00000	0.000000	0.0	0.0	0.000000	0.000000	0.062500	0.000000	0.000000	...	0.00	0.0	0.000000	0.000000	0.0	0.000000	0.062500
4	80337	0.00000	0.033898	0.0	0.0	0.016949	0.000000	0.000000	0.016949	0.016949	...	0.00	0.0	0.000000	0.033898	0.0	0.016949	0.000000

5 rows × 188 columns

Having done that, we can sort the values and create a dataframe displaying the most common venues of each neighborhood:

	Neighborhood	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
0	80331	Café	German Restaurant	Bavarian Restaurant	Plaza	Coffee Shop	Cocktail Bar	Clothing Store	Italian Restaurant	Afghan Restaurant	Tea Room
1	80333	Café	History Museum	Nightclub	Plaza	Art Museum	Restaurant	Bar	Fountain	Burger Joint	Movie Theater
2	80335	Middle Eastern Restaurant	Bakery	Asian Restaurant	Coffee Shop	Mexican Restaurant	Fast Food Restaurant	Sausage Shop	Circus	Salad Place	Beer Garden
3	80336	Plaza	BBQ Joint	Middle Eastern Restaurant	Café	Movie Theater	Chinese Restaurant	Cocktail Bar	German Restaurant	Eastern European Restaurant	Bakery
4	80337	Italian Restaurant	Café	Burger Joint	Seafood Restaurant	Plaza	Nightclub	Bakery	German Restaurant	African Restaurant	Greek Restaurant

The next step is clustering the districts, which will be done using the machine learning algorithm k-means. Our goal is to assign each district to one of a few clusters based on the venues located in its area. Seeing which neighborhoods are similar and what their characteristics are should help everyone interested in renting an apartment. First, we want to know the optimal number of clusters. We can determine it by running the silhouette score algorithm:



The maximum being at k = 5 tells us that 5 clusters are the best choice, but 4 and 6 wouldn't be a lot worse.

Running the k-means algorithm and assigning the cluster labels to each district results in this dataframe:

	Cluster Label	Neighborhood	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
0	0	80331	Café	German Restaurant	Coffee Shop	Plaza	Bavarian Restaurant	Italian Restaurant	Cocktail Bar	Clothing Store	Tea Room	Museum
1	0	80333	Café	History Museum	Plaza	Nightclub	Bar	Art Museum	Restaurant	Fountain	Burger Joint	Museum
2	0	80335	Coffee Shop	Asian Restaurant	Middle Eastern Restaurant	Afghan Restaurant	Bavarian Restaurant	Diner	Doner Restaurant	Restaurant	Salad Place	Circus
3	0	80336	Plaza	BBQ Joint	Middle Eastern Restaurant	Café	Movie Theater	Chinese Restaurant	Cocktail Bar	German Restaurant	Eastern European Restaurant	Bakery
4	0	80337	Italian Restaurant	Café	Bakery	Burger Joint	Restaurant	Nightclub	Seafood Restaurant	Plaza	Vietnamese Restaurant	German Restaurant

Before we create the final map, we can examine each cluster. Here are the first rows of cluster 0:

	plz	Cluster Label	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	80331	0	Café	German Restaurant	Coffee Shop	Plaza	Bavarian Restaurant
1	80333	0	Café	History Museum	Plaza	Nightclub	Bar
2	80335	0	Coffee Shop	Asian Restaurant	Middle Eastern Restaurant	Afghan Restaurant	Bavarian Restaurant
3	80336	0	Plaza	BBQ Joint	Middle Eastern Restaurant	Café	Movie Theater
4	80337	0	Italian Restaurant	Café	Bakery	Burger Joint	Restaurant

We can determine the most frequent venue categories of this dataframe:

name	count
Italian Restaurant	27
Café	26
German Restaurant	14
Bakery	12
Park	12

Doing that for all clusters, we can think of descriptions for all clusters:

Cluster 0 (blue): "Restaurants"

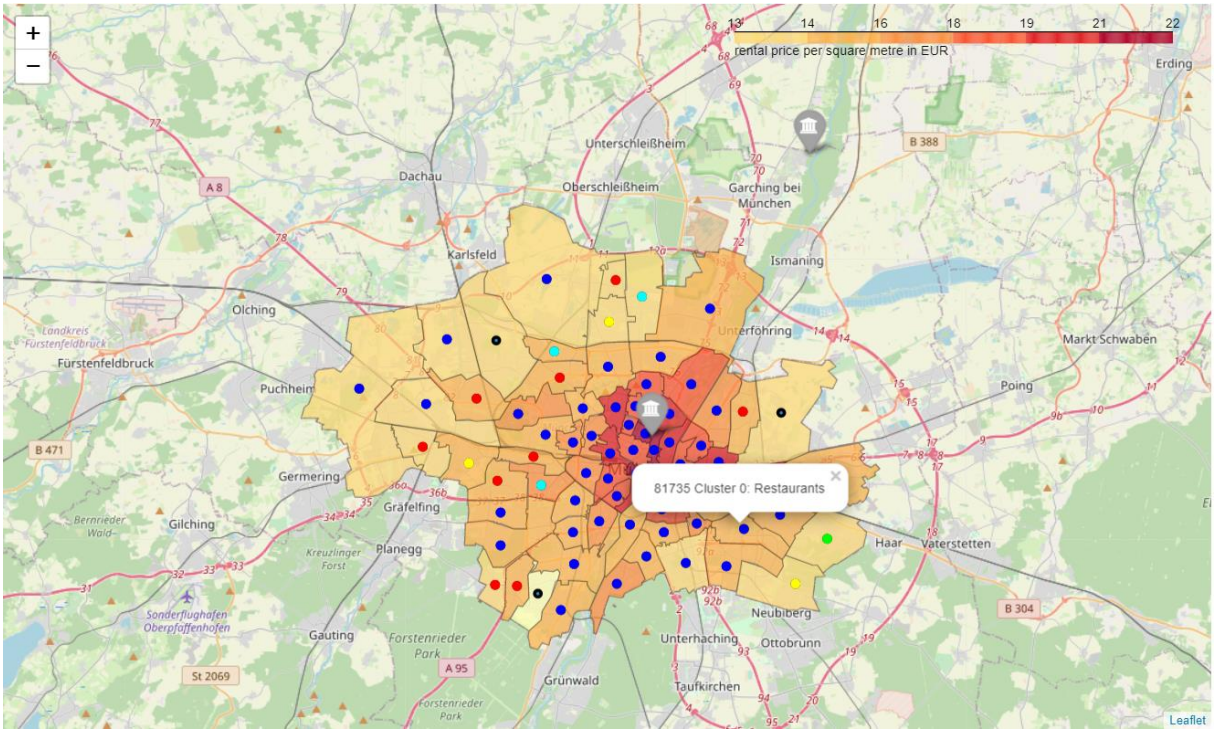
Cluster 1 (cyan): "Food"

Cluster 2 (only one district, green): "Health & Beauty"

Cluster 3 (red): "Sports"

Cluster 4 (yellow): "Beer Gardens"

Finally, we can visualize our results:



Summary

This is what we have done:

- Create a dataframe including the rental prices sorted by postal codes
- Generate a choropleth map with the help of a geojson file based on the rental prices
- Add markers for every district and the two biggest campuses
- Extract interesting venues with Foursquare
- List the ten most common venues for each district
- Cluster the districts with the k-means algorithm, having calculated the optimal number of clusters
- Examine all clusters and display them on the map

Discussion of the Results

Since this project is rather short, it is apparent that its recommendations have to be taken with a grain of salt. There are a few aspects that have to be kept in mind: Firstly, there could be some bargains in a region with high average rental prices. Secondly, low rental prices most likely are low for good reasons, so look out for them. Consequently, I don't recommend choosing a district just based on its average rental price. There are also a few problems using Foursquare (which was a requirement by IBM for this Capstone Project). We can't tell to which extent Foursquare's database provides an accurate representation of each district. Clearly, not every venue can be listed, but not finding even a single venue for three districts is a rather unfortunate result. Additionally, we considered only the relative frequency but not the absolute numbers of venues in each area. This may lead to wrong conclusions, especially for districts where Foursquare found just a few results.

Conclusion

I hope that this map can be a good starting point for everybody considering moving to Munich.

Thank you for reading!