Analysis for Business Entrepreneurs in Berlin

Azeer Esmail
july/2019

1. Introduction

1.1 Background Description

Berlin is the largest city in Germany by both area and population, and 2nd largest in Europe by population within the city limits.
Tourism figures have more than doubled within the last ten years and Berlin has become the third most-visited city destination in Europe, In 2018, the GDP of Berlin totaled €147 billion, an increase of 3.1% over the year of 2017.
All that coupled with its very high diversity where foreign residents originate from about 190 different countries, and the fact that it's a very active city with a booming economy in many sectors, made it a desired destination not only for tourists and expatriates but also for entrepreneurs and businesses.
Therefore studying the opportunities and understanding the status quo before starting a venture in this city is of big importance and the cornerstone of the business's future success.

1.2 Problem

Despite its very promising characteristics for entrepreneurs, Berlin is a very dynamic city and ever changing, as a resident here, I noticed how frequently small businesses start and close their doors after a relatively short time.
This is because of the lack of knowledge of the demographics and other variables of a certain area.
To avoid such a scenario of loss, and to ensure higher possibility of success, Data-Science will be used to try to understand and decide where a certain kind of business could be successful or not.

1.3 Interested Stakeholders

The interested stakeholders will be those who want to check different variables of success before starting a business in Berlin, whether it's a small business or housing/real estate business.

2. Data Description

I decided to use **neighborhood density**, **business density**, and **tourism** as the deciding variables in this project.

2.1 Data Acquisition

The data was acquired from different sources with different methods and in some cases extrapolated from the already existing data.

The **neighborhoods**, **boroughs** and **neighborhood density** data was scraped from the wiki page: https://en.wikipedia.org/wiki/Boroughs_and_neighborhoods_of_Berlin

Tourism data was acquired from:
https://www.statistik-berlin-brandenburg.de/webapi/jsf/tableView/tableView.xhtml

Venues data was acquired from the Foursquare API

Geojson of neighborhoods (Ortsteile) of Berlin city from:
https://common-data.carto.com/tables/berlin_ortsteile/public/map

2.2 Data Processing

The acquired data from the wiki page was used to create the column of the **neighborhoods**, **neighborhood density** and a column of the corresponding **borough** in the dataframe.

Since berlin has very big 12 boroughs i decided to take a more granular approach so the comparison can be between smaller neighborhoods(ortsteile), however i did not consider all of the neighborhoods in the city, but only the more populous half of them.

As for the **coordinates** of each neighborhood i tried to acquire them by scraping the geohack pages that were linked in each neighborhood wiki page, but unfortunately when I plotted them on the map they were inaccurate, so i tried with geocoder and also those were inaccurate, I finally resorted to calculating them myself using the geojson file that divides Berlin neighborhoods.

I used the geojson file to divide Berlin into its neighborhoods in order to create the choropleth map of the population density.

Tourism data was consisting of **number of guests** and **number of overnight stay** for each **borough**, out of these 2 variables i created the **Tourism metric** variable and attached it to the dataframe.
Since the tourism metric was for the boroughs and not neighborhoods, for each neighborhood I assigned the tourism metric corresponding to its borough.

Venues data was used to create a dataframe with columns of **venue, venue coordinates**, and **venue category** plus the columns of neighborhood and neighborhood coordinates, then it was used to calculate the **most common venues** in each neighborhood, and eventually to cluster the data using K-means algorithm.
In addition this data was used to calculate the **business density** in each neighborhood and out of it the **business density metric** that was created and attached to the dataframe.

Certain measures were taken in order to ensure that the data used was valid and consistent:
When a call is made to fetch venues from Foursquare, the circle of coverage with radius 1000m from a certain neighborhood's coordinates, could cover an area of a nearby neighborhood, this could result in assigning an 'outsider' venue to this neighborhood, this is accounted for and corrected.

3. Methodology

3.1 Tourism data processing

I had the tourism data read from a text file into a dataframe:

	Borough	Number of guests	Overnightstay	Tourism metric
0	Mitte	5732248	13923473	63.106762
1	Friedrichshain-Kreuzberg	1668654	4185545	18.642645
2	Pankow	523334	1467454	6.165038
3	Charlottenburg-Wilmersdorf	2747572	6570319	30.035502
4	Spandau	259017	582374	2.755367
5	Steglitz-Zehlendorf	210769	508449	2.313169
6	Tempelhof-Schöneberg	823926	2040358	9.150990
7	Neukölln	403625	899056	4.276291
8	Treptow-Köpenick	327712	703177	3.416936
9	Marzahn-Hellersdorf	83131	243166	1.000000
10	Lichtenberg	498068	1226725	5.518084
11	Reinickendorf	224496	521538	2.422645

For the tourism metric, first the overnight stay and guest numbers were normalized by the minimum value of each:

overnightstay_metric = overnightstay / min(all_overnightstay)
guests_metric = guests / min(all_guests)

Then the tourism metric was calculated as the average of both:

tourism_metric = (guests_metric + overnightstay_metric)/2

The reason behind normalizing **by the minimum and not the average** is because its a **'battle of the neighborhoods',** meaning that when we visualize the data we want to see **how each neighborhood scale with respect to the other** neighborhoods and not to the collective, with the minimum being the smallest.

3.2 Neighborhood Density Data

Was scraped from wiki page for 96 neighborhoods so the tourism data had to be adjusted because its only for the 12 general boroughs:

	Neighborhood	Borough	Neighborhood density	Tourism metric
0	Mitte	Mitte	7445	63.106762
1	Moabit	Mitte	8993	63.106762
2	Hansaviertel	Mitte	11111	63.106762
3	Tiergarten	Mitte	2415	63.106762
4	Wedding	Mitte	8273	63.106762
5	Gesundbrunnen	Mitte	13496	63.106762
6	Friedrichshain	Friedrichshain-Kreuzberg	11662	18.642645
7	Kreuzberg	Friedrichshain-Kreuzberg	14184	18.642645
8	Prenzlauer Berg	Pankow	12991	6.165038
9	Weißensee	Pankow	5736	6.165038

3.3 Calculating Coordinates

As mentioned before the coordinates that were scraped from the web and those that were acquired by geocoder were off when plotted on the map, some neighborhoods had 2 points and others non.
So from the **geojson file** that i used to plot the choropleth map of the population density i extracted the **neighborhood polygons**, and calculated the coordinates:

centroid=[(((max(lat)-min(lat))/2) + min(lat) **),** **(** (max(lng)-min(lng)) /2) + min(lng) **)]**

Basically I took as **latitude** halfway between the furthest vortex north and furthest south of the polygon of the neighborhood
And did the same east to west for the **longitude.**

- A better and more sophisticated calculation could have been performed, where we calculate the ‘center of mass’, but it will it would have required a lot of programming time and math and I don't know if there are any libraries that could help with that.

After the calculation the coordinates where added:

	Neighborhood	Borough	Neighborhood density	Tourism metric	Latitude	Longitude
0	Mitte	Mitte	7445	63.106762	52.522220	13.397631
1	Moabit	Mitte	8993	63.106762	52.529050	13.342629
2	Hansaviertel	Mitte	11111	63.106762	52.518229	13.342319
3	Tiergarten	Mitte	2415	63.106762	52.510762	13.353846
4	Wedding	Mitte	8273	63.106762	52.550058	13.338227
5	Gesundbrunnen	Mitte	13496	63.106762	52.550437	13.384709
6	Friedrichshain	Friedrichshain-Kreuzberg	11662	18.642645	52.508536	13.455603
7	Kreuzberg	Friedrichshain-Kreuzberg	14184	18.642645	52.496556	13.410784
8	Prenzlauer Berg	Pankow	12991	6.165038	52.539254	13.434476
9	Weißensee	Pankow	5736	6.165038	52.552953	13.462319

3.4 Halving the data

Then I decided to work only with the more populous half of the neighborhoods since the Foursquare API would return very few venues for the less dense half which will cause an inflated sense of ratio when visualizing the more populous half:

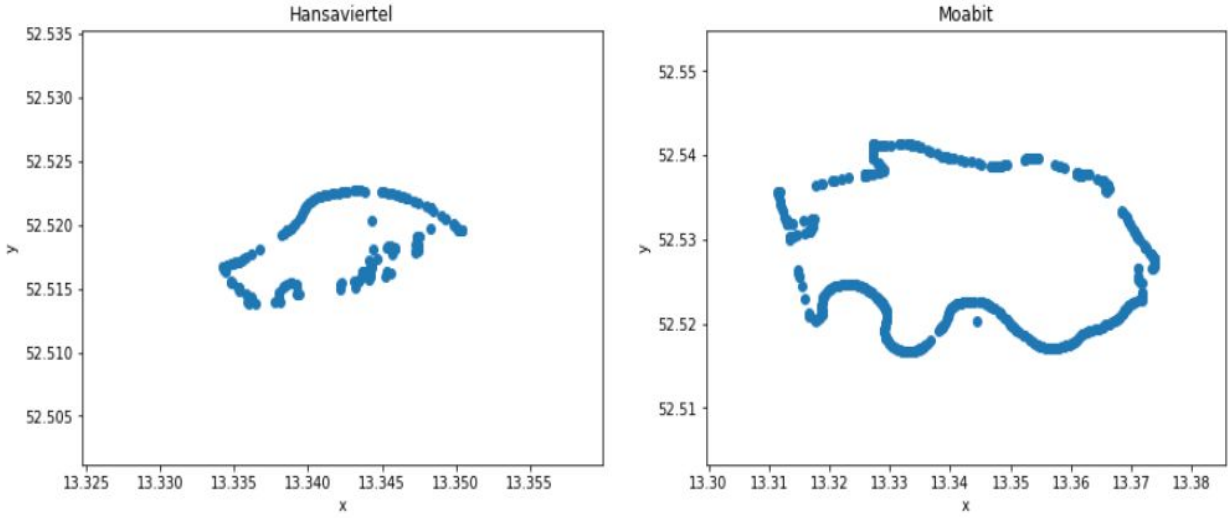
	Neighborhood density	Tourism metric	Latitude	Longitude
count	96.000000	96.000000	96.000000	96.000000
mean	4697.302083	10.017518	52.512402	13.402081
std	4120.801471	15.564494	0.066364	0.132896
min	144.000000	1.000000	52.369611	13.137632
25%	1687.750000	2.755367	52.457439	13.308584
50%	3301.500000	4.276291	52.510233	13.398779
75%	5830.250000	6.165038	52.570168	13.494899
max	16204.000000	63.106762	52.647514	13.708946

So the only neighborhoods with density bigger that 3300 were considered, 48 neighborhoods were filtered out.

3.4 Venues Data

For each neighborhood a call was made to Foursquare to acquire venues in **radius of 1000 meters**, with limit of 100 venues per call.
Venue names, location and category were used.
Important things that had to be considered as mentioned before:

since the acquired venues are at a max radius from a certain coordinate, it could be that **2 coverage circles** from different neighborhoods are **overlapping**, this will result in getting the same venue twice or even more.




```
Hansaviertel: Correct Neighborhood
Moabit: Wrong Neighborhood
```

Venues that were in the wrong neighborhood were reassigned to the correct one::

correct_or_wrong	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude
Wrong Neighborhood	Moabit	52.529050	13.342629	Lir Irish Pub	52.520284	13.344341
Wrong Neighborhood	Moabit	52.529050	13.342629	Konditorei & Café Buchwald	52.521197	13.347825
Wrong Neighborhood	Hansaviertel	52.518229	13.342319	Siegessäule	52.514487	13.350116
Wrong Neighborhood	Hansaviertel	52.518229	13.342319	Schloss Bellevue	52.517455	13.352745
Wrong Neighborhood	Hansaviertel	52.518229	13.342319	Teehaus im Englischen Garten	52.516998	13.347868

The venue were considered a **duplicate** when they had the **same coordinates** (venue category is in also in this dataframe to the right)

and those that are in a neighborhood that was filtered out were discarded:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude
Hansaviertel	52.529050	13.342629	Lir Irish Pub	52.520284	13.344341
Hansaviertel	52.529050	13.342629	Konditorei & Café Buchwald	52.521197	13.347825
outsider	52.518229	13.342319	Siegessäule	52.514487	13.350116
outsider	52.518229	13.342319	Schloss Bellevue	52.517455	13.352745
outsider	52.518229	13.342319	Teehaus im Englischen Garten	52.516998	13.347868

the n the venues were counted in each neighborhood and business density metric was calculated in the same way:

Business_density_metric = venues_count / min(all_venues_count)

Neighborhood	Borough	Neighborhood density	Tourism metric	Latitude	Longitude	Venues count	Business density metric
Mitte	Mitte	7445	63.106762	52.522220	13.397631	100	25.00
Moabit	Mitte	8993	63.106762	52.529050	13.342629	112	28.00
Hansaviertel	Mitte	11111	63.106762	52.518229	13.342319	12	3.00
Wedding	Mitte	8273	63.106762	52.550058	13.338227	49	12.25
Gesundbrunnen	Mitte	13496	63.106762	52.550437	13.384709	93	23.25
Friedrichshain	Friedrichshain-Kreuzberg	11662	18.642645	52.508536	13.455603	111	27.75
Kreuzberg	Friedrichshain-Kreuzberg	14184	18.642645	52.496556	13.410784	99	24.75
Prenzlauer Berg	Pankow	12991	6.165038	52.539254	13.434476	105	26.25
Weißensee	Pankow	5736	6.165038	52.552953	13.462319	51	12.75
Pankow	Pankow	9868	6.165038	52.567234	13.411147	62	15.50

3.5 Clustering the data

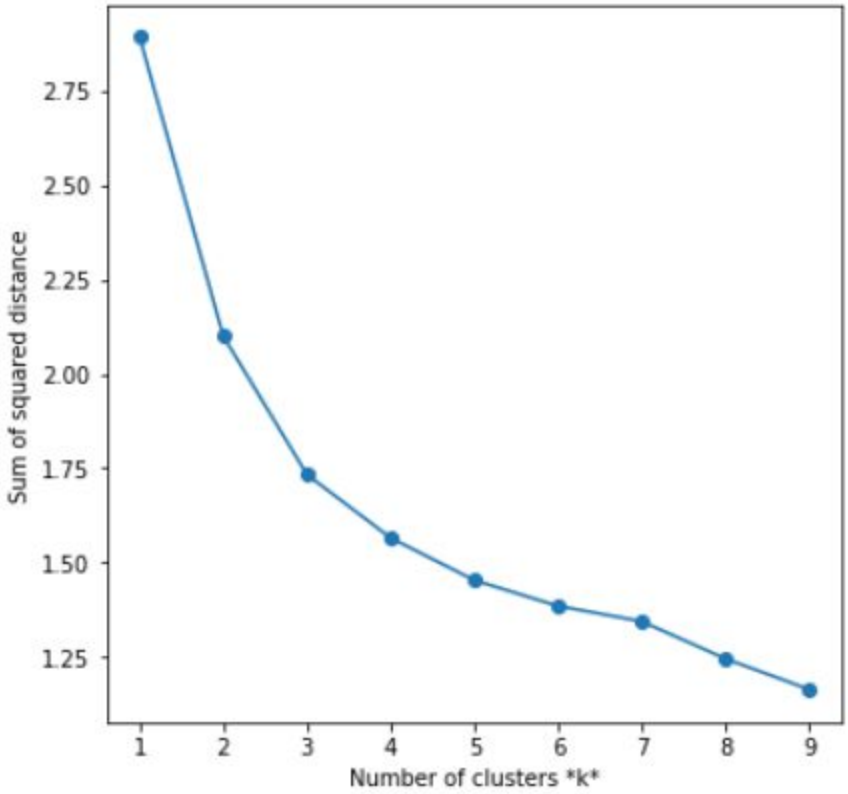
First the data of the venues was processed to onehot encoding and frequencies of different venue categories in each neighborhood was calculated

	Neighborhood	ATM	Adult Boutique	African Restaurant	American Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	Ar Craf Sto
0	Alt-Hohenschönhausen	0.000000	0.0	0.0	0.0	0.0	0.0	0.019608	0
1	Alt-Treptow	0.019608	0.0	0.0	0.0	0.0	0.0	0.000000	0
2	Altglienicke	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0
3	Baumschulenweg	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0
4	Buckow	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0

then the top 10 most common venues in each neighborhood were acquired:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th M Common Ve
0	Alt-Hohenschönhausen	Tram Station	Supermarket	Fast Food Restaurant	Bakery	Coffee S
1	Alt-Treptow	Italian Restaurant	Café	Park	Seafood Restaurant	Beer Gar
2	Altglienicke	Supermarket	Dessert Shop	Flea Market	Flower Shop	Food & D S
3	Baumschulenweg	Supermarket	Drugstore	Garden Center	Café	Vietnam Restau
4	Buckow	Supermarket	Hotel	Chinese Restaurant	Fast Food Restaurant	Bus S

K-means algorithm was used to cluster the data, to choose how many clusters i ran the algorithm for different values of ‘k’ and plotted the Error (SSE) to be able to utilize the **‘elbow method’**:



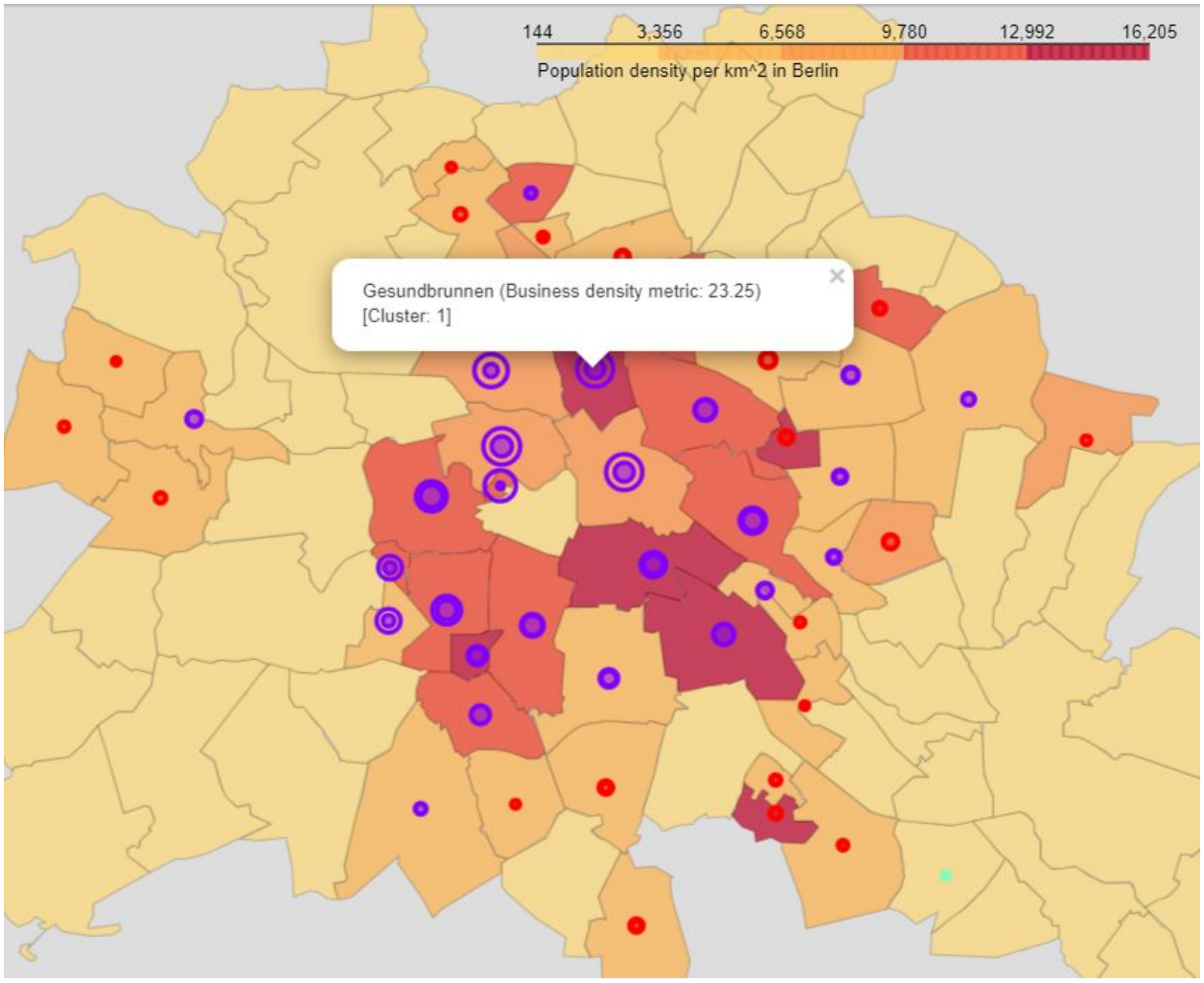
There was no obvious ‘elbow’ and the Sum of Squared distance(Error) (SSE) always drops with increasing values of clusters(k), however **rate of drop**/decrease is **not sharp after k=3**, so I chose k=3 for the model.

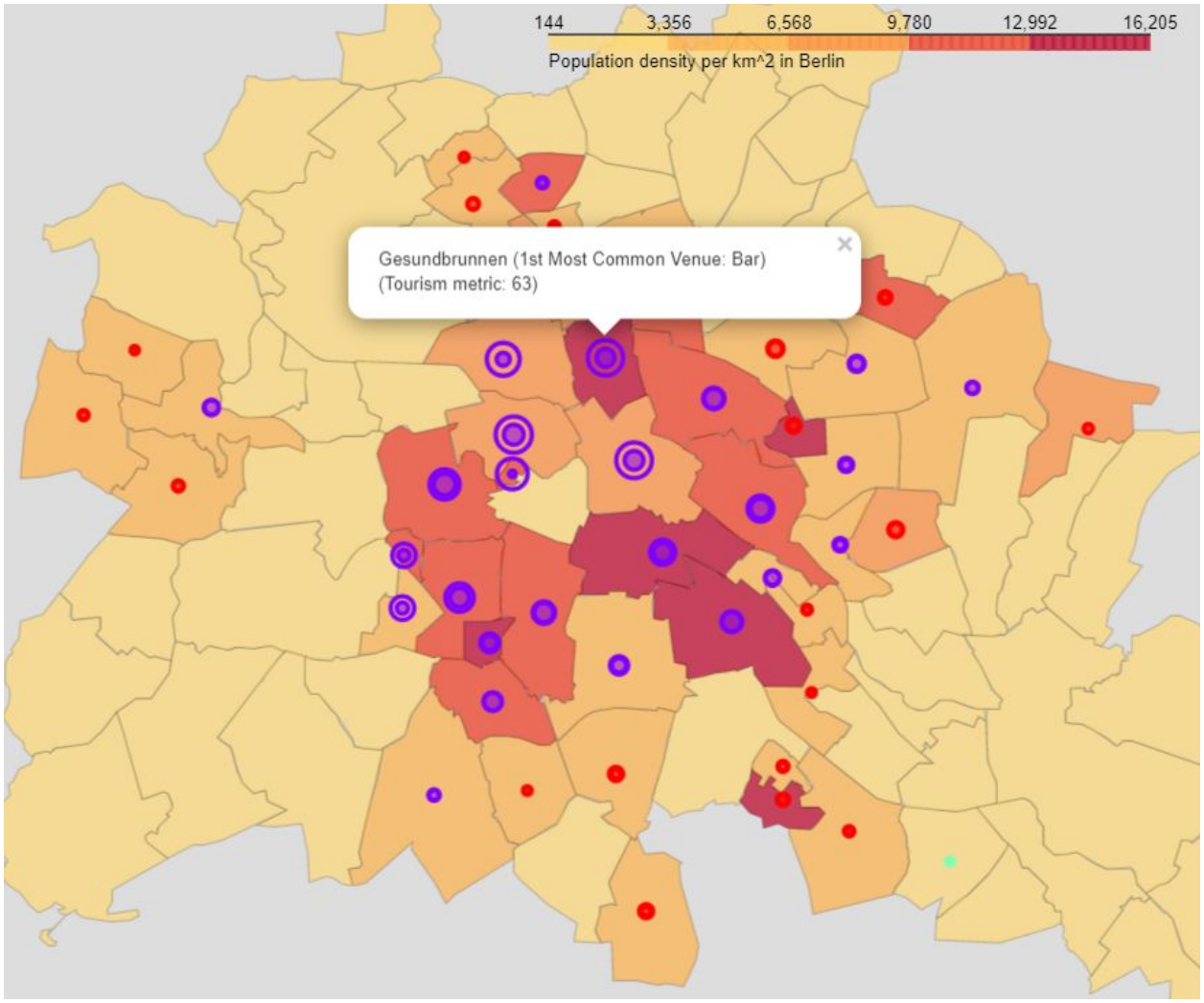
Neighborhood	Borough	Neighborhood density	Tourism metric	Latitude	Longitude	Venues count	Business density metric	Cluster Labels	1st Most Common Venue	2 M Comm Ver
Mitte	Mitte	7445	63.106762	52.522220	13.397631	100	25.00	1	Hotel	Gall
Moabit	Mitte	8993	63.106762	52.529050	13.342629	112	28.00	1	Café	Cof Sh
Hansaviertel	Mitte	11111	63.106762	52.518229	13.342319	12	3.00	1	Art Museum	F
Wedding	Mitte	8273	63.106762	52.550058	13.338227	49	12.25	1	Café	
Gesundbrunnen	Mitte	13496	63.106762	52.550437	13.384709	93	23.25	1	Bar	Bak

4. Results

After clustering the a choropleth map was created with the following features:

1. the **inner circle** area represents the **Business density metric** in neighborhood - scaled by 5 for better visualization
   Radius = sqrt(5*Business_density_metric/pi)
2. the **outer circle** area represents the **Tourism metric** in the corresponding borough added to the Business density metric (click on outer) - scaled by 5 for better visualization
   Radius = sqrt((5*Tourism_metric+5*Business_density_metric) / pi)
3. the **color of marker** represents the **cluster**
4. **Color of the map** represents the **population density**
5. **Inner circle popup** will show **Neighborhood, Business density metric and Cluster number**
6. **Outer circle popup** will show **Neighborhood, 1st Most Common Venue and Tourism metric**





So the map above can convey a lot of information visually for the stakeholders to make a decision, for example the **larger area difference** between the inner and outer circles indicated **low business density with respect to tourism**, and if the **map tile is dark** as well that means more population and **more demand**.
Or **1st most common venue** category in the neighborhood could mean a **high demand** on this certain type of venues but also **high competition**.

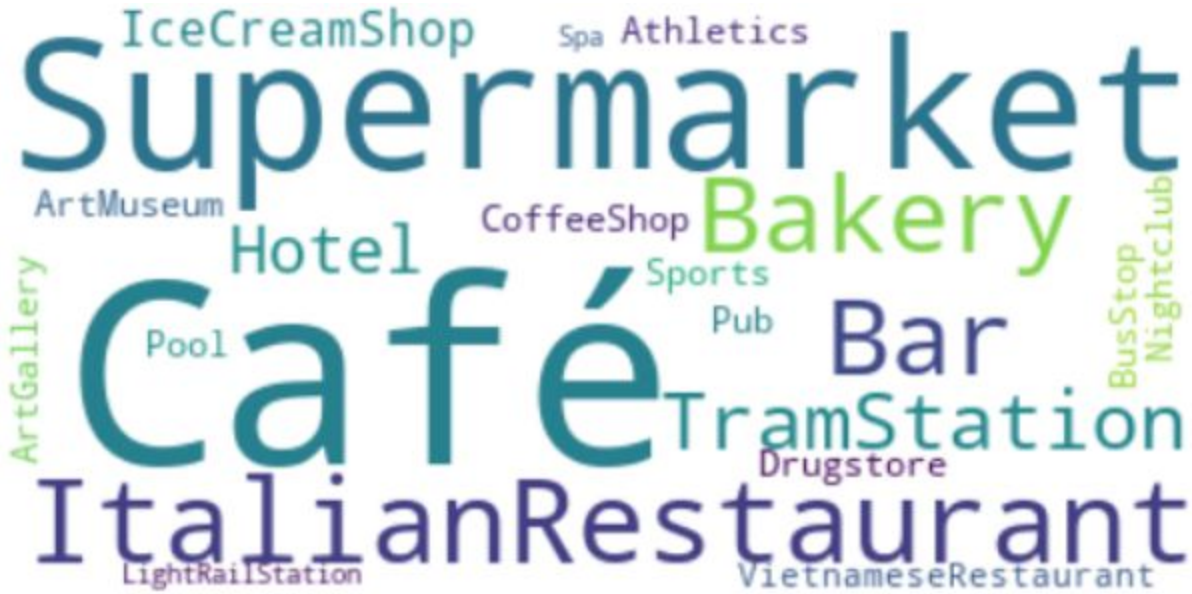
There are so many other ways to correlate the variables plotted above, like with respect to geographical distance from each other or the fact that purple cluster is in the middle and the red one around.

Also to see what is the **most common venue in** a certain **cluster**(not neighborhood) i used **Wordcould**, with the option of how many columns to choose 1st ----to-- 10th most common.
Cluster 1 (Purple):

Cluster 0 (Red):

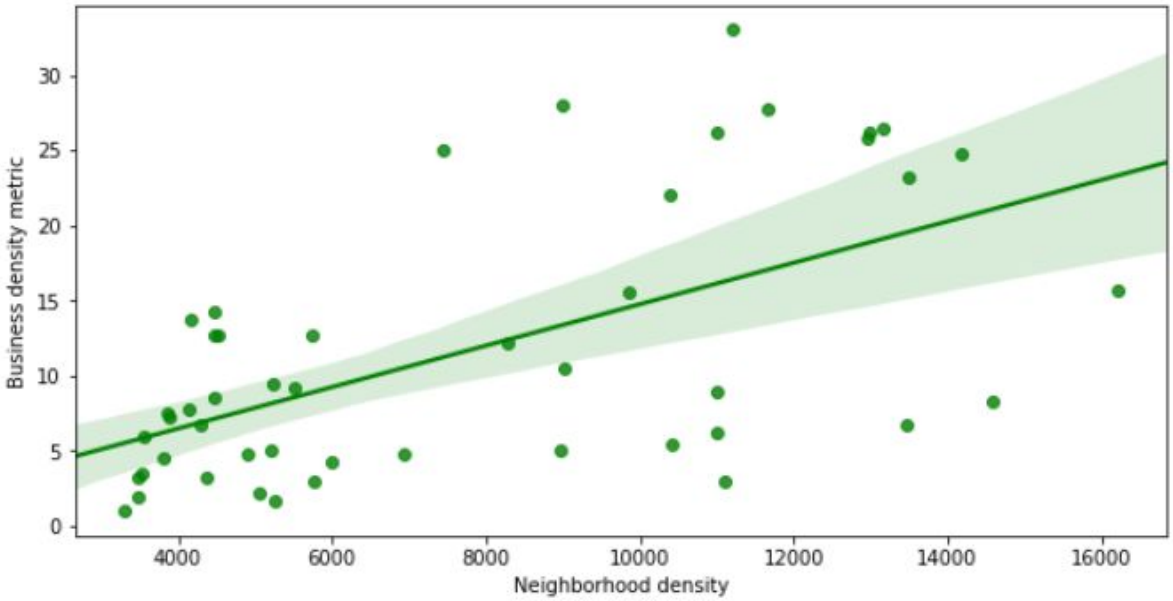
One conclusion we can draw is that in the inner city there are more cafes and in outer less dense/touristic parts there are more drugstors, however there are many supermarkets in both clusters.

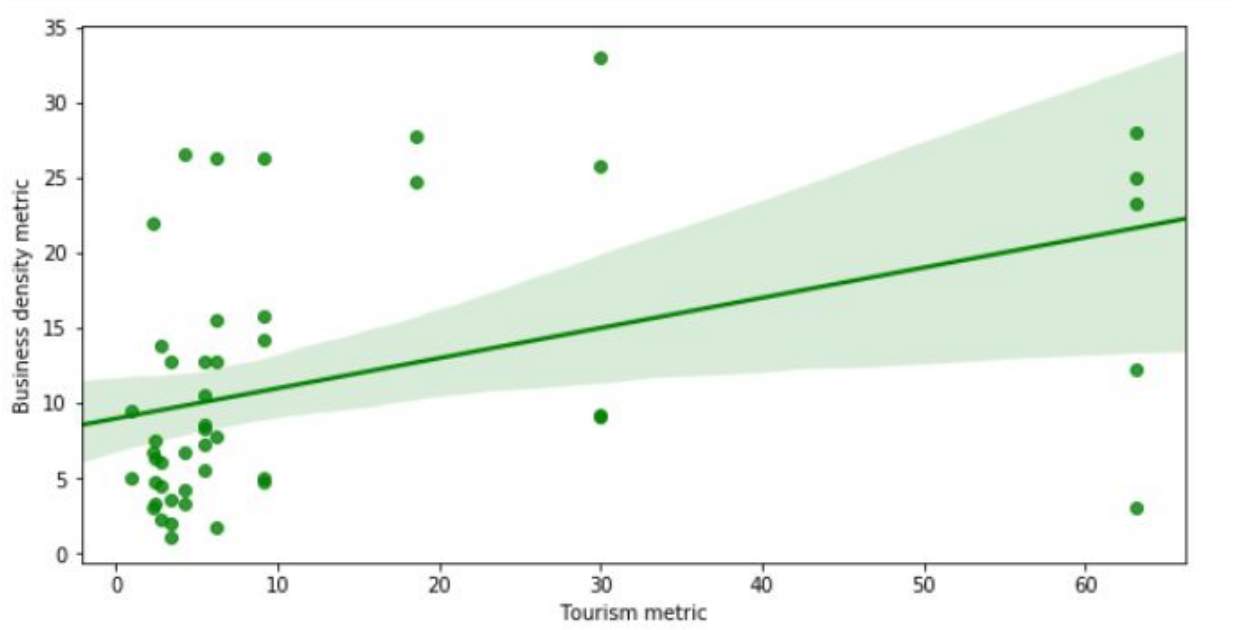
We also can reduce the number of considered columns to redact the less common venues and see the more common ones clearer.
Cluster 1 (Purple) 3 columns only:



We can see that bars are fairly common, possibly high demand/competition

And finally a look at regression plots could give some insight, to further understand how strong the relationship is between variables:





One can notice that the relationship between business density - neighborhood density is stronger than between business density - tourism, with that in mind a stakeholder could give more weight to the population density than tourism when deciding.

4. Discussion

As mentioned in the previous section there are so many ways to interpret the data, and that is better done with the stakeholders, as for the data itself it could be more comprehensive, the same research could be done with more data on tourism, businesses and even more variables could be integrated for better assessment.
Only half of the city was considered but if more data to be provided, one can consider the whole city and see if any patterns will emerge.
It is also worth mentioning that a temporal data could be so impactful if considered, however this all relates to the desire of the stakeholders and their interests.

5. Conclusion

Things that could be considered by business entrepreneurs:

1. where the 'color' is darker - neighborhood density high - high demand

2. where outer circle area bigger than the inner circle area - low business density with respect to tourism - less competition and possible high demand.

3. where outer circle area close to the inner circle area - high business density with respect to tourism - more competition and less demand.

4. options of the business category are words shown by the word cloud (depends on the cluster)
big words: more competition but possible high demand.
small words: less competition but possible low demand.

- for housing/real estate business it's the opposite of 1 and 2 and the more diverse the Wordcloud is the better.

To a prosperous Future,
Azeer Esmail