Business Problem

Imagine there is a **travel provider**, who has a special offer in his portfolio: Customers can book a trip to New York and attend a home game of the New York Yankees. So, the customers can enjoy a Yankee match and in the rest of the time (usually a few days) they are free to explore New York City.

The travel provider has access to various lodgings in different neighborhoods. It is important that the Yankees stadium is easy to reach from the lodging with regard to the high traffic volume before Yankees matches. From his experience, he knows that the boroughs Manhattan and Bronx are optimal in this particular case.
Now, the provider doesn't want to just randomly pick neighborhoods in Manhattan or Bronx in order to prevent customer dissatisfaction. At the same time he doesn't want to present customers a long list of neighborhoods, making them choose without any information about the neighborhoods.

The provider has an idea: He hires a data scientist and gives him the task to cluster the neighborhoods in Manhattan and Bronx with regard to similar venues in a specific radius of each neighborhood. In this way, he can **propose a specific set of neighborhoods to his customers based on their personal preferences**. As soon as the customer chooses a cluster, the provider can check which lodgings are available in the neighborhoods of that cluster. The audience of this project is clearly the travel provider, who wants to improve customer experience with the help of data science.

Data

For this project, the JSON-file *nyu_2451_34572-geojson.json* is used which can be downloaded via https://ibm.box.com/shared/static/fbpwbovar7lf8p5sgddm06cgipa2rxpe.json . It comprises all boroughs and neighborhoods in New York together with their geometric coordinates. The relevant information is stored in a dataframe and all neighborhoods that are not located in Bronx or Manhattan are removed.

Additionally, Foursquare location data is used to obtain information about the venues in the vicinity of each neighborhood. A request URL looks for example like this: https://api.foursquare.com/v2/venues/explore?&client_id=xxx&client_secret=xxx&v=20180605&ll=40.894705,-73.847201&radius=500&limit=100 (client_id and client_secret have been anonymized) and provides information like this:


```
[{'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4c537892fd2ea593cb077a28',
   'name': 'Lollipops Gelato',
   'location': {'address': '4120 Baychester Ave',
    'crossStreet': 'Edenwald & Bussing Ave',
    'lat': 40.894123150205274,
    'lng': -73.84589162362325,
    'labeledLatLngs': [{'label': 'display',
      'lat': 40.894123150205274,
     'lng': -73.84589162362325}],
   'distance': 127,
   'postalCode': '10466',
   'cc': 'US',
   'city': 'Bronx',
   'state': 'NY',
   'country': 'United States',
   'formattedAddress': ['4120 Baychester Ave (Edenwald & Bussing Ave)',
    'Bronx, NY 10466',
    'United States']},
  'categories': [{'id': '4bf58dd8d48988d1d0941735',
    'name': 'Dessert Shop',
    'pluralName': 'Dessert Shops',
    'shortName': 'Desserts',
    'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/des
sert_',
     'suffix': '.png'},
    'primary': True}],
  'photos': {'count': 0, 'groups': []}},
 'referralId': 'e-0-4c537892fd2ea593cb077a28-0'}]
```


We will be interested in the information that is stored under 'categories' 'name', in this case *Dessert Shop*.

Data Exploration

In the beginning, the data from the JSON-file is loaded and stored into a pandas dataframe. All neighborhoods that do not belong to the boroughs Bronx or Manhattan are removed. The dataframe looks as follows:

	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

Figure 1: Neighborhoods of Bronx and Manhattan and their location values, retrieved from the JSON-file.

The neighborhoods can be nicely visualized with a folium map:

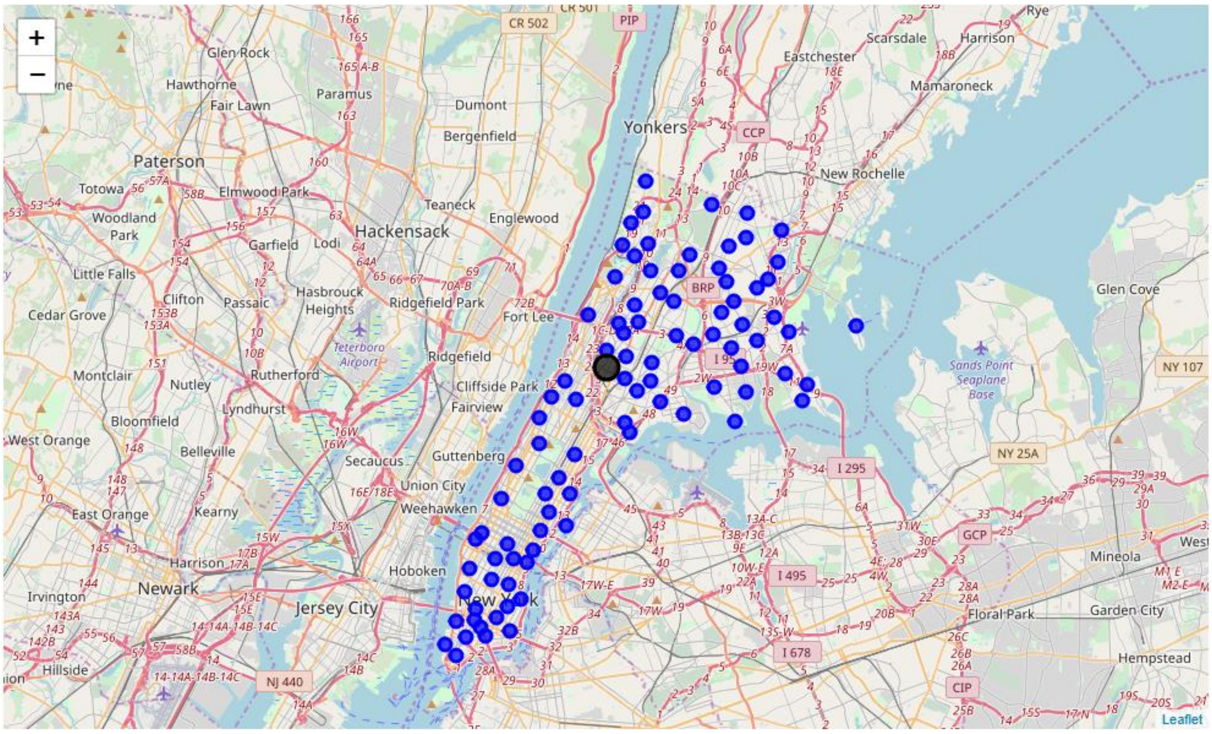

Figure 2: Part of New York as a folium map. The neighborhoods of Bronx and Manhattan are marked with blue circles. The larger black circle indicates the location of the Yankee Stadium

Now, the Foursquare location data is used to obtain information about venues. A function is defined to obtain the top 100 venues within a radius of 500m of each neighborhood. The data is stored in a pandas dataframe and looks as follows:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Wakefield	40.894705	-73.847201	Lollipops Gelato	40.894123	-73.845892	Dessert Shop
1	Wakefield	40.894705	-73.847201	Rite Aid	40.896521	-73.844680	Pharmacy
2	Wakefield	40.894705	-73.847201	Carvel Ice Cream	40.890487	-73.848568	Ice Cream Shop
3	Wakefield	40.894705	-73.847201	Dunkin Donuts	40.890631	-73.849027	Donut Shop
4	Wakefield	40.894705	-73.847201	SUBWAY	40.890656	-73.849192	Sandwich Place

Figure 3: Dataframe of all neighborhoods and their respective top 100 venues within a radius of 500m.

In an overview, we can check how many venues were found for each neighborhood. A small part of the overview is shown here:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Allerton	29	29	29	29	29	29
Battery Park City	100	100	100	100	100	100
Baychester	20	20	20	20	20	20
Bedford Park	36	36	36	36	36	36
Belmont	95	95	95	95	95	95
Bronxdale	16	16	16	16	16	16
Carnegie Hill	100	100	100	100	100	100
Castle Hill	8	8	8	8	8	8
Central Harlem	43	43	43	43	43	43
Chelsea	100	100	100	100	100	100

Figure 4: Dataframe that shows how many venues were returned for each neighborhood.

In Castle Hill for example only 8 venues were found within 500m, whereas in Chelsea the defined limit of 100 was fully exhausted.

The data is then manipulated in a way that for each neighborhood, the frequency of occurrence of each venue category is provided. Again, a small part shall be shown for clarification:

	Neighborhood	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	Airport Tram	American Restaurant	Animal Shelter	Antiqu Sho
0	Allerton	0.000000	0.00	0.00	0.000000	0.000000	0.034483	0.00	0.(
1	Battery Park City	0.000000	0.00	0.00	0.000000	0.000000	0.010000	0.00	0.(
2	Baychester	0.000000	0.00	0.00	0.000000	0.000000	0.100000	0.00	0.(
3	Bedford Park	0.000000	0.00	0.00	0.000000	0.000000	0.000000	0.00	0.(
4	Belmont	0.000000	0.00	0.00	0.000000	0.000000	0.010526	0.00	0.(
5	Bronxdale	0.000000	0.00	0.00	0.000000	0.000000	0.000000	0.00	0.(
6	Carnegie Hill	0.000000	0.00	0.00	0.000000	0.000000	0.010000	0.00	0.(
7	Castle Hill	0.000000	0.00	0.00	0.000000	0.000000	0.000000	0.00	0.(

Figure 5: Dataframe where all venue categories are represented by columns and the entries state the frequency of their occurrence in the different neighborhoods.

These statistics are used to finally create a dataframe, where each neighborhood is listed among with its 10 most common venues within 500m:

	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	Allerton	Pizza Place	Supermarket	Deli / Bodega	Chinese Restaurant	Spa	Department Store	Pharmacy	Breakfast Spot	Bus Station	Spanish Restaurant
1	Battery Park City	Coffee Shop	Park	Hotel	Italian Restaurant	Wine Shop	Gym	Women's Store	Memorial Site	Shopping Mall	Sandwich Place
2	Baychester	American Restaurant	Supermarket	Pet Store	Spanish Restaurant	Mexican Restaurant	Fast Food Restaurant	Men's Store	Mattress Store	Electronics Store	Baseball Field
3	Bedford Park	Diner	Fried Chicken Joint	Deli / Bodega	Mexican Restaurant	Chinese Restaurant	Supermarket	Pizza Place	Sandwich Place	Pharmacy	Train Station
4	Belmont	Italian Restaurant	Pizza Place	Deli / Bodega	Bakery	Grocery Store	Dessert Shop	Liquor Store	Mediterranean Restaurant	Mexican Restaurant	Sandwich Place
5	Bronxdale	Italian	School	Supermarket	Breakfast	Spanish	Mexican	Paper / Office Supplies	Bank	Chinese	Eastern European

Figure 6: Dataframe where the 10 most frequent venue categories are given for each neighborhood.

Modeling

The travel provider wants to find similar neighborhoods with regard to their nearby venues. Hence, it is an (unsupervised) clustering task. The *k*-means algorithm is predestined for this case.
The algorithm is applied on the dataframe in Figure 5, dropping the “Neighborhood” column. Since the values in this dataframe represent frequencies, they are already of same scale and there is no need to normalize them.

Choosing an appropriate *k*-value is always a challenging task. We try different values between 1 and 10 and compute the distortion as the sum of the Euclidean distances between each point of a cluster and its respective center. This is referred to as “elbow method” because plotting the distortion against *k* often results in a curve that is shaped like an elbow. The kink of the elbow then represents the optimal value for *k*. In this case the plot looks as follows:

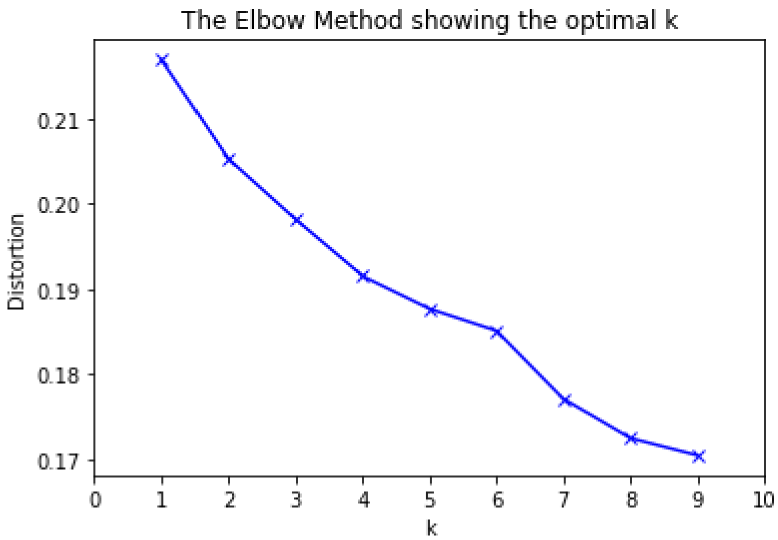

Figure 7: Distortion of clusters calculated for different k-values.

Unfortunately, the curve doesn’t follow the shape of an elbow and therefore it is not possible to determine a *k*-value via this method. However, trying out different values turns out that choosing *k*>3 results in at least one cluster that has only one neighborhood. Since such a cluster is pointless for our business problem, so we choose *k*=3.

Results

After the algorithm has finished running, the clustered neighborhoods are again visualized with a folium map:

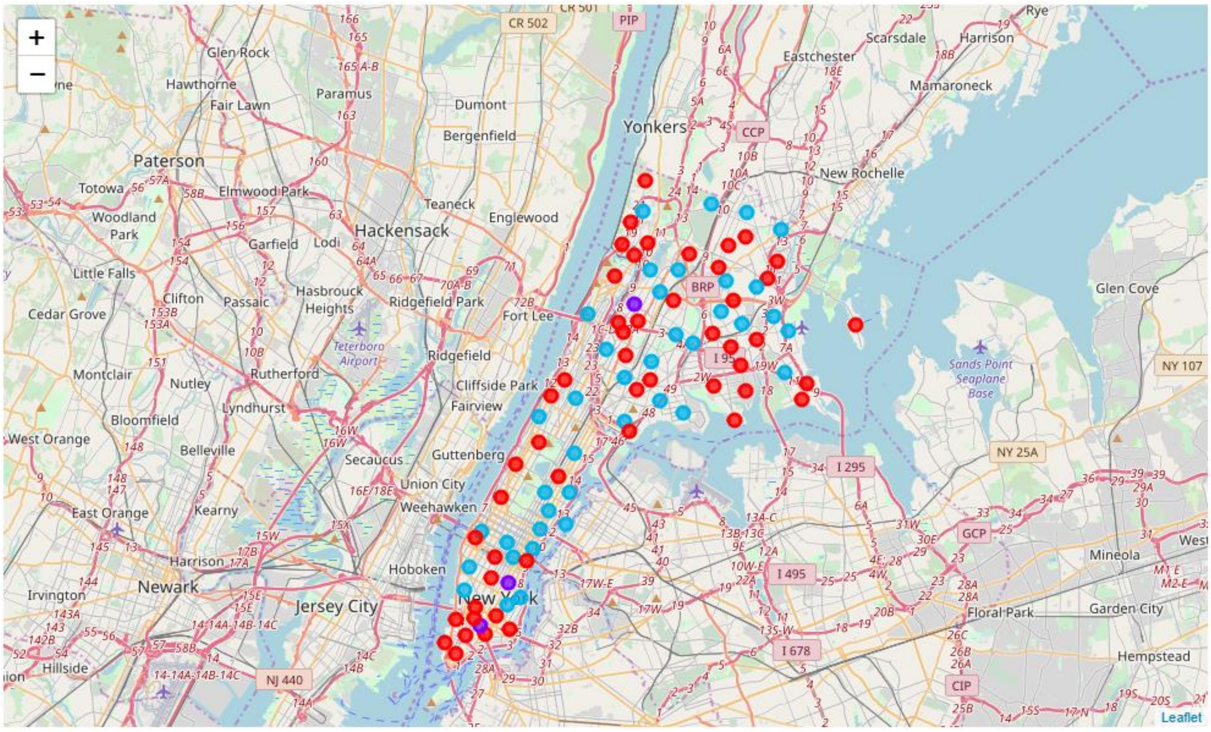

Figure 8: Neighborhoods divided into 3 clusters by the k-means algorithm. Purple = cluster 0, blue = cluster 1, red = cluster 2.

The clusters obviously contain different numbers of neighborhoods:

- Cluster 0: 03 neighborhoods
- Cluster 1: 39 neighborhoods
- Cluster 2: 50 neighborhoods

For the travel provider it is important to know what makes up the different clusters. Therefore, the five most frequent venue categories and their relative occurrence are determined for each cluster:

- Cluster 0:

Rank	Venue category	Relative occurrence
1	Pizza Place	6.7 %
2	Sandwich Place	6.7 %
3	Chinese Restaurant	6.7 %
4	Cocktail Bar	6.7 %
5	Shoe Store	3.3 %

- Cluster 1:

Rank	Venue category	Relative occurrence
1	Deli / Bodega [1]	4.6 %
2	Pizza Place	4.4 %
3	Sandwich Place	3.6 %
4	Italian Restaurant	3.6 %
5	Coffee Shop	3.3 %

- Cluster 2:

Rank	Venue category	Relative occurrence
1	Pizza Place	4.8 %
2	Italian Restaurant	4.2 %
3	Coffee Shop	4.0 %
4	Park	3.0 %
5	Grocery Store	2.8 %

[1] A „bodega“ is a little corner store that acts like a supermarket and also as a neighborhood hangout spot. They are often open 24/7. For more information see for example https://streeteasy.com/blog/what-is-a-bodega/

Discussion

The three resulting clusters have some top venues in common, for example they all have a pizza place on rank 1 or 2. This is due to the fact that pizza places are very popular in New York. Cluster 1 and cluster 2 have Italian restaurants in their top 5, whereas Cluster 0 has Chinese restaurants instead. So, potential customers can choose here which cuisine they favor. Another possible argument for cluster 0 could be the cocktail bars which will probably have more importance for younger customers. In contrast to that, the other two clusters offer coffee shops. Customers choosing cluster 1 or 2 can provide themselves with food and drinks from bodegas or grocery stores, respectively. Cluster 0 does not provide such an opportunity in its top 5 venues and is hence more suitable for customers who want to eat and drink outside of their lodging (this also goes well with the cocktail bars). Finally, cluster 2 also has parks in its top 5 and is great for customers who like to hang out there or take a walk.

Conclusion

The travel provider wanted to improve his business by clustering neighborhoods with similar venues in order to better satisfy his customers' preferences. The location data of the NY neighborhoods were loaded into a dataframe from a publicly available JSON-file. Then, for each neighborhood the top 100 venues within a radius of 500m were requested via the Foursquare API. With the help of the *k*-means algorithm, the neighborhoods were divided into three clusters based on the frequency of the various venues in their vicinity. It became clear that the travel provider can now offer his customers lodgings in neighborhoods that suit their personal preferences.