Berliner Restaurant Recommender System

October 2019

By: Özge Celenk

1. Introduction

1.1 Background

Berlin is the capital, as well as the Germany's largest city that is inhabited by approximately 3.5 Million people. The German capital is also quite famous with its multicultural and diverse urban lifestyle, since it is populated by people coming from many different nations and cultures. Recent statistics count 190+ nationalities living in Berlin.

The diversity of Berlin's population also highly reflects in the food options available in the city. Berlin is like a large collection of every food tradition and it reflects the huge diversity of people who have made Germany's new capital their home. Here the restaurants range from Tex-Mex, Mexican, Russian, Italian, Turkish, Iranian, Thai, Indian, Vietnamese, Polish, to Balkan and beyond. Especially Turkish, Indian and Thai cuisine are very popular in the city.

1.2 Problem Description

Whether as a tourist, or even as a local, it can be challenging and sometimes even stressful to choose a restaurant in a vast number of food options in Berlin. One might end up spending so much of their valuable time for searching the restaurant that fits their criteria. Plus, one might also end up paying astronomical prices for just average food, just because it was located nearby a famous tourist attraction. It is also possible that new and upcoming restaurants struggle to be discovered among the older, most rated places. Therefore, it would be very beneficial to have a restaurant recommender system that addresses the following points for each restaurant in a chosen area:

1. The type of food / cuisine
2. The user ratings
3. Distance from the user
4. Difference between the costs of this restaurant and the other "similar" restaurants

To address such questions, a company in Berlin decides to allocate this project to me for building a system that uses Foursquare data to help in recommending new places based on their rankings compared to the previously visited ones.

Expectations from this recommender system is to get answer for the above questions, in such a way that it compiles all of the perspectives of managing recommendations. The Berliner food recommendation system is expected to leverage the Foursquare location data to express the following:

1. What types of restaurants are present in a particular neighbourhood?
2. Based on a particular food preference, where are the related restaurants?
3. How are the rankings of these restaurants with respect to my preferences?

1.3 Target Audience

The target audience of this project is not only limited to tourists but everyone in Berlin who needs help with choosing a restaurant for their specific preferences. Especially for someone who is in the city for a limited time, it would help them choose where to eat without losing time and avoid paying very high prices. It could also prove useful for locals who are tired of going to the same restaurants, and want to discover new places.

2. Data Requirements

2.1 Data Description:

In order to be able to build a food recommendation system in Berlin, lots of location data are needed. For this system, the most important three data attributes that we need from every restaurant in Berlin would be:

1. Its geographical coordinates (latitude and longitude) to find out where exactly it is located.
2. Population of the neighborhood where the restaurant is located.
3. Average income of neighborhood to estimate how much is the restaurant worth.

If we take a closer look at each of these attributes, the geographical coordinates will help us create a map and pin each restaurant on this map according to its geographical coordinates.

Part 1: Collecting the neighbourhoods and boroughs of Berlin, and obtaining their coordinates:

- Pandas library was used for parsing the list of neighbourhoods, boroughs, population and their sizes into a data frame that is available in this webpage. Berlin has 94 districts in total, however, in order to keep the demonstration of the model simple, only the most populated 20 neighbourhoods are kept.
- There was no list available that includes the latitudes and longitudes of the neighbourhoods, so the corresponding coordinates of each neighborhood was added manually to the data frame. Here is a glimpse of the dataset produced after the first two steps:

	Neighborhood	Borough	Population	Area(km2)
0	Neukölln	Neukölln	167248	12.0
1	Prenzlauer Berg	Pankow	160127	11.0
2	Kreuzberg	Friedrichshain-Kreuzberg	153887	10.0
3	Friedrichshain	Friedrichshain-Kreuzberg	127189	9.9
4	Charlottenburg	Charlottenburg-Wilmersdorf	126800	11.0

Figure 1: Collecting the Neighborhoods into a data frame

Part 2: Adding the average income data per neighbourhood

- The information related to the average income per neighbourhood also had to be done manually, using the statistics available here.

- The average income per neighbourhood was also normalized and added to the data frame. The income values available were from 2015 and they may not be accurate today, however it should be accurate enough to create a model for demonstration. Here is a glimpse of the dataset produced after adding the normalized values of the income, area and population to the data frame:

	Neighborhood	Borough	Population	Area(km2)	Latitude	Longitude	Avg_Income_Euro	Normalized_income	Normalized_population	Normalized_area
0	Neukölln	Neukölln	167248	12.0	52.4408	13.4445	1550	0.738095	1.000000	0.342857
1	Prenzlauer Berg	Pankow	160127	11.0	52.5392	13.4242	1850	0.880952	0.957423	0.314286
2	Kreuzberg	Friedrichshain-Kreuzberg	153887	10.0	52.4983	13.4066	1675	0.797619	0.920113	0.285714
3	Friedrichshain	Friedrichshain-Kreuzberg	127189	9.9	52.5158	13.4540	1675	0.797619	0.760481	0.282857
4	Charlottenburg	Charlottenburg-Wilmersdorf	126800	11.0	52.5166	13.3041	1800	0.857143	0.758156	0.314286

Figure 2: Adding population and average income data to the dataset

Part 3: Creating a map of Berlin and adding labels

Using the Folium library, a map of berlin was created, and corresponding labels were added using the coordinates of the 20-most populated neighbourhoods. The generated map of the city is shown below.

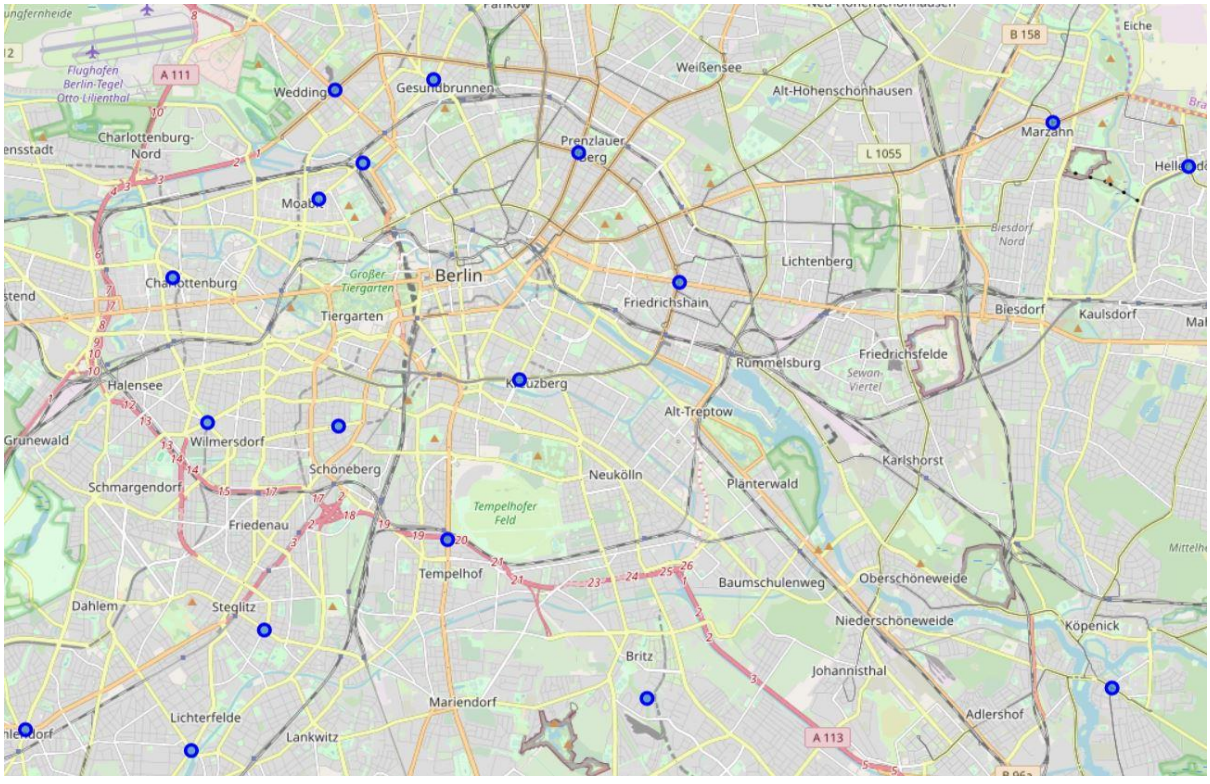

Figure 3: Map of Berlin and added neighborhood labels

Part 4 : Leveraging the Foursquare API

Foursquare API is used to fetch nearest venue locations so that they can be used to form clusters. Foursquare API leverages the power of finding nearest venues in a radius (500m) and also

corresponding coordinates, venue location and names. Upon calling the API, the following dataframe is created:

	Neighborhood	Borough	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Neukölln	Neukölln	52.4408	13.4445	P.A.M. - Pizza And More	52.437406	13.446066	Fast Food Restaurant
1	Neukölln	Neukölln	52.4408	13.4445	Wochenmarkt Britz-Süd	52.437445	13.446328	Food & Drink Shop
2	Neukölln	Neukölln	52.4408	13.4445	H U Britz-Süd	52.437049	13.447439	Bus Stop
3	Neukölln	Neukölln	52.4408	13.4445	U Britz-Süd	52.437038	13.448485	Metro Station
4	Prenzlauer Berg	Pankow	52.5392	13.4242	Grand Tang Xi Yu	52.537738	13.423279	Chinese Restaurant

Figure 4: Finding the venues using the Foursquare API

3. Methodology

3.1 Exploratory Data Analysis

After collecting the data, we need to explore the current state of dataset and then list up all the features needed to be fetched.

Exploring the dataset is important because it gives initial insights and may help getting an idea of the answers that you are seeking from the data.

While exploring the dataset, I found out that the most populated five neighborhoods in Berlin also happens to have the greatest number of venues. The graph that show the number of venues per neighborhood is shown below.

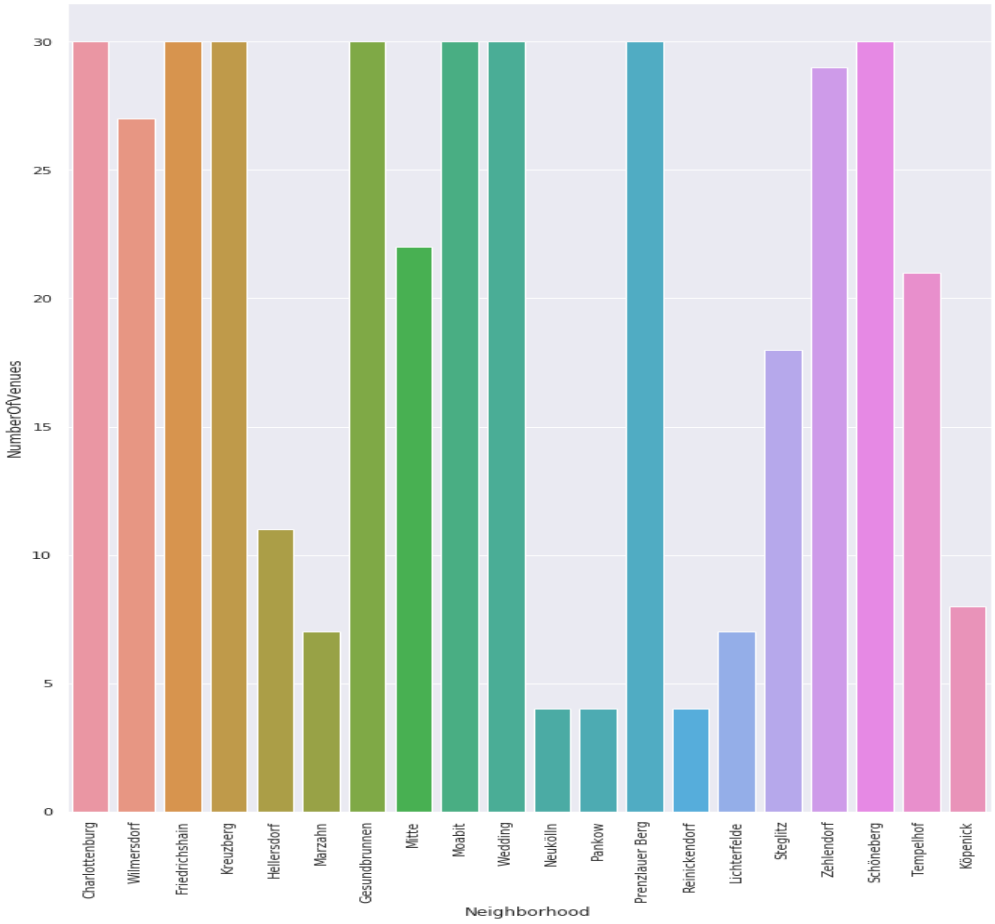

Figure 5: Number of Venues per Neighborhood

In order to get a better idea about each neighborhood and what type of venues are popular there, the categories of the top-5 most popular venues in each are summarized per neighborhood. In the example below we can see the results for Charlottenburg and Friedrichshain.

```
--------- Charlottenburg ---------
                              Category  Frequency
0  Venue Category_Italian Restaurant       0.10
1              Venue Category_Bakery       0.10
2         Venue Category_Supermarket       0.10
3                Venue Category_Café       0.10
4         Venue Category_Pizza Place       0.07


--------- Friedrichshain ---------
                         Category  Frequency
0            Venue Category_Café       0.13
1  Venue Category_Ice Cream Shop       0.07
2     Venue Category_Pizza Place       0.07
3              Venue Category_Pub       0.07
4     Venue Category_Coffee Shop       0.07
```

Figure 6: Most popular venue categories per neighborhood (Example Charlottenburg and Friedrichshain)

3.2 K-means Clustering

After the data analysis, K- means clustering is applied to the data and the neighborhoods are clustered into three groups. The number for k was chosen by using the elbow method. After that, every datapoint on the dataset is assigned to a cluster, and the map is generated showing the clusters with different colors. The resulting map will be presented in the results section.

4. Results

The following map shows the map of Berlin with the neighbors assigned to three clusters.

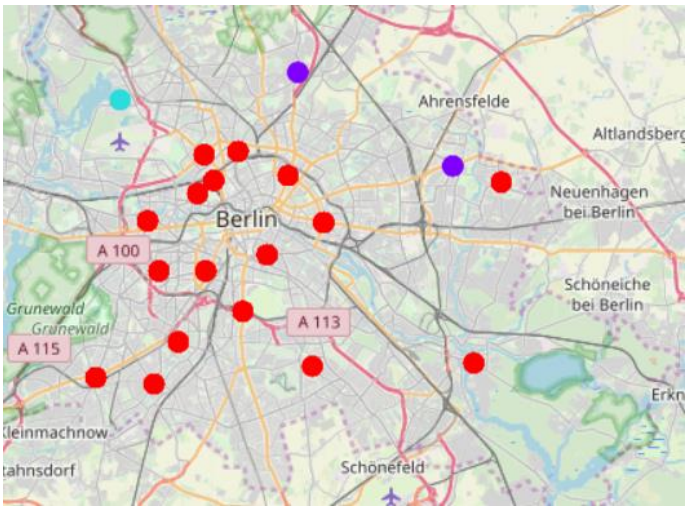

Figure 7: Map of Berlin with the neighbors assigned to three clusters

The result of the recommender system is that it produces a list of top restaurants and the most common venue item that the user can enjoy. During the runtime of the model, a simulation was done by taking ‘Marzahn’ as the neighborhood and then processed through our model so that it could recommend neighborhoods with similar venue characteristics as that of ‘Marzahn’.

The following image shows the result:

```
possible_neighborhoods = neighborhoods_venues_sorted[neighborhoods_venues_sorted['Cluster']==target_cluster]
possible_neighborhoods.tail()
```

]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	Venue Category_African Restaurant	Venue Category_Art Gallery	Categor Re
152	Marzahn	Venue Category_Tram Station	Venue Category_Asian Restaurant	Venue Category_Supermarket	Venue Category_German Restaurant	Venue Category_Ice Cream Shop	0.0	0.0	
209	Pankow	Venue Category_Tram Station	Venue Category_Asian Restaurant	Venue Category_Supermarket	Venue Category_Hotel	Venue Category_Electronics Store	0.0	0.0	
210	Pankow	Venue Category_Tram Station	Venue Category_Asian Restaurant	Venue Category_Supermarket	Venue Category_Hotel	Venue Category_Electronics Store	0.0	0.0	
211	Pankow	Venue Category_Tram Station	Venue Category_Asian Restaurant	Venue Category_Supermarket	Venue Category_Hotel	Venue Category_Electronics Store	0.0	0.0	
212	Pankow	Venue Category_Tram Station	Venue Category_Asian Restaurant	Venue Category_Supermarket	Venue Category_Hotel	Venue Category_Electronics Store	0.0	0.0	

5 rows × 152 columns

```
print("There are {} neighborhoods which has similar venue characteristics to Marzahn.".format(possible_neighborhoods.shape[0])
```

```
There are 11 neighborhoods which has similar venue characteristics to Marzahn.
```

Figure 8: Recommender System output (Example Marzahn)

5. Discussion

The key phase of this project were the data analysis and cluster forming phases. During the clustering, similar neighborhoods must be dumped into the right cluster. During the cluster forming phase I have noticed that the results can be very diverse depending on the number of clusters chosen, and in order to form more balanced clusters for this case, perhaps more data is needed.

In the future I would like to improve the model by taking all neighborhoods into account instead of the limited number of neighborhoods.

6. Conclusion

The aim of this project was to demonstrate a food recommendation system in Berlin that points out similar venues that matches the specific preferences of the user, using K-Means clustering and leveraging the Foursquare data.

The desired result was obtained from the model as discussed in the “Results” section; however, the model still has to be improved. For demonstrating the concept, only the top-20 neighborhoods’ data were collected. In order to improve the recommendation system, data from all of the neighborhoods should be used. Also, further preprocessing steps should be considered in order to achieve a cleaner dataset.