Capstone Project Report - The Battle of Neighborhoods

Nicklas Norrick
February 12th, 2019

Abstract

This report contains a detailed description of the steps involved in completion of the Capstone Project - The Battle of Neighborhoods of the Applied Data Science Capstone course by IBM. The introduction will explain the problem and business case. In the next section, we will describe the data to be used in detail. The methods used to solve the problem will be explained and results will be presented. For three realistic test cases based on a dataset for the top 20 european cities, we discuss the results and finally draw our conclusions. Depending on the quality of the input data, the list of proposed venues is more or less specifically suited as an answer to our problem. For the future, more data to help determine neighborhood similarity, for example socioeconomic data, population average age, ethnicity, mean income, etc. might prove useful.

Contents

Chapter 1

Introduction/Business Problem

Imagine you are the manager of a band and in the process of planning a tour. You know roughly where in the world you want to play (North America, Europe, capital cities around the world, etc.), but don't know which venues would suit the band you represent the best.

You could of course start writing to or calling all the possible venues in a certain city. This can be very time-consuming and does not necessarily lead to good results. Wouldn't it be great if you could name a reference venue, for example in the home base town or neighborhood of the band, and find venues in similar neighborhoods in other cities?

What characteristics define a reference venue? There are many answers to this question, but we want to put ourselves in the position of the band manager. Turnout and revenue are the measureable quantities which will finally decide whether the venue choices were correct. Cultural match of a neighborhood to a certain style of music is a soft factor which cannot be quantified as easily, but we want to try to use location data to shed light on this.

Assuming that similar neighborhoods will be home to people with similar socioeconomic backgrounds but also similar lifestyles and tastes, this would greatly facilitate finding promising venues. In addition, it can be assumed that the turnout for certain types concerts (i.e. musical genres) will be higher.

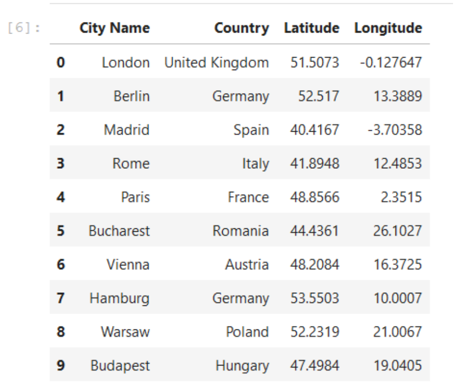
Overall, the tool we want to develop will save time and effort on the part of the band manager organizing a tour, as well as increasing concert turnout and thereby revenue for the band.

To achieve this goal, we will use Foursquare location data to classify neighborhoods in different cities and find similar neighborhoods across whole countries or the whole world. In these similar neighborhoods, concert venues will then be recommended to the user.

Chapter 2

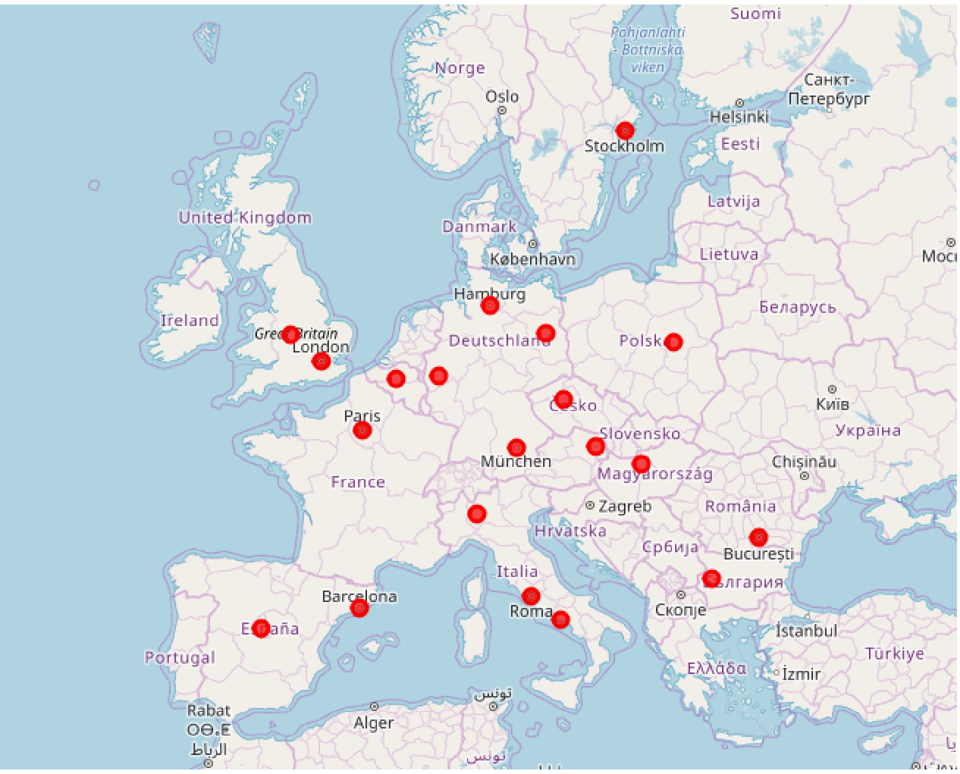
Data

In this project, we will rely mainly on Foursquare location data. We may need to scrape auxiliary data from the web, for example if we want a list of major cities in North America or Europe to loop over. For example, we could use the information on *https://en.wikipedia.org/wiki/List_of_cities_in_the_European_Union_by_population_within_city_limits* to give us the top 20 cities in the European Union by population. We can extract the data from the table and get a list of the top cities and the corresponding country. We can access *geocode* data to get the latitude and longitude of each city. For our European Union dataset the first 10 rows of the resulting dataframe could look like this:



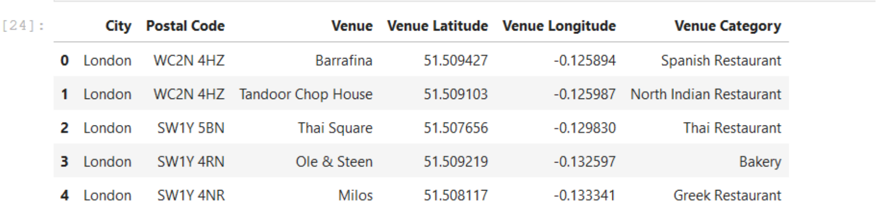
	City Name	Country	Latitude	Longitude
0	London	United Kingdom	51.5073	-0.127647
1	Berlin	Germany	52.517	13.3889
2	Madrid	Spain	40.4167	-3.70358
3	Rome	Italy	41.8948	12.4853
4	Paris	France	48.8566	2.3515
5	Bucharest	Romania	44.4361	26.1027
6	Vienna	Austria	48.2084	16.3725
7	Hamburg	Germany	53.5503	10.0007
8	Warsaw	Poland	52.2319	21.0067
9	Budapest	Hungary	47.4984	19.0405

To visualize this data and check its plausibility, we can use the *folium* library to plot a map.



The next step will be to use Foursquare location data to cluster the neighborhoods of these cities. Using the Foursquare API we will use the **explore** function to request venues in the cities. We will use the postal code data to distinguish neighborhoods from each other. In urban areas, postal codes are assigned to comparatively small areas, so we can argue that all of the addresses belonging to a city postal code are a "neighborhood".

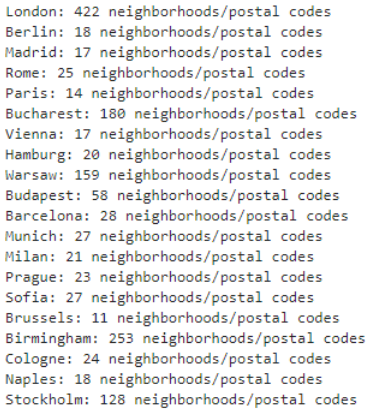
If the postal code data is not given for a certain venue in the Foursquare data, we will not load the venue into our database. To get more rows of data, we could write a script to automatically look up postal codes using the known part of the address, but a first look at the data showed that for most cities, the postal code is given for over 80% of the venues. So in the following, we will use the column name "Postal Code" as an alias for neighborhood. The first rows of the resulting dataframe look like this:



[24]:

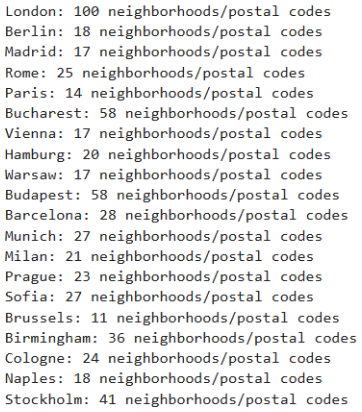
	City	Postal Code	Venue	Venue Latitude	Venue Longitude	Venue Category
0	London	WC2N 4HZ	Barrafina	51.509427	-0.125894	Spanish Restaurant
1	London	WC2N 4HZ	Tandoor Chop House	51.509103	-0.125987	North Indian Restaurant
2	London	SW1Y 5BN	Thai Square	51.507656	-0.129830	Thai Restaurant
3	London	SW1Y 4RN	Ole & Steen	51.509219	-0.132597	Bakery
4	London	SW1Y 4NR	Milos	51.508117	-0.133341	Greek Restaurant

The postal codes in certain countries, for example the UK, refer to rather small areas on the map, which leads to a multitude of "neighborhoods" with only one venue. We will not be able to cluster these sensibly. The distribution is shown in the following screen shot:



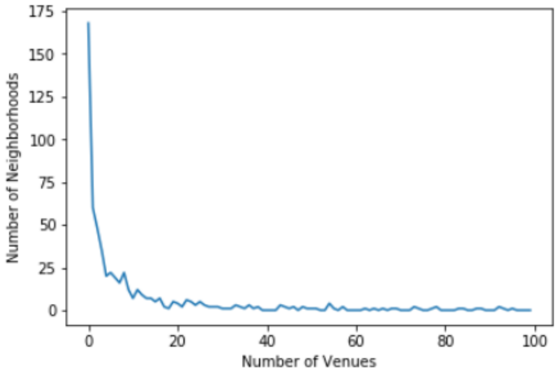
```
London: 422 neighborhoods/postal codes
Berlin: 18 neighborhoods/postal codes
Madrid: 17 neighborhoods/postal codes
Rome: 25 neighborhoods/postal codes
Paris: 14 neighborhoods/postal codes
Bucharest: 180 neighborhoods/postal codes
Vienna: 17 neighborhoods/postal codes
Hamburg: 20 neighborhoods/postal codes
Warsaw: 159 neighborhoods/postal codes
Budapest: 58 neighborhoods/postal codes
Barcelona: 28 neighborhoods/postal codes
Munich: 27 neighborhoods/postal codes
Milan: 21 neighborhoods/postal codes
Prague: 23 neighborhoods/postal codes
Sofia: 27 neighborhoods/postal codes
Brussels: 11 neighborhoods/postal codes
Birmingham: 253 neighborhoods/postal codes
Cologne: 24 neighborhoods/postal codes
Naples: 18 neighborhoods/postal codes
Stockholm: 128 neighborhoods/postal codes
```

Because of this, for Birmingham, London, Warsaw, Stockholm and Bucharest we drop the final 2 digits of the postal code. This reduces the number of neighborhoods to the following:



```
London: 100 neighborhoods/postal codes
Berlin: 18 neighborhoods/postal codes
Madrid: 17 neighborhoods/postal codes
Rome: 25 neighborhoods/postal codes
Paris: 14 neighborhoods/postal codes
Bucharest: 58 neighborhoods/postal codes
Vienna: 17 neighborhoods/postal codes
Hamburg: 20 neighborhoods/postal codes
Warsaw: 17 neighborhoods/postal codes
Budapest: 58 neighborhoods/postal codes
Barcelona: 28 neighborhoods/postal codes
Munich: 27 neighborhoods/postal codes
Milan: 21 neighborhoods/postal codes
Prague: 23 neighborhoods/postal codes
Sofia: 27 neighborhoods/postal codes
Brussels: 11 neighborhoods/postal codes
Birmingham: 36 neighborhoods/postal codes
Cologne: 24 neighborhoods/postal codes
Naples: 18 neighborhoods/postal codes
Stockholm: 41 neighborhoods/postal codes
```

For our test case, the query of data from Foursquare results in a dataframe with 1472 postal codes/neighborhoods and 8414 venues. After shortening the postal codes, a total of 583 neighborhoods remain. Exploring the dataset, we see that a large number of these neighborhoods only contain one or two venues. This is visualized in the following plot.



In our next data cleaning step, we get rid of postal codes with only 1 or 2 venues, since we will not be able to do any sensible analysis with these postal codes/neighborhoods. We can also argue that these are neighborhoods where not a lot is going on, so they won't be promising locations for a concert anyway.

The final cleaning step results in 8126 venues in 355 neighborhoods. There are 289 unique venue categories in our test data set.

We will get the most common venue categories in each neighborhood, and then use this feature to group the neighborhoods into clusters. The frequency of venue types around the neighborhood centerpoints is an indirect measure of neighborhood similarity.

Finally, we will need a subset of our dataframe containing only concert locations. This will be the starting point for our search of similarly situated concert locations in other cities. In our particular case, this venue categories chosen are:

- Rock Club
- Concert Hall
- Piano Bar
- Irish Pub
- Jazz Club
- Performing Arts Venue
- Theater
- Karaoke Bar
- Music Venue
- Opera House

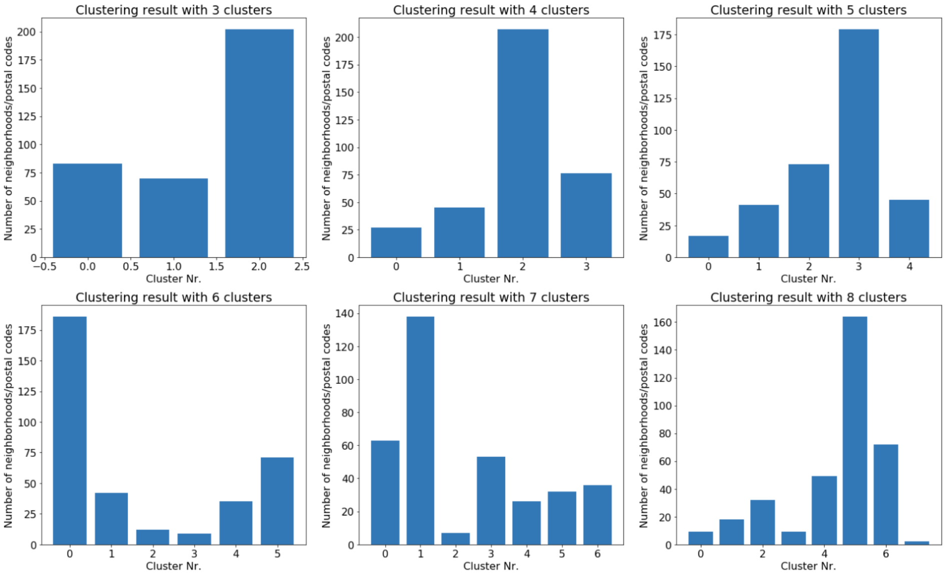
- Amphitheater
- Salsa Club
- Country Dance Club
- Cultural Center

These categories are based on experience and may not be complete. As data science is an iterative process, future iterations of our algorithm may include further categories or exclude categories.

Chapter 3

Methodology

The neighborhoods are clustered using the k-means clustering algorithm, since it is simple but effective in many cases. Input data is the one-hot encoded dataframe of all the venues. The k-means fit is carried out using the standard metrics. To determine a sensible value for the number of clusters, we test values from 3 to 8 and look at the distribution of the neighborhoods in the clusters. We want to achieve a good partioning of the neighborhoods, but not too many small clusters, since these won't contain enough potential venues in the resulting data.



We chose 5 clusters, since this seems to be a good trade-off between being able to differentiate different neighborhoods while at the same time not resulting in clusters with so few neighborhoods that sensible propositions cannot be made.

Once the neighborhoods are clustered we can look at the subsets to see if the results are plausible. If so, the next step is simple: We choose a reference venue (a concert location where we know our band has a lot of fans and enjoys a good turnout), take the cluster number and search for music venues from the list above with the same cluster number. This list is the final output for our band manager.

Chapter 4

Results

For our test case, we will choose 3 reference locations in different cities and see how our algorithm performs and if the results are plausible.

First, we will look the the clustering results for k-means clustering with 5 clusters. From the top ten venues in each cluster, we can name the clusters to get a feeling for the type of neighborhood we have in each cluster.

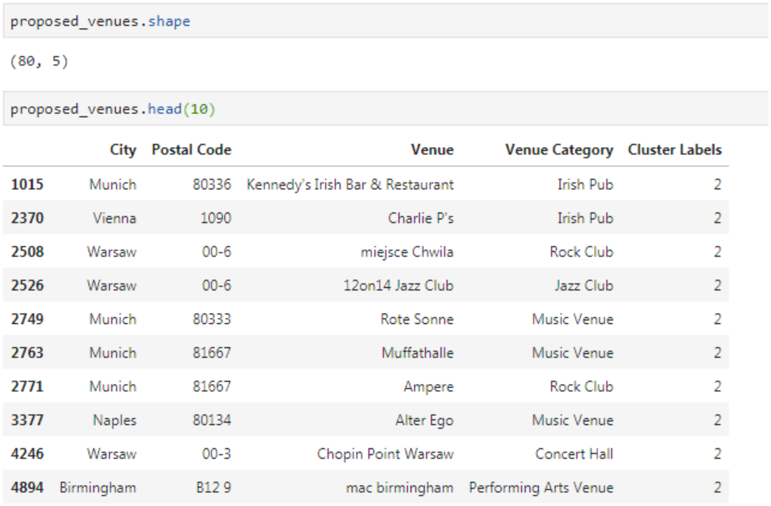
Cluster label:

0: Markets and food shopping
1: Theater
2: Melting pot
3: Worldwide cuisine
4: Coffee & Wine

Each of the three reference locations belongs to a different cluster:

Case 1: Reference location "Ampere" in Munich, Germany

The "Ampere" is a venue of the category "rock club" and the corresponding neighborhood belongs to cluster 2. The first ten rows of the list of 80 proposed venues is as follows:



```
proposed_venues.shape
```

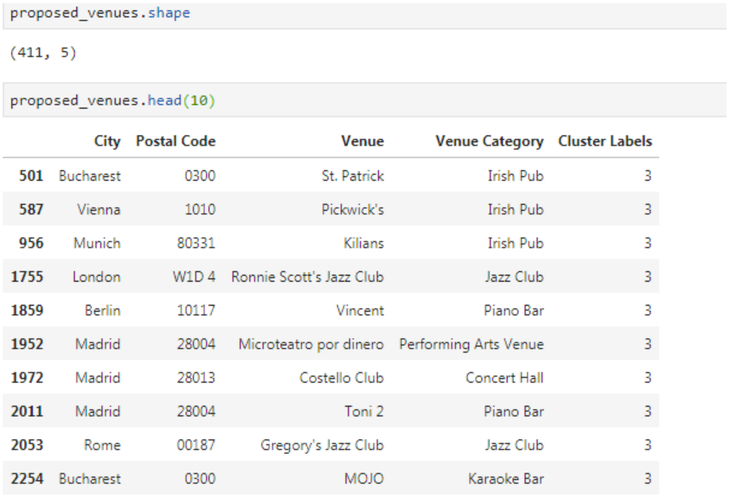
```
(80, 5)
```

```
proposed_venues.head(10)
```

	City	Postal Code	Venue	Venue Category	Cluster Labels
1015	Munich	80336	Kennedy's Irish Bar & Restaurant	Irish Pub	2
2370	Vienna	1090	Charlie P's	Irish Pub	2
2508	Warsaw	00-6	miejsce Chwila	Rock Club	2
2526	Warsaw	00-6	12on14 Jazz Club	Jazz Club	2
2749	Munich	80333	Rote Sonne	Music Venue	2
2763	Munich	81667	Muffathalle	Music Venue	2
2771	Munich	81667	Ampere	Rock Club	2
3377	Naples	80134	Alter Ego	Music Venue	2
4246	Warsaw	00-3	Chopin Point Warsaw	Concert Hall	2
4894	Birmingham	B12 9	mac birmingham	Performing Arts Venue	2

Case 2: Reference location "Costello Club" in Madrid, Spain

The "Costello Club" is a venue of the category "concert hall" and was clustered to neighborhood type 3. This is one of the larger clusters, which results in a very large number (411) of suggested venues. The top of the list looks like this:



```
proposed_venues.shape
```

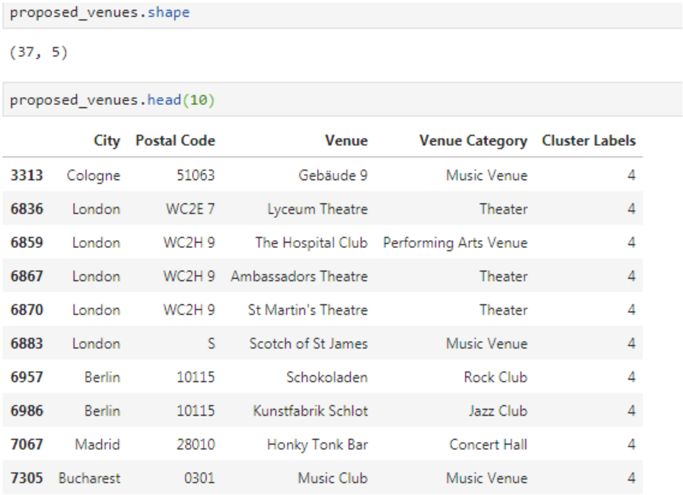
```
(411, 5)
```

```
proposed_venues.head(10)
```

	City	Postal Code	Venue	Venue Category	Cluster Labels
501	Bucharest	0300	St. Patrick	Irish Pub	3
587	Vienna	1010	Pickwick's	Irish Pub	3
956	Munich	80331	Kilians	Irish Pub	3
1755	London	W1D 4	Ronnie Scott's Jazz Club	Jazz Club	3
1859	Berlin	10117	Vincent	Piano Bar	3
1952	Madrid	28004	Microteatro por dinero	Performing Arts Venue	3
1972	Madrid	28013	Costello Club	Concert Hall	3
2011	Madrid	28004	Toni 2	Piano Bar	3
2053	Rome	00187	Gregory's Jazz Club	Jazz Club	3
2254	Bucharest	0300	MOJO	Karaoke Bar	3

Case 3: Reference location "VK Concerts" in Brussels, Belgium

The "VK Concerts" is a venue of the category "music venue" and clustered in neighborhood type 4. There are 37 proposed venues, the beginning of the list is as follows:



```
proposed_venues.shape
```

```
(37, 5)
```

```
proposed_venues.head(10)
```

	City	Postal Code	Venue	Venue Category	Cluster Labels
3313	Cologne	51063	Gebäude 9	Music Venue	4
6836	London	WC2E 7	Lyceum Theatre	Theater	4
6859	London	WC2H 9	The Hospital Club	Performing Arts Venue	4
6867	London	WC2H 9	Ambassadors Theatre	Theater	4
6870	London	WC2H 9	St Martin's Theatre	Theater	4
6883	London	S	Scotch of St James	Music Venue	4
6957	Berlin	10115	Schokoladen	Rock Club	4
6986	Berlin	10115	Kunstfabrik Schlot	Jazz Club	4
7067	Madrid	28010	Honky Tonk Bar	Concert Hall	4
7305	Bucharest	0301	Music Club	Music Venue	4

Chapter 5

Discussion

The number of suggested venues depends greatly upon the size of the clusters and therefore upon the quality of the data input into the clustering algorithm. For test case 2, it is questionable whether the list of over 400 proposed venues will be of any help. For test cases 1 and 3, the length of the list is more useful.

It might be helpful to filter the resulting venues according to category if the number of results is too high. For example, if the reference venue is of the category "rock club", experience might tell us that "jazz club" or "piano bar" are less promising due to the venue type, even if the neighborhood information is similar to that of the reference venue.

Chapter 6

Conclusions

Our problem was to develop a tool to help a band manager save time and effort in organizing a tour, as well as increasing concert turnout and thereby revenue for the band. We have shown how the use of Foursquare location data, cleaned up and processed using a clustering algorithm, can help to sort possible venues in many cities according to the similarity of their respective neighborhoods.

Depending on the quality of the data, the list of proposed venues is more or less specifically suited to answer our question. In the future, it would help to add more specific data to our dataframe to determine neighborhood similarity, for example socioeconomic data, population average age, ethnicity, mean income, etc.