Final Assignment – The Battle of Neighborhoods

By Håkon Ronold Mathisen

Contents

About

This assignment marks the end of the IBM Data Science Professional Certificate. As a final task we were challenged to, based on what we have learned during the Capstone course (final course in the IBM Data Science Professional Certificate), to solve a Data Science related business problem while using the Foursquare location API.

Introduction & Business Problem

The Foursquare API enables us access to many interesting geolocations and venue information (e.g. restaurant reviews and coordinates). This gives us (for an example) the opportunity to research where, based upon certain neighborhoods venue information, a new restaurant optimally could be placed.

I have chosen to try and use this information to optimally place a Food Truck (serving mostly burgers and hot dogs) in a city metropole with many people, amazing restaurants and big business opportunities.

I can formulate my business problem as following:

- *A client has requested me to find the best possible location for her Food Truck (serving mostly burgers and hot dogs) in Berlin, Germany.*

With the following requirements:

- *It should be in one of the more central boroughs of Berlin*
- *It should be in the neighborhood that maximizes her potential and predicted customer base.*

To solve this business problem, I will need to research the best possible neighborhood by locating venues/locations where potential customers reside. This could be office spaces, sport arenas etc. In addition to this I need to consider other venues which could rival the food truck of its customers. As the client only has one Food Truck, I need to find the one best suitable neighborhood to place it.

To pinpoint the best suitable neighborhood to place it, I have chosen three positive drivers and three negative drivers. Positive drivers will increase a neighborhood's suitability, while negative drivers will decrease a neighborhood's suitability.

Driver	Effect
Nearby Offices	Positive
University related buildings	Positive
Shopping Malls	Positive
Other food trucks	Negative
Burger joints	Negative
Hot Dog joints	Negative

Table 1 - Drivers

The drivers will be further discussed in the discussion section of this report.

This business problem is a relevant and potentially very interesting. Not only because it will help us pinpoint the optimal Food Truck location, but because the idea and approach could be applied in other similar situations.

Data Description

Berlin is the capital and the largest city of Germany. It has a huge number of exciting restaurants and incredible nightlife, renowned universities and beautiful architecture. Berlin's 3.7 million inhabitants live across 13 different boroughs ("Bezirk") in 96 official recognized localities ("Ortsteil").

To solve the business problem, I first need to get all borough names and their corresponding localities. After doing this I will use the names to find their coordinates (latitude and longitude). This will enable me to find nearby venues using the Foursquare API.

Boroughs and Localities

The first thing I need to do is to find an overview of Berlin's 13 boroughs and their localities. The following information has been scraped from Wikipedia:

Boroughs:

- Charlottenburg-Wilmersdorf
- Friedrichshain-Kreuzberg
- Lichtenberg
- Marzahn-Hellersdorf
- Mitte
- Neukölln

- Pankow
- Renickendorf
- Spandau
- Steglitz-Zehlendorf
- Tempelhof-Schöneberg
- Treptow-Köpenick

Their corresponding neighborhoods are also found using Wikipedia (Boroughs and neigborhoods of Berlin).

As one of the requirements was to place the food truck in a neighborhood within one of the more central boroughs, I can already start to eliminate some of them. I end up with the following 6 boroughs: Mitte, Friedrichshain-Kreuzberg, Neukölln, Charlottenburg-Wilmersdorf, Tempelhof-Schöneberg and Pankow. The number of localities/neighborhoods have been reduced to 39.

Importing excel files containing Berlin (central) boroughs and their corresponding localities

In [9]:
```
df_berlin_boroughs = pd.read_excel("berlin_boroughs.xlsx")
df_berlin_localities = pd.read_excel("berlin_localities.xlsx")
df_berlin_initial = pd.concat([df_berlin_boroughs, df_berlin_localities], axis=1).reindex(df_berlin_boroughs.index)
df_berlin_initial.groupby('Borough').count()
```

Out[9]:

	Locality
Borough	
Charlottenburg-Wilmersdorf	7
Friedrichshain-Kreuzberg	2
Mitte	6
Neukölln	5
Pankow	13
Tempelhof-Schöneberg	6

Figure 1 - Borougs and their localities

Using the names of the borough and their corresponding localities/neighborhoods I can find both longitude and latitude using Geocoder Nominatim. This enables me to later create a map of Berlin and finding each locality's nearby venues.

In [10]:
```
df_berlin_complete.head()
```

Out[10]:

	Borough	Locality	Latitude	Longitude
0	Mitte	Mitte	52.517690	13.402376
1	Mitte	Moabit	52.530102	13.342542
2	Mitte	Hansaviertel	52.519123	13.341872
3	Mitte	Tiergarten	52.509778	13.357260
4	Mitte	Wedding	52.550123	13.341970

Figure 2 - Coordinates

After finding all coordinates, I create a map of Berlin using Folium.

In [11]:
```
map_berlin
```
Out[11]:

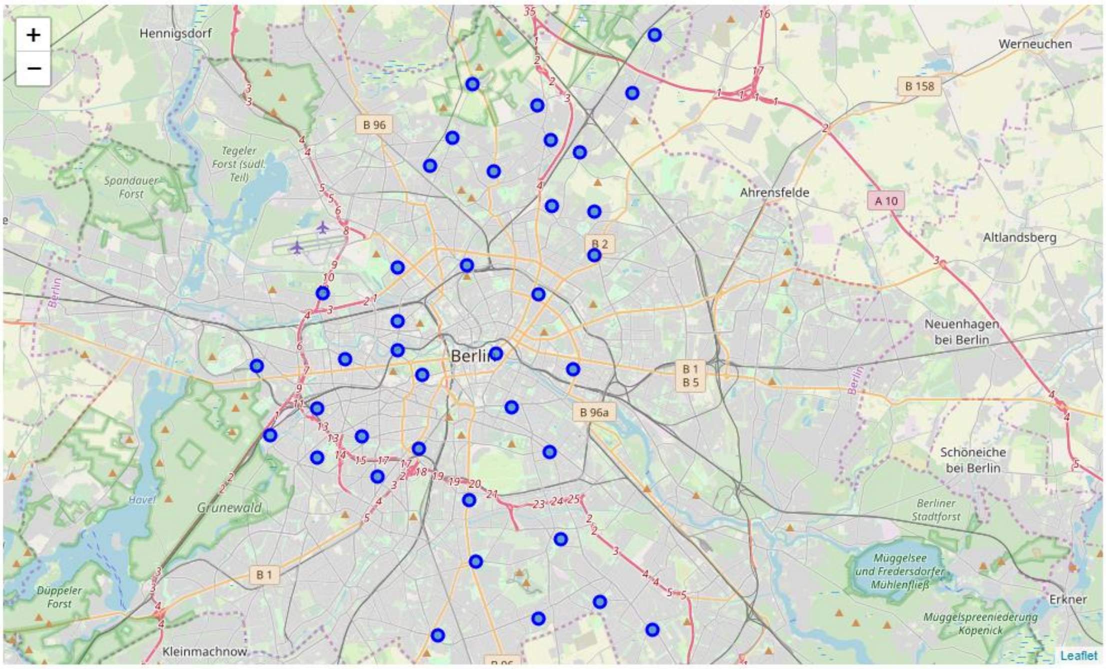


Figure 3 - Map of Berlin and its localities

Venues

Now that I have found all the necessary coordinates, I use them to pinpoint nearby venues. This will be done using the Foursquare API. In total there are 4 841 venues for the 39 localities (given a venues radius of 1000).

Offices

Using the Foursquare API, I find that there are in total 902 office buildings in the localities. These are represented as green dots in the map below.

In [27]:
```
map_berlin_offices = folium.Map(location=[latitude, longitude], zoom_start=12)
addToMap(berlin_venues_offices, 'green', map_berlin_offices)
map_berlin_offices
```
Out[27]:

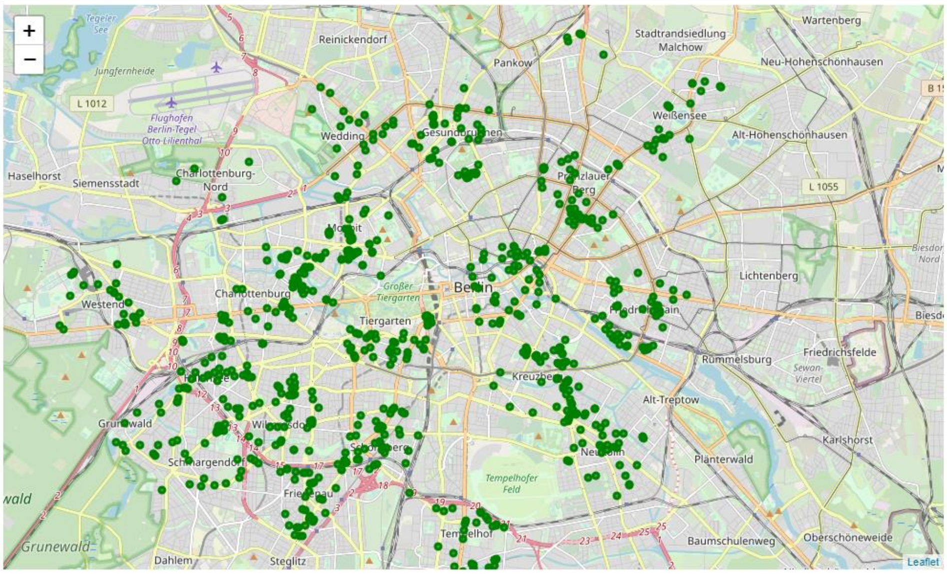


Figure 4 - Map Offices

University Buildings

Using the Foursquare API, I find that there are in total 641 University related buildings in the localities. These are represented as blue dots in the map below.

In [30]:

```
map_berlin_universities = folium.Map(location=[latitude, longitude], zoom_start=12)
addToMap(berlin_venues_universities, 'blue', map_berlin_universities)
map_berlin_universities
```

Out[30]:

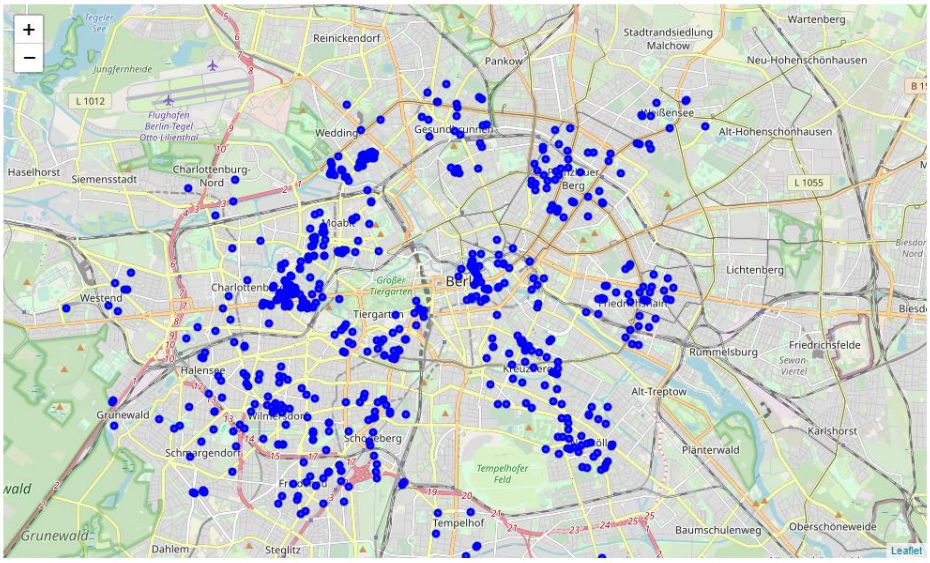


Figure 5 - Map University Buildings

Shopping Malls

Using the Foursquare API, I find that there are in total 74 Shopping Malls in the localities. These are represented as orange dots in the map below.

In [36]:

```
map_berlin_shopping_malls = folium.Map(location=[latitude, longitude], zoom_start=12)
addToMap(berlin_venues_shopping_malls, 'orange', map_berlin_shopping_malls)
map_berlin_shopping_malls
```

Out[36]:

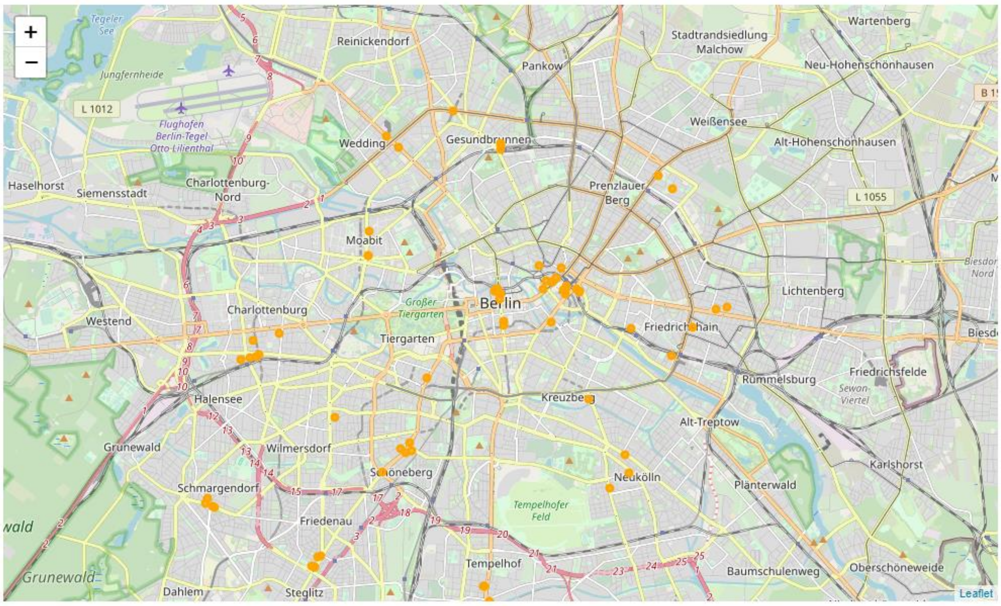


Figure 6 - Map Shopping Malls

Food Trucks

Using the Foursquare API, I find that there are in total 81 food trucks in the localities. These are represented as red dots in the map below.

In [39]:
```
map_berlin_food_trucks = folium.Map(location=[latitude, longitude], zoom_start=12)
addToMap(berlin_venues_food_trucks, 'red', map_berlin_food_trucks)
map_berlin_food_trucks
```

Out[39]:

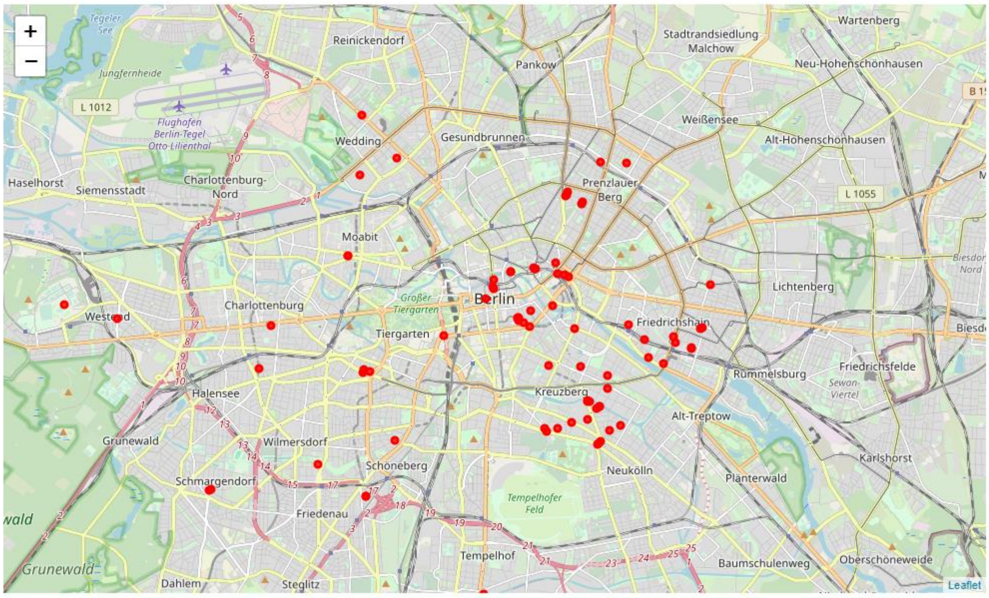


Figure 7 - Map Food Trucks

Burger Joints

Using the Foursquare API, I find that there are in total 169 burger joints in the localities. These are represented as black dots in the map below.

In [61]:
```
map_berlin_burger = folium.Map(location=[latitude, longitude], zoom_start=12)
addToMap(berlin_venues_burger_joints, 'black', map_berlin_burger)
map_berlin_burger
```

Out[61]:

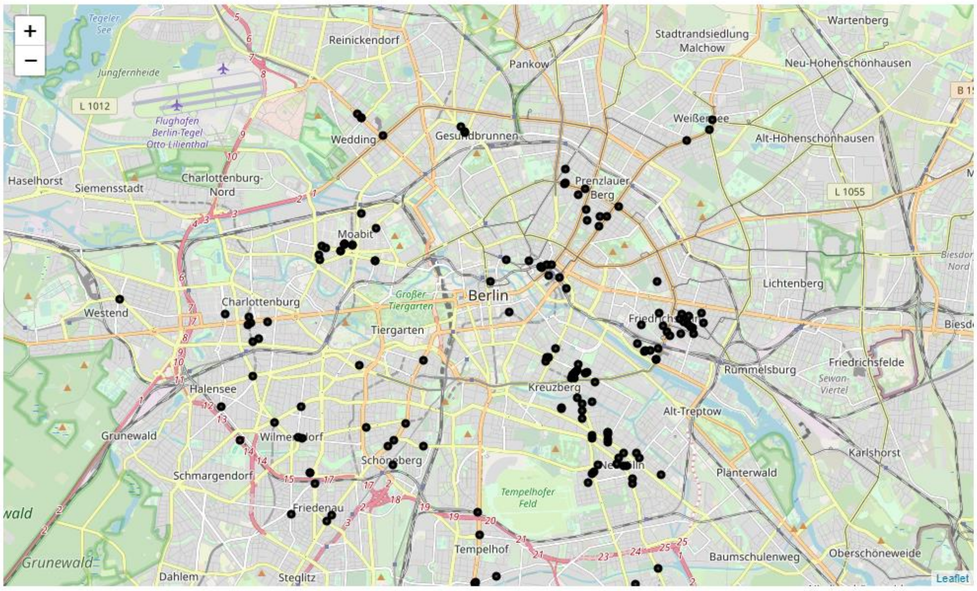


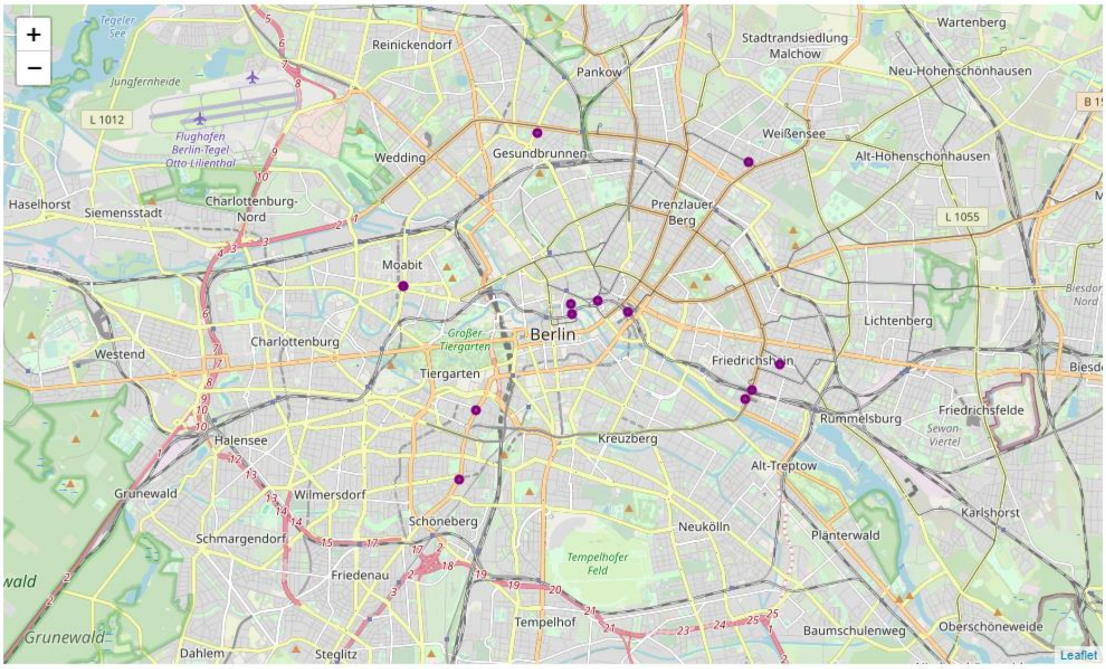
Figure 8 - Map Burger Joints

Hot Dog Joints

Using the Foursquare API, I find that there are in total 14 Hot Dog joints in the localities. These are represented as purple dots in the map below.

```
In [47]: map_berlin_hot_dogs = folium.Map(location=[latitude, longitude], zoom_start=12)
         addToMap(berlin_venues_hot_dogs, 'purple', map_berlin_hot_dogs)
         map_berlin_hot_dogs
```

Out[47]:



Methodology

After finding all localities nearby venues (Offices, University related buildings, Shopping Malls, Food Trucks, Burger joints, Hot Dog joints) I count how many times each venue occur in every locality.

```
In [32]: df_data = df_berlin_complete.copy()
         add_column(df_data, 'Offices', berlin_venues_offices)
         add_column(df_data, 'Universities', berlin_venues_universities)
         add_column(df_data, 'Shopping Malls', berlin_venues_shopping_malls)
         add_column(df_data, 'Food Trucks', berlin_venues_food_trucks)
         add_column(df_data, 'Burger Joints', berlin_venues_burger_joints)
         add_column(df_data, 'Hot Dogs Joints', berlin_venues_hot_dogs)
         df_data.head()
```

Out[32]:

	Borough	Locality	Latitude	Longitude	Offices	Universities	Shopping Malls	Food Trucks	Burger Joints	Hot Dogs Joints
0	Mitte	Mitte	52.517690	13.402376	50.0	50.0	24.0	22.0	13.0	4.0
1	Mitte	Moabit	52.530102	13.342542	49.0	32.0	2.0	1.0	11.0	1.0
2	Mitte	Hansaviertel	52.519123	13.341872	47.0	44.0	1.0	1.0	6.0	1.0
3	Mitte	Tiergarten	52.509778	13.357260	47.0	36.0	1.0	4.0	2.0	1.0
4	Mitte	Wedding	52.550123	13.341970	20.0	49.0	2.0	3.0	4.0	0.0

Figure 9 - Summing up venues

For finding which locality is the most suited for the client's food truck I first need to assign a weighting value for each venue. The amount of the specific venue will be multiplied with corresponding weighting value.

I have assigned the highest (positive) weight on nearby offices. The assumption is that the Food Truck's presence nearby offices will lead to many lunch and dinner customers. The next highest (positive) weight has been assigned to nearby University related buildings. Being close to University related buildings will probably lead to a lot of customers (young people have a greater tendency to eat fast food). The third (positive) weight has been assigned Shopping Malls. The assumption here is that a Shopping Mall may attract many people (and thereby increasing the expected customer base). However, the weight is relatively small. The reason for this is that shopping malls usually has a lot of different food stalls (food courts etc.) and therefore bring some degree of competition.

The largest (negative) weight has been assigned to other nearby Food Trucks as these will be direct competition to the client's food truck. Burger joints have been given a negative weight of -1.0 as the food truck will mostly serve burgers, while Hot dog joints have been given a negative weight of -0.5. See table 2 below.

Driver	Effect
Nearby Offices	+1.5
University related buildings	+1.0
Shopping Malls	+0.5
Other food trucks	-1.5
Burger joints	-1.0
Hot Dog joints	-0.5

Table 2 - Drivers and weights

To be able to draw any conclusion about the most suitable locality, I sum up each venue multiplied with its corresponding weighting value.

Results

Using the weights to calculate optimal neighborhood

In [41]:
```
df_weighted = df_data.copy()
df_weighted['Score'] = df_data['Offices']*weight_offices + df_data['Universities']*weight_universities + df_data['Shopping Malls
df_weighted = df_weighted.sort_values(by=['Score'], ascending=False)
df_weighted.head()
```

Out[41]:

	Borough	Locality	Latitude	Longitude	Offices	Universities	Shopping Malls	Food Trucks	Burger Joints	Hot Dogs Joints	Score
21	Charlottenburg-Wilmersdorf	Charlottenburg	52.515747	13.309683	48.0	47.0	5.0	1.0	7.0	0.0	113.0
2	Mitte	Hansaviertel	52.519123	13.341872	47.0	44.0	1.0	1.0	6.0	1.0	107.0
5	Mitte	Gesundbrunnen	52.550920	13.384846	49.0	27.0	4.0	0.0	3.0	1.0	99.0
3	Mitte	Tiergarten	52.509778	13.357260	47.0	36.0	1.0	4.0	2.0	1.0	98.5
22	Charlottenburg-Wilmersdorf	Wilmersdorf	52.487115	13.320330	43.0	36.0	1.0	1.0	5.0	0.0	94.5

Figure 10 - Optimal Locality

After summing up each locality venues multiplied with its corresponding weight, I find that the locality Charlottenburg (in borough Charlottenburg-Wilmersdorf) is the most suitable (based on chosen venues and weights). This locality will maximize the client's predicted customer base. Charlottenburg get a score of 113.0 where, the second-best locality, Hansaviertel in borough Mitte, gets a score of 107.0.

Charlottenburg is the most suited locality because it has many nearby offices (48 in total), university related buildings (47 in total), some shopping malls (5 in total), while having very little competition (1 other food truck, 7 burger joints and 0 hot dog joints).

```
map_berlin_results
```

Out[60]:

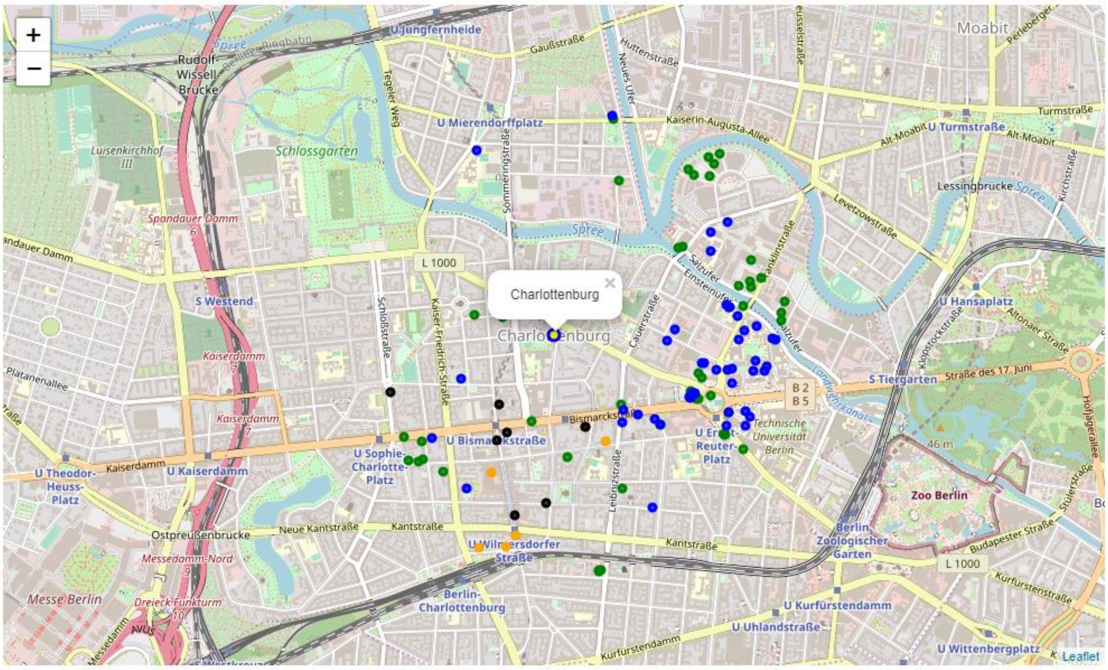


Figure 11 - Charlottenburg Map

A Folium map of Charlottenburg with nearby venues can be found in figure 11.

Discussion

A shown in the result section, the optimal locality to place the food truck (based upon chosen venues and their weights) is Charlottenburg. Figure 12 below sums up how many of the relevant venues that the locality has.

In [41]:
```
df_charlottenburg = df_weighted[df_weighted['Locality']=='Charlottenburg']
df_charlottenburg
```

Out[41]:

	Borough	Locality	Latitude	Longitude	Offices	Universities	Shopping Malls	Food Trucks	Burger Joints	Hot Dogs Joints	Score
21	Charlottenburg-Wilmersdorf	Charlottenburg	52.515747	13.309683	48.0	47.0	5.0	1.0	7.0	0.0	113.0

Figure 12 – Charlottenburg summary

The reason why this was the most suitable locality is that is has many venues that will be advantageous for the client's food truck (many offices, university related buildings, some shopping malls while having very little competition).

While I have shown which locality is the most suited to have the client's food truck, it is important to highlight the weaknesses this model has. First and foremost, the model only includes a few positive and negative drivers, while many other aspects should be considered (e.g. how many people live there, average income in locality, competitive menu etc.). Furthermore, I have taken a radical assumption of being able to place the food truck anywhere I want, when there will be significant placement restrictions (especially in Germany I might add…). Lastly, the weights assigned have been decided arbitrarily instead of being sophistically calculated.

Conclusion

I have in this report shown how to find the suitable location for the client's Food Truck. Out of 39 localities (spread over 6 different boroughs), the most suitable locality is Charlottenburg (in borough Charlottenburg-Wilmersdorf).

This locality was deemed most suitable due to its high density of offices, university related buildings and shopping malls, while having little to no competition.

While this model has clear weaknesses and limitations, I firmly believe this line of thought can be applied to other situations/business problems.