image source: https://www.juni.studio/project/interior-design-restaurant-in-berlin/

# Ideal Locations for New Restaurant Business in Berlin

(Prepared By: Sancak Özdemir)

20.04.2020

**Summary**

This project aims at serving recommendations for ideal geographical locations for a new restaurant business in Berlin. A restaurant business is a very prospective and profitable business in general, particularly in big cities such as Berlin. One of the most important parts of this business is to choose a well-suited place where a restaurant will be established. This project focuses on recommending a model for profitable places for restaurant business based on analyzing open data especially Foursquare API data and other public internet resources.

## Table of Contents

1. Introduction
2. Data gathering and preliminary analyses
3. Methodology
   - 3.1. Explore the Neighborhoods of Berlin
   - 3.2. Analyze and Cluster Neighborhoods
     - 3.2.1. Clustering Results
   - 3.3. Analyze and Make Prediction Models
     - 3.3.1. Multiple Linear regression model results
     - 3.3.2. Ridge regression model results
4. Discussion and Conclusion

## 1. Introduction

This project aims at serving recommendations for ideal geographical locations for a new restaurant business in Berlin. A restaurant business is a very prospective and profitable business in general, particularly big cities such as Berlin. One of the most important parts of this business is to choose a well-suited place where a restaurant will be established. This project is focusing on recommending a model for profitable places for this type of business based on analyzing open data especially Foursquare API data and other internet resources.

Berlin is the capital and largest city of Germany by both area and population. Its 3,769,495 (2019) inhabitants make it the most populous and crowded city of the EU. The city is one of Germany's 16 federal states. Its economy is based on high-tech firms and the service sector, encompassing a diverse range of creative industries, research facilities, media corporations and convention venues. Berlin serves as a continental hub for air and rail traffic and has a highly complex public transportation network. In addition, Berlin is hosting many universities, museums, movie theaters, and diverse historical and cultural places. From this perspective, a new restaurant business is a real option for this city with this target population group.

Before starting to analyze, we should determine the data which we need for this business. One of the challenges in data analysis is to define the required data which can represent the problem adequately and this should be done with the project partners. As business understanding, let's check the open sources to define the location-based requirements for the restaurant business such as https://fitsmallbusiness.com/choose-a-restaurant-location/, and summarize the required demographic and geographic data for this analysis.

What we need for this analysis is as follows:

1. Neighborhoods (For clustering and classification purposes)
2. General population and age classification
    1. 15-35 (Fast-food)
    2. 25-45 (Bar-Bistro)
    3. 30-50 (Casual dining)
    4. 35-65 (Fine dining)
    5. 65+
    6. Male/female distribution
    7. Ethnicity (or domestic and foreign population)
3. Locations of parking places
4. Locations of stadiums, theaters, transportation hubs, airports, malls, and universities
5. Cuisine type and/or restaurant styles
6. Crime rates

After getting all the required data, our first goal is to classify them according to neighborhoods of Berlin and to make cluster analysis using venue categories. The final target is to be able to define a prediction model to choose or rank the best locations from a given dataset by using machine learning algorithms on the data we obtained from the attributes of previously established restaurants.

## 2. Data gathering and preliminary analyses

Let's start with the neighborhoods of Berlin data in the following link: https://en.wikipedia.org/wiki/Boroughs_and_neighborhoods_of_Berlin for our analysis. Table 1 shows the neighborhoods and geographic coordinates of them. Let's visualize the neighborhoods of Berlin. Let's generate a Folium map and add the neighborhoods layer to be able to see borders of neighborhoods. The colors of neighborhoods are proportional to the populations of each neighborhood. The color scale is on the top right corner of the map. The generated map is shown in Figure 1.

Table 1. Berlin neighborhoods

| | Borough | Neighborhood | Population | Latitude | Longitude |
|---|---|---|---|---|---|
| 0 | Berlin | Charlottenburg-Wilmersdorf | 319.628 | 52.497058 | 13.296490 |
| 1 | Berlin | Friedrichshain-Kreuzberg | 268.225 | 52.501500 | 13.435120 |
| 2 | Berlin | Lichtenberg | 259.881 | 52.514581 | 13.498392 |
| 3 | Berlin | Marzahn-Hellersdorf | 248.264 | 52.539720 | 13.584280 |
| 4 | Berlin | Mitte | 332.919 | 52.516740 | 13.366790 |
| 5 | Berlin | Neukölln | 310.283 | 52.480200 | 13.433640 |
| 6 | Berlin | Pankow | 366.441 | 52.571050 | 13.404970 |
| 7 | Berlin | Reinickendorf | 240.454 | 52.567550 | 13.331650 |
| 8 | Berlin | Spandau | 223.962 | 52.550090 | 13.200356 |
| 9 | Berlin | Steglitz-Zehlendorf | 293.989 | 52.443640 | 13.229080 |
| 10 | Berlin | Tempelhof-Schöneberg | 335.060 | 52.447630 | 13.385350 |
| 11 | Berlin | Treptow-Köpenick | 241.335 | 52.445817 | 13.574580 |

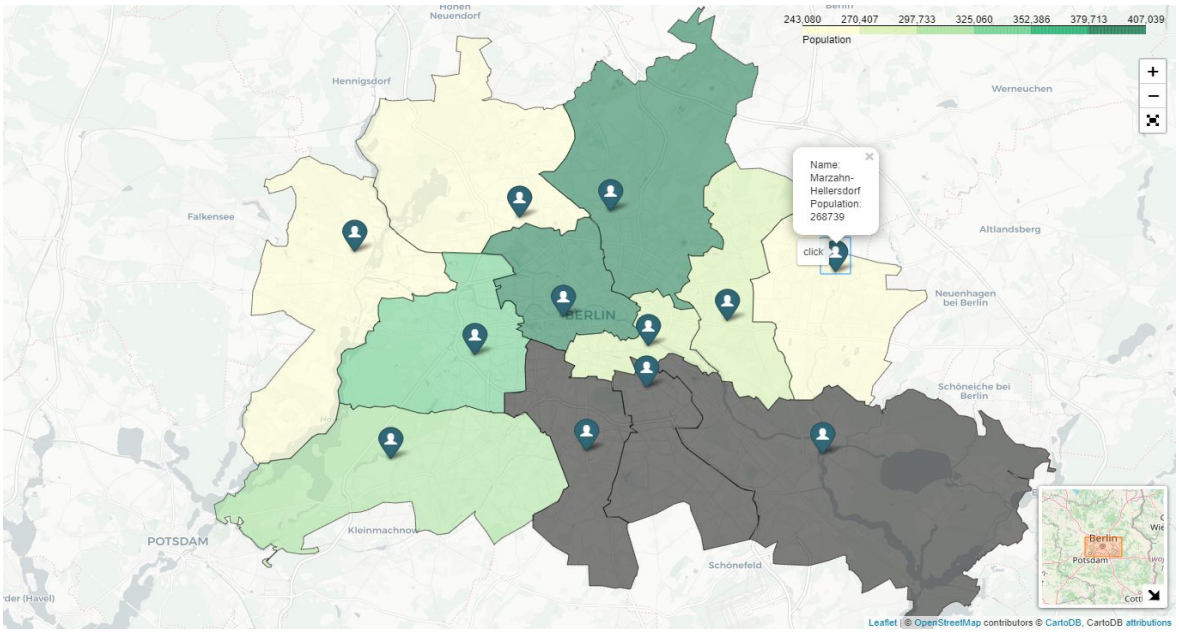



Image 1. Berlin neighborhoods (Color scale shows the population of neighborhoods)

Next, let's continue with the Foursquare data which can be fetched by means of Foursquare API freely. Categories of venues in the API can be found in this link: https://developer.foursquare.com/docs/build-with-foursquare/categories/ .

When we examined the category list, in addition to 'Food':'4d4b7105d754a06374d81259' category, we see that the following venues have valuable information for this project:

'College & University':'4d4b7105d754a06372d81259',
'Shopping Mall':'4bf58dd8d48988d1fd941735',
'Movie Theater':'4bf58dd8d48988d17f941735',
'Stadium':'4bf58dd8d48988d184941735',
'Train Station':'4bf58dd8d48988d129951735',
'Parking':'4c38df4de52ce0d596b336e1',
'Airport':'4bf58dd8d48988d1ed931735'

While using the Foursquare API for this type of analysis 'intend' parameter should be 'browse' or 'global' depending on the purpose.

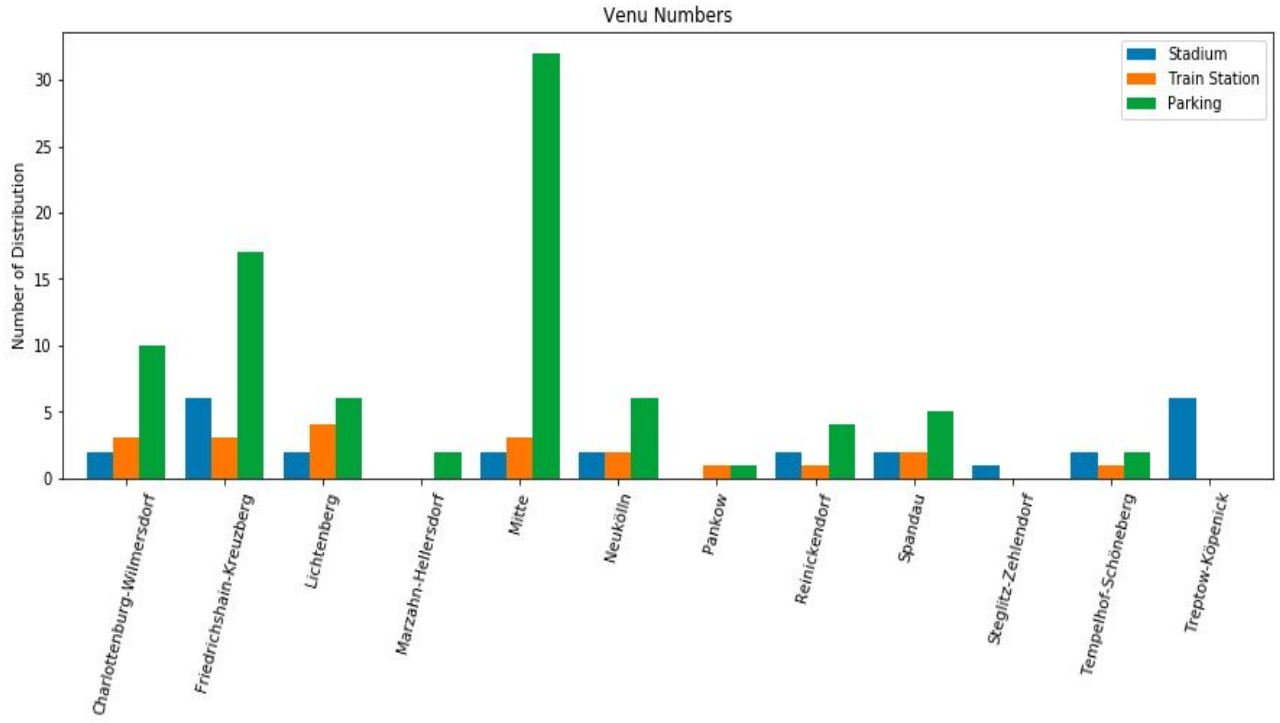
## 3. Methodology

### 3.1. Explore the Neighborhoods of Berlin

We obtained the numbers of universities and colleges, shopping malls, movie theaters, stadiums, train stations and parking places in the neighborhoods of Berlin. Let's take a closer look at these data. When we examine the 'Airport' category, Data are inconsistent for our purpose. They do not show the airports exclusively. Thus, Let's completely skip the 'Airport' category in the analysis.

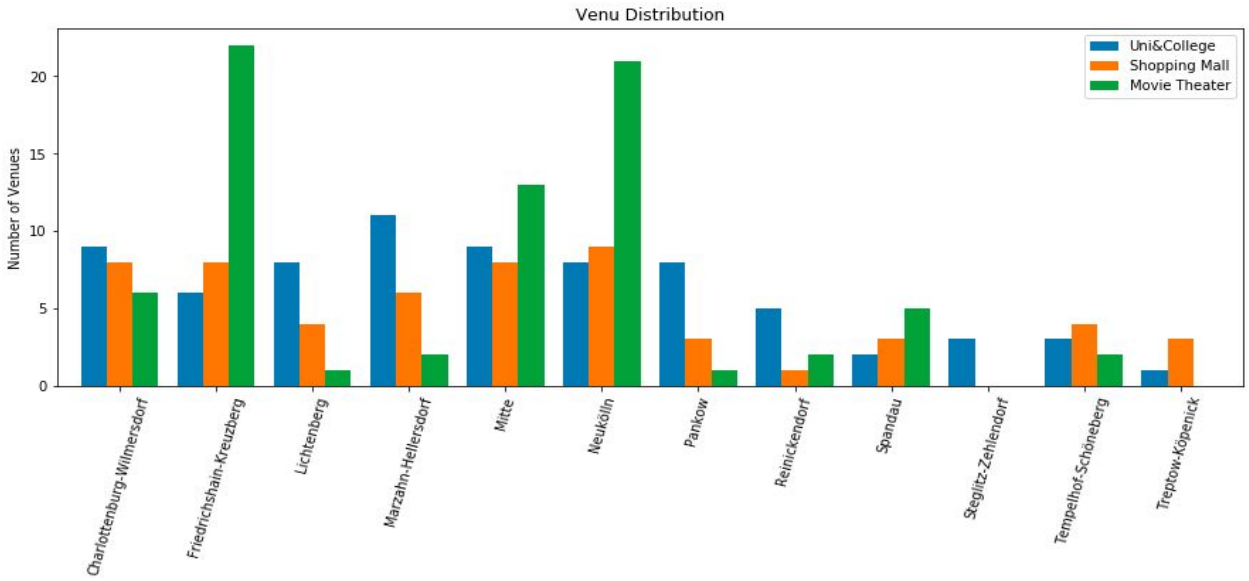
You can see the number of all these important gathering places in Figure 1 and Figure 2.

Figure 1. Number of important gathering places in Berlin
(Number of Stadiums, train stations, parking places)




When we look at Figure 1, the top three in the Parking category are Charlottenburg-Wilmersdorf, Friedrichshain-Kreuzberg and Mitte. This means that these neighborhoods are very crowded central regions of the city.

Figure 2. Number of important gathering places in Berlin
(Uni&Colleges, Shopping Malls, Movie Theaters)




This figure shows us; the top three neighborhoods in *Shopping Mall* and *Movie Theater* categories are *Friedrichshain-Kreuzberg*, *Neukölln and Mitte*. This means that these neighborhoods have much more attraction facilities compared to others and particularly appeals to the younger generation.

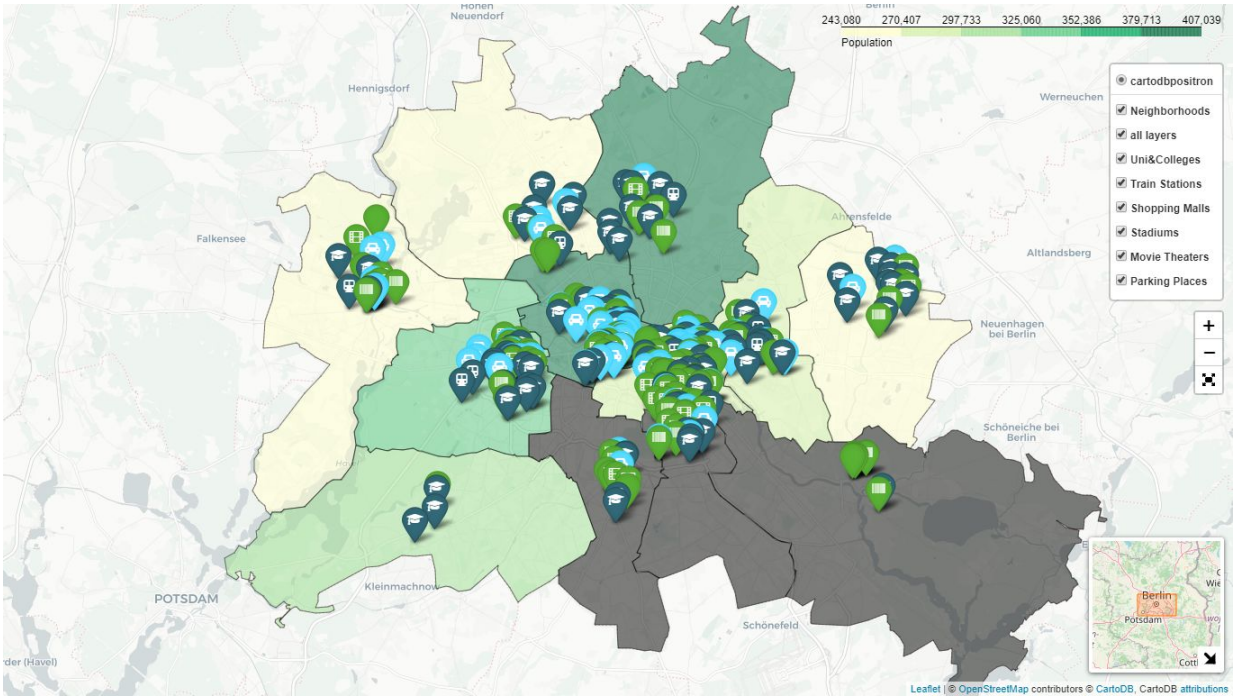



Image 2. People gathering places shown on the Berlin neighborhoods map.

Let us look at Image 2 which shows the people gathering places shown on the Berlin Neighborhoods map. we can see all categories ane by one and examine easily with this interactive map using the top

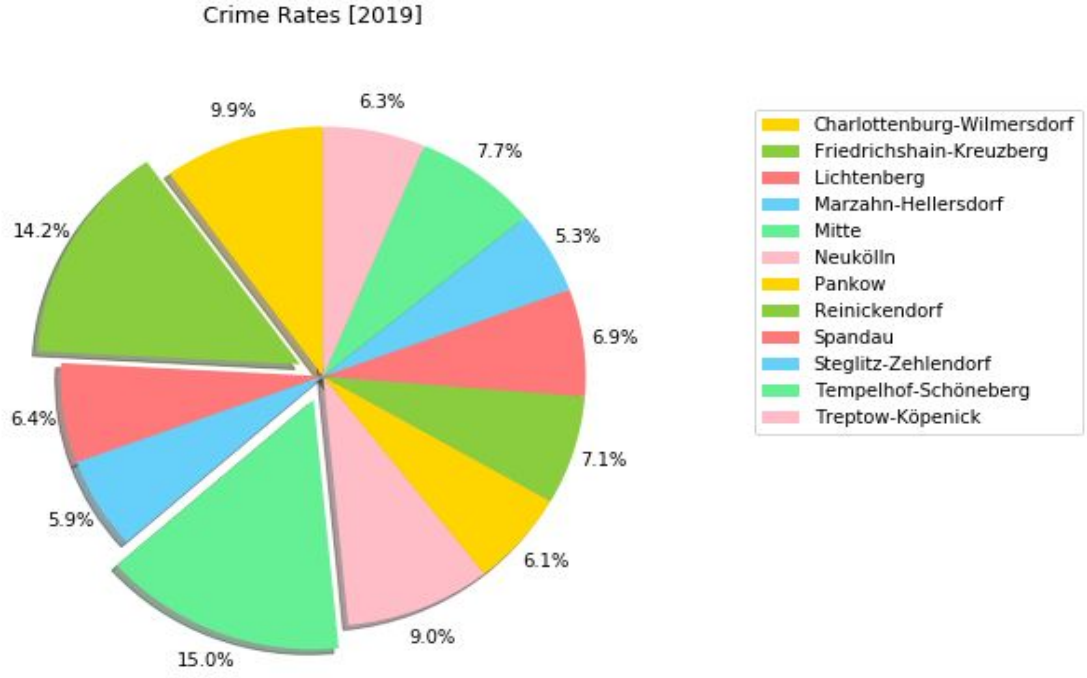
right-hand side widget, . This interactive map is useful when searching for suitable places for the restaurant business to see and understand the distances to highly populated areas of the city. Besides, We can see the details of each venue clicking the corresponding point of interest.

In addition to Foursquare API, we can get the “Crime statistics of Berlin” from the following link: https://www.kriminalitaetsatlas.berlin.de/K-Atlas/bezirke/Fallzahlen&HZ%202012-2019.xlsx

Let's examine the crime data. As we can see data contains the number of events in the neighborhoods of Berlin. On the other hand, it should be normalized according to the population of the corresponding neighborhood. Let's calculate the crime rates by dividing the number of crimes to the population of the corresponding neighborhood. Figure 3 shows the crime rates of the neighborhoods.

As we can see from the pie chart below, Charlottenburg-Wilmersdorf, Friedrichshain-Kreuzberg and Mitte have the highest crime rates. If we combine with the above bar chart, we can say that these neighborhoods are very crowded and central locations in Berlin on the other hand, highest crime rates of these region should be taken into consideration.

Figure 3. Crime rates of Neighborhoods of Berlin




Finally, we can find population-related data which we need from the following link: https://www.statistik-berlin-brandenburg.de/opendata/EWR_Ortsteile_2018-12-31.csv This data named as 'Residents in the districts of Berlin on December 31, 2018' officially and public for research and other purposes.

In addition to these demographic data, we can generate age classification according to age dependent dining types. In open source there are some age classifications such as between 15 and 35 ages prefer fast-food etc. likewise we can do the following classification;

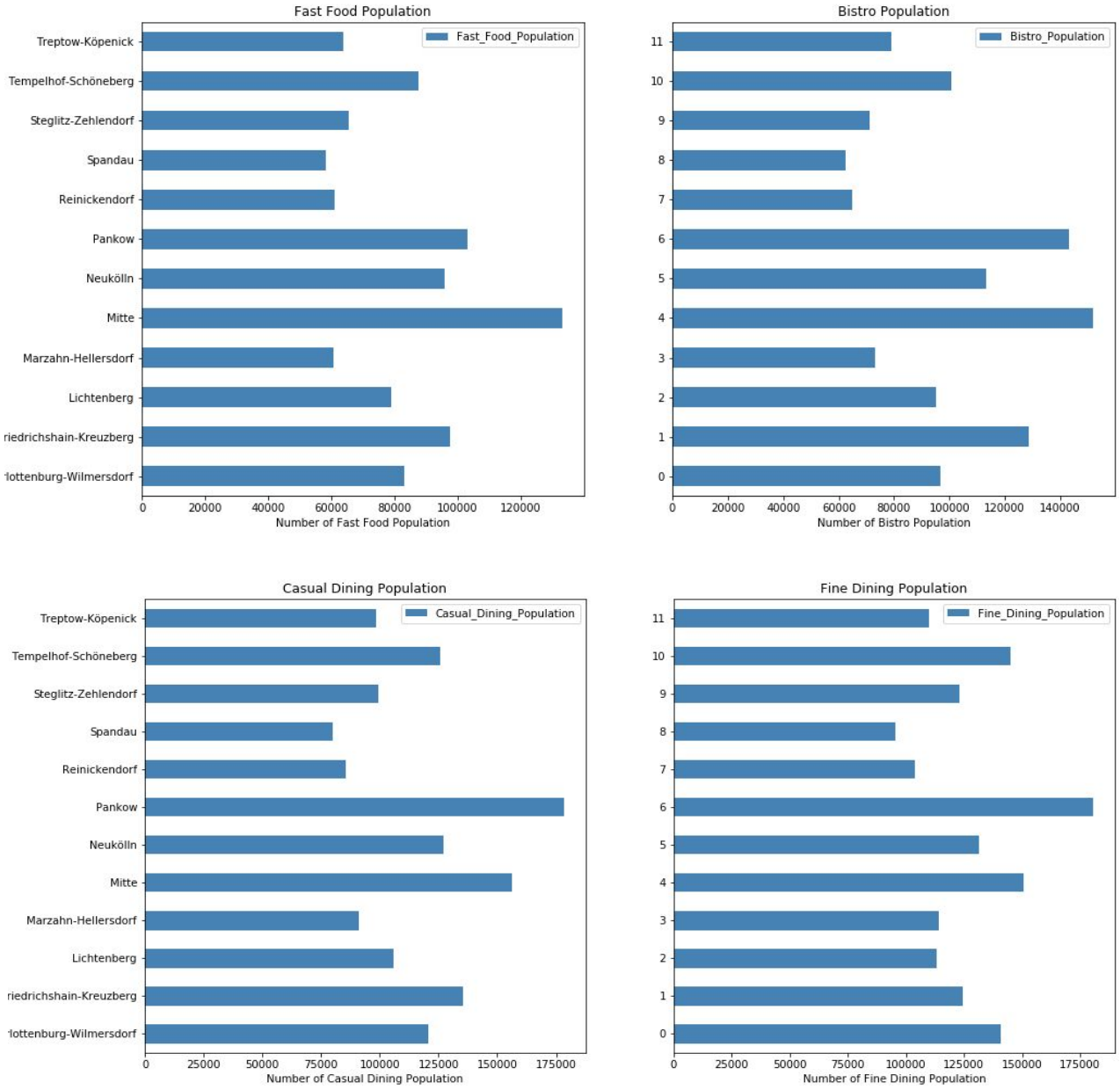
Ages 15-35 (Fast-food)
Ages 25-45 (Bar-Bistro)

Ages 30-50 (Casual dining)
Ages 35-65 (Fine dining)

We can extract these groups of data from 'Residents in the districts of Berlin on December 31, 2018'.

After extracting the required data, let's draw bar charts of these groups versus neighborhoods. Now, we can examine the distribution. Let us look at Figure 4. The first and second sub-figure show as *Mitte, Pankow* and *Friedrichshain-Kreuzberg* are promising locations for *fast food*, *casual dining* and *bistro* type restaurants from a population point of view. Likewise, *Mitte, Pankow* and *Tempelhof-Schönnenberg* are preferable for the *Fine Dining* restaurant business.

Figure 4. Age classification according to dining types in the neighborhoods of Berlin




Now, we can explore the 'Food':'4d4b7105d754a06374d81259' category in detail. After getting all the Food category data of each neighborhood, we should make some cleansing. When we examine the Food category, we see some non-restaurant type venues that should be excluded. We excluded them

from the main table. Then, we finalized the restaurant type of venue list. The list can be summarized as follows:

```
array(['Seafood Restaurant', 'German Restaurant', 'Thai Restaurant',
       'Italian Restaurant', 'Doner Restaurant', 'Gourmet Shop',
       'Snack Place', 'Breakfast Spot', 'Fast Food Restaurant',
       'Japanese Restaurant', 'Chinese Restaurant', 'Steakhouse',
       'Burger Joint', 'Sushi Restaurant', 'French Restaurant',
       'Greek Restaurant', 'Food Court', 'Kofte Place', 'Pizza Place',
       'Falafel Restaurant', 'Middle Eastern Restaurant',
       'Vietnamese Restaurant', 'Gastropub', 'Korean Restaurant',
       'Sandwich Place', 'Lebanese Restaurant', 'Ramen Restaurant',
       'Mexican Restaurant', 'Deli / Bodega', 'Pakistani Restaurant',
       'Restaurant', 'Asian Restaurant', 'Turkish Restaurant',
       'BBQ Joint', 'Indian Restaurant', 'Diner', 'Persian Restaurant',
       'Vegetarian / Vegan Restaurant', 'Bistro', 'Bavarian Restaurant',
       'Currywurst Joint', 'Brasserie', 'Cocktail Bar',
       'Trattoria/Osteria', 'Kebab Restaurant', 'Argentinian Restaurant',
       'Mediterranean Restaurant', 'Brewery', 'Food & Drink Shop',
       'North Indian Restaurant', 'Taverna', 'Food Stand', 'Taco Place',
       'Burrito Place', 'Eastern European Restaurant', 'Hot Dog Joint'],
      dtype=object)
```

Table 2. Unique values of 'Venue Category'

Besides, we can get details of each restaurant venues using 'Venue Id' and Foursquare API. This part of the data is restricted by the data provider. Restrictions apply based on your account type. On the other hand, we can make 500 API calls per day for venue details free of charge and that provides us valuable information such as 'price', 'rating' and 'likes' data for each venue. These data are defined in API reference as follows:

**price:** An object containing the price tier from 1 (least pricey) - 4 (most pricey) and a message describing the price tier.

**rating:** Numerical rating of the venue (0 through 10). Returned as part of an explore result, excluded in search results. Not all venues will have a rating.

**likes:** The count of users who have liked this venue, and groups containing any friends and others who have liked it. The groups included are subject to change.

The complete list of venue details key-value pairs can be found in the following link: https://developer.foursquare.com/docs/api-reference/venues/details

After downloading all venue details and merging into previously gathered data, we finally reached a comprehensive table that contains 31 columns of valuable data for further analysis. These columns are listed below

Table 3. Data columns of Neighborhoods venue table

Data columns (total 31 columns):

| | |
|---|---|
| Neighborhood | 546 non-null object |
| Neighborhood Latitude | 546 non-null float64 |
| Neighborhood Longitude | 546 non-null float64 |
| Venue Id | 546 non-null object |
| Venue | 546 non-null object |
| Venue Latitude | 546 non-null float64 |
| Venue Longitude | 546 non-null float64 |
| Venue Distance | 546 non-null int64 |
| Venue Category | 546 non-null object |
| Price | 546 non-null int64 |
| Rating | 546 non-null float64 |

| | |
|---|---|
| Likes | 546 non-null int64 |
| Borough | 546 non-null object |
| Population | 546 non-null float64 |
| Latitude | 546 non-null float64 |
| Longitude | 546 non-null float64 |
| Uni&College | 546 non-null int64 |
| Shopping Mall | 546 non-null float64 |
| Movie Theater | 546 non-null float64 |
| Stadium | 546 non-null float64 |
| Train Station | 546 non-null float64 |
| Parking | 546 non-null float64 |
| Crime_numbers | 546 non-null int64 |
| Male_Population | 546 non-null int64 |
| Female_Population | 546 non-null int64 |
| Foreign_Population | 546 non-null int64 |
| German_Population | 546 non-null int64 |
| FasFood_Population | 546 non-null int64 |
| Bistro_Population | 546 non-null int64 |
| Casual_Dining_Population | 546 non-null int64 |
| Fine_Dining_Population | 546 non-null int64 |

## 3.2. Analyze and Cluster the Neighborhoods

We have all we need for further analyses. Let's start with clustering the venue categories in Berlin neighborhoods.

After grouping the rows by neighborhood, let’s take the mean of the frequency of occurrence of each category and show each neighborhood along with the top 5 most common venues below.

Table 4. Neighborhoods with  the top 5 most common venues in restaurants category

| | Neighborhood | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|
| 0 | Charlottenburg-Wilmersdorf | Japanese Restaurant | Sushi Restaurant | Doner Restaurant | Snack Place | Chinese Restaurant |
| 1 | Friedrichshain-Kreuzberg | Pizza Place | Vietnamese Restaurant | Falafel Restaurant | Middle Eastern Restaurant | Breakfast Spot |
| 2 | Lichtenberg | Vietnamese Restaurant | Pizza Place | Asian Restaurant | Currywurst Joint | Gastropub |
| 3 | Marzahn-Hellersdorf | Doner Restaurant | Snack Place | Italian Restaurant | Restaurant | Asian Restaurant |
| 4 | Mitte | Fast Food Restaurant | Burger Joint | Vegetarian / Vegan Restaurant | German Restaurant | Sandwich Place |
| 5 | Neukölln | Pizza Place | Breakfast Spot | Korean Restaurant | Middle Eastern Restaurant | Bistro |
| 6 | Pankow | Breakfast Spot | Fast Food Restaurant | Trattoria/Osteria | Italian Restaurant | Burger Joint |
| 7 | Reinickendorf | Fast Food Restaurant | Doner Restaurant | Trattoria/Osteria | Italian Restaurant | Restaurant |
| 8 | Spandau | Turkish Restaurant | Italian Restaurant | Fast Food Restaurant | Doner Restaurant | German Restaurant |
| 9 | Steglitz-Zehlendorf | Italian Restaurant | German Restaurant | Greek Restaurant | Asian Restaurant | Trattoria/Osteria |
| 10 | Tempelhof-Schöneberg | Restaurant | Indian Restaurant | Taverna | German Restaurant | Gastropub |
| 11 | Treptow-Köpenick | German Restaurant | Fast Food Restaurant | Italian Restaurant | Greek Restaurant | Asian Restaurant |

Let's run the k-means clustering model of the scikit-learn machine learning library to cluster the neighborhoods into 5 clusters.

### 3.2.1. Clustering results

After running the k-means model, and some data operations to add the cluster values to the dataset, we created a new table that includes the clusters column as well as the top 5 restaurant venue

categories of each neighborhood in the following table.

Table 5. Neighborhoods with the top 5 most common venues in restaurants category including clustering labels.

| | Borough | Neighborhood | Population | Latitude | Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Berlin | Charlottenburg-Wilmersdorf | 341327 | 52.497058 | 13.296490 | 2 | Japanese Restaurant | Sushi Restaurant | Doner Restaurant | Snack Place | Chinese Restaurant |
| 1 | Berlin | Friedrichshain-Kreuzberg | 289120 | 52.501500 | 13.435120 | 3 | Pizza Place | Vietnamese Restaurant | Falafel Restaurant | Middle Eastern Restaurant | Breakfast Spot |
| 2 | Berlin | Lichtenberg | 290493 | 52.514581 | 13.498392 | 1 | Vietnamese Restaurant | Pizza Place | Asian Restaurant | Currywurst Joint | Gastropub |
| 3 | Berlin | Marzahn-Hellersdorf | 268739 | 52.539720 | 13.584280 | 2 | Doner Restaurant | Snack Place | Italian Restaurant | Restaurant | Asian Restaurant |
| 4 | Berlin | Mitte | 383457 | 52.516740 | 13.366790 | 0 | Fast Food Restaurant | Burger Joint | Vegetarian / Vegan Restaurant | German Restaurant | Sandwich Place |
| 5 | Berlin | Neukölln | 330786 | 52.480200 | 13.433640 | 3 | Pizza Place | Breakfast Spot | Korean Restaurant | Middle Eastern Restaurant | Bistro |
| 6 | Berlin | Pankow | 407039 | 52.571050 | 13.404970 | 2 | Breakfast Spot | Fast Food Restaurant | Trattoria/Osteria | Italian Restaurant | Burger Joint |
| 7 | Berlin | Reinickendorf | 264826 | 52.567550 | 13.331650 | 2 | Fast Food Restaurant | Doner Restaurant | Trattoria/Osteria | Italian Restaurant | Restaurant |
| 8 | Berlin | Spandau | 243080 | 52.550090 | 13.200356 | 2 | Turkish Restaurant | Italian Restaurant | Fast Food Restaurant | Doner Restaurant | German Restaurant |
| 9 | Berlin | Steglitz-Zehlendorf | 308077 | 52.443640 | 13.229080 | 4 | Italian Restaurant | German Restaurant | Greek Restaurant | Asian Restaurant | Trattoria/Osteria |
| 10 | Berlin | Tempelhof-Schöneberg | 351429 | 52.447630 | 13.385350 | 2 | Restaurant | Indian Restaurant | Taverna | German Restaurant | Gastropub |
| 11 | Berlin | Treptow-Köpenick | 269775 | 52.445817 | 13.574580 | 4 | German Restaurant | Fast Food Restaurant | Italian Restaurant | Greek Restaurant | Asian Restaurant |

Finally, let's visualize the results using Folium map. We can see the central region of Berlin which named Mitte, distinguished from the other clusters.

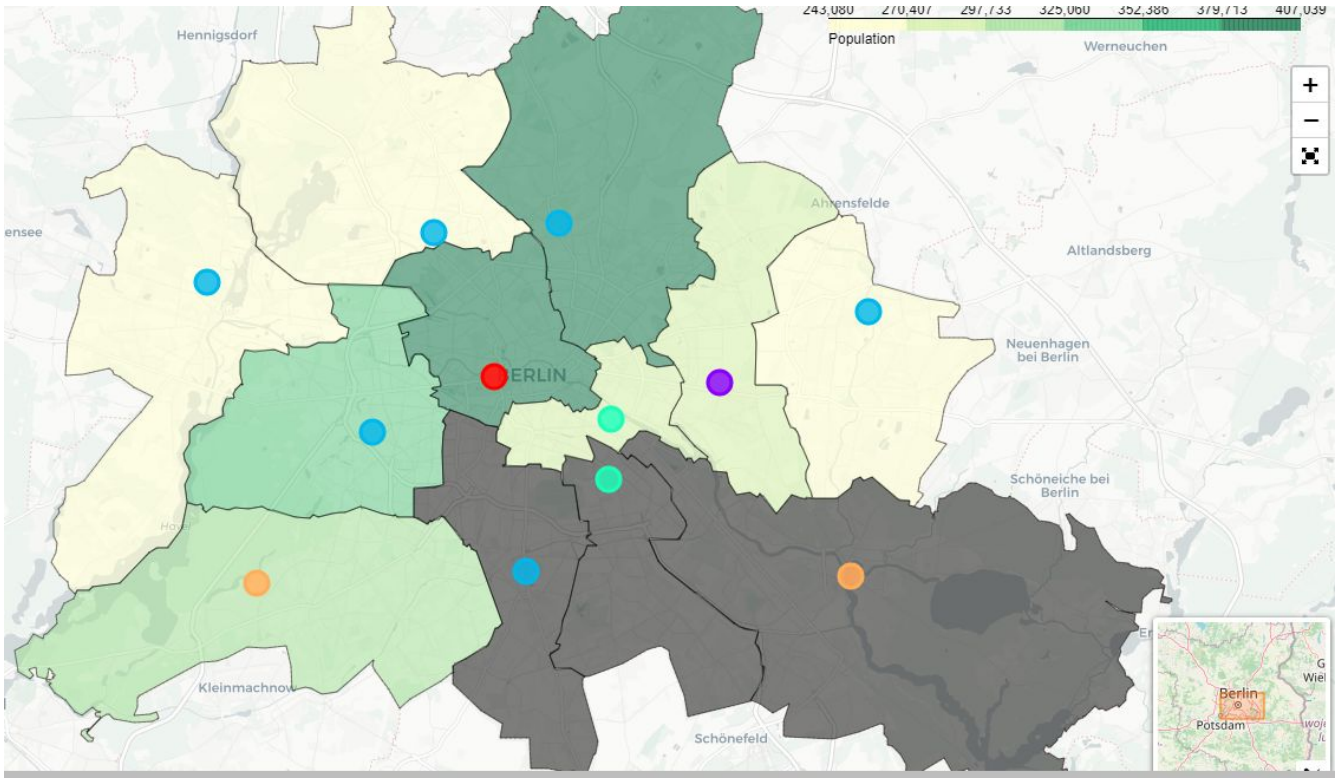



Image 3. Map of Berlin neighborhoods including the results of clustering

Now, let us look at each cluster and determine the venue categories that distinguish each cluster. The tables containing each cluster's venue categories are given below. In Cluster-1, only Mitte region is

classified and the most common venues in the top two are fast-food restaurants. We can not see any Asian restaurants in this neighborhood. So the Asian restaurant business can be promising in this region. If we look at the Cluster 2 we see just the Lichtenberg region. The most common restaurant venue categories in this neighborhood are Vietnamese Restaurant, Pizza Places and Asia restaurants. This says us that we should be aware of the saturation of Asian restaurants in this region. In addition, we take into account that, Asian restaurants include Vietnamese restaurants too. Let us look at the Cluster-3. In this cluster there are 5 neighborhoods (Charlottenburg-Wilmersdorf, Marzahn-Hellersdorf, Pankow, Reinickendorf, Spandau, Tempelhof-Schöneberg) with various types of cuisines, especially with the commonalities of Turkish, Italian and Fast Food restaurants. In cluster-4, Friedrichshain-Kreuzberg and Neukölln have Pizza palace and Middle east Restaurants in common. Finally, we see that Italian, German and Greek restaurants are in common in Cluster-5. In addition to this clustering study, the population, number of these venues, number and location of touristic regions must be taken into account.

Table 6. Berlin neighborhoods with the top 5 restaurant venue categories for each Cluster

Cluster 1

```
neighborhoods_merged.loc[neighborhoods_merged['Cluster Labels'] == 0, neighborhoods_merged.columns[[1] + list(range(5, neighborhoods_merged.shape[1]))]]
```

| | Neighborhood | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|---|
| 4 | Mitte | 0 | Fast Food Restaurant | Burger Joint | Vegetarian / Vegan Restaurant | German Restaurant | Sandwich Place |

Cluster 2

```
neighborhoods_merged.loc[neighborhoods_merged['Cluster Labels'] == 1, neighborhoods_merged.columns[[1] + list(range(5, neighborhoods_merged.shape[1]))]]
```

| | Neighborhood | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|---|
| 2 | Lichtenberg | 1 | Vietnamese Restaurant | Pizza Place | Asian Restaurant | Currywurst Joint | Gastropub |

Cluster 3

```
neighborhoods_merged.loc[neighborhoods_merged['Cluster Labels'] == 2, neighborhoods_merged.columns[[1] + list(range(5, neighborhoods_merged.shape[1]))]]
```

| | Neighborhood | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|---|
| 0 | Charlottenburg-Wilmersdorf | 2 | Japanese Restaurant | Sushi Restaurant | Doner Restaurant | Snack Place | Chinese Restaurant |
| 3 | Marzahn-Hellersdorf | 2 | Doner Restaurant | Snack Place | Italian Restaurant | Restaurant | Asian Restaurant |
| 6 | Pankow | 2 | Breakfast Spot | Fast Food Restaurant | Trattoria/Osteria | Italian Restaurant | Burger Joint |
| 7 | Reinickendorf | 2 | Fast Food Restaurant | Doner Restaurant | Trattoria/Osteria | Italian Restaurant | Restaurant |
| 8 | Spandau | 2 | Turkish Restaurant | Italian Restaurant | Fast Food Restaurant | Doner Restaurant | German Restaurant |
| 10 | Tempelhof-Schöneberg | 2 | Restaurant | Indian Restaurant | Taverna | German Restaurant | Gastropub |

Cluster 4

```
neighborhoods_merged.loc[neighborhoods_merged['Cluster Labels'] == 3, neighborhoods_merged.columns[[1] + list(range(5, neighborhoods_merged.shape[1]))]]
```

| | Neighborhood | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|---|
| 1 | Friedrichshain-Kreuzberg | 3 | Pizza Place | Vietnamese Restaurant | Falafel Restaurant | Middle Eastern Restaurant | Breakfast Spot |
| 5 | Neukölln | 3 | Pizza Place | Breakfast Spot | Korean Restaurant | Middle Eastern Restaurant | Bistro |

Cluster 5

```
neighborhoods_merged.loc[neighborhoods_merged['Cluster Labels'] == 4, neighborhoods_merged.columns[[1] + list(range(5, neighborhoods_merged.shape[1]))]]
```

| | Neighborhood | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|---|
| 9 | Steglitz-Zehlendorf | 4 | Italian Restaurant | German Restaurant | Greek Restaurant | Asian Restaurant | Trattoria/Osteria |
| 11 | Treptow-Köpenick | 4 | German Restaurant | Fast Food Restaurant | Italian Restaurant | Greek Restaurant | Asian Restaurant |

## 3.3. Analyze and Make Prediction Models

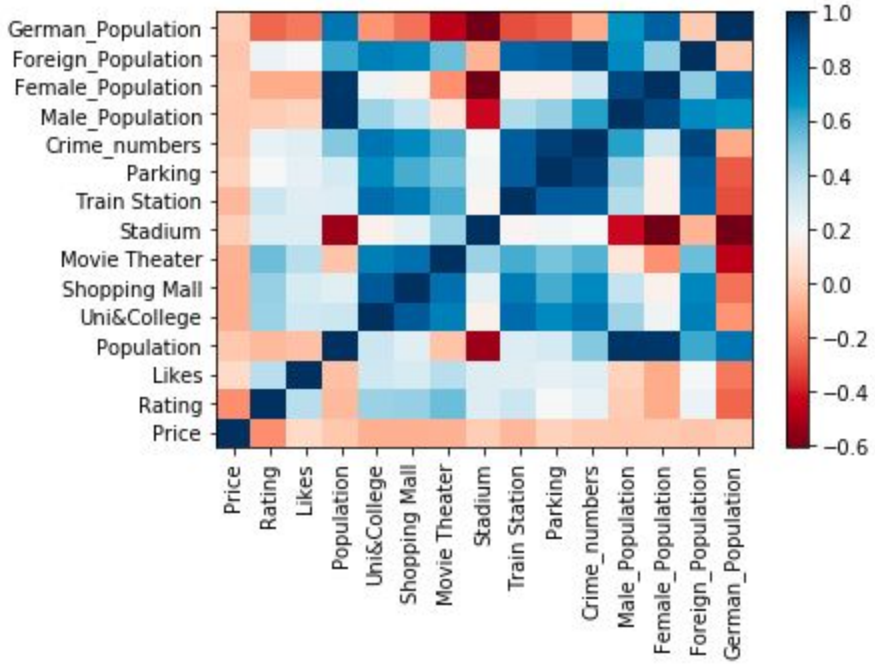
In reality, there are multiple variables that affect the success of a restaurant business. When more than one independent variable like in this study is present, the multiple linear regression model can be used to predict the success of the business. We have limited open data about the success of this restaurant business. And this target should be communicated and discussed with the project counterpart effectively. On the other hand, we have Foursquare API data and some other public data in hand. Within these data, we can choose a number of 'Likes', 'Rating' or some combination of them as a target/success variable (dependent variable). After checking API explanations, we decided to use 'Rating' as a dependent variable representing the success of the business.

Cleaning the data and determining the independent variables are the first step of modeling. Let us determine the independent variables and extract as a new dataset. Columns of our preliminary dataset are as follows

['Neighborhood', 'Venue Id', 'Venue', 'Venue Category', 'Price',
'Rating', 'Likes', 'Population', 'Uni&College', 'Shopping Mall',
'Movie Theater', 'Stadium', 'Train Station', 'Parking', 'Crime_numbers',
'Male_Population', 'Female_Population', 'Foreign_Population',
'German_Population']

Now, let's look at the correlation between numeric columns using the heatmap below.

Figure 5. Heat map of numeric data within the Venue Details data set.




We can see the dependencies between variables in the heatmap. Let us a closer look at actual values for our dependent/target variable 'Rating' below. The list is in ascending order.

Table 7. Correlation between Rating and independent variables

```
dfnew.corr()['Rating'].sort_values()
```

```
German_Population    -0.249195
Price                -0.159745
Female_Population    -0.093123
Population           -0.046426
Male_Population      -0.000610
Parking               0.203134
Foreign_Population    0.241458
Crime_numbers         0.250090
Stadium               0.287544
Train Station         0.350554
Likes                 0.392928
Uni&College           0.456468
Shopping Mall         0.471028
Movie Theater         0.545049
Rating                1.000000
Name: Rating, dtype: float64
```

Let's dismiss the independent variables which have less than 0.20 correlation score with 'Rating' and select *'Likes', 'Population', 'Uni&College', 'Shopping Mall', 'Movie Theater', 'Stadium', 'Train Station', 'Parking', 'Crime_numbers', 'Foreign_Population', 'German_Population'* columns as independent variables for our model. Let us construct our model with these independent and dependent data.

**3.3.1. Multiple regression model results**

When we run a multiple regression model by importing from sklearn library, we found the following coefficients and R-square values.

*Multiple Regression Model Coefficients:*

[ 9.71291557e-04 2.92238414e+09 1.70648096e-01 -1.65198471e-01
3.02673696e-02 4.61008151e-01 7.13941708e-02 5.72347906e-02
-1.51110784e-04 -2.92238414e+09 -2.92238414e+09]

*The R-square is:* 0.4126715675897107

In addition to R-Square, we calculate 'variance score' and mean squared error (MSE) as well.

*Residual sum of squares:* 0.86 and *Variance score:* 0.22
*The mean square error of 'Rating' and predicted value:* 0.7654593187366441

## Prediction with normalized data

Let's repeat the same prediction model after normalizing the data.

***Multiple Regression Model Coefficients:***

[ 0.25418166 0.70721605 0.8650806 -0.49001694 0.23463565 0.88998317
0.09311164 0.5791981 -2.83848416 -0.24770434 1.4420277 ]

*The R-square:* 0.43563698313733556

*Residual sum of squares:* 0.77 and *Variance score:* 0.30
*The mean square error of Rating and predicted value:* 0.7654593187366455

When we compared the non-normalized model results with the normalized one, R-square values are almost the same. So, there are not many implications to the result in this multiple regression model. On the other hand, if we look at the Coefficient values, normalized values seem much better (i.e. order of magnitude). the variance scores are 0.41 and 0.43 respectively.

## Cross-validation Score

When we look at the data set, we see that the samples/rows are limited. To eliminate this difficulty we can use cross-validation. Let's do cross-validation and see the effect of changing train and test data sets using.

When we run the test, we get the following R-Square values for each iteration.

*Rcross:* [0.18368166, 0.35121787, 0.39947013, 0.34026652, 0.27882088]

We can calculate the average and standard deviation of our estimate:

*The mean of the iterations(R-Square):* 0.31069141214039125

*The standard deviation:* 0.07422391892277923

These scores are much closer to actual values. However, these values are not promising results.
Let us make another model and check the results.

## Ridge regression model

We use the same dataset with the previous model to be able to compare the results. Before running the ridge model 'Grid Search' give us valuable data such as the best parameters.

Sklearn has the class *GridSearchCV* to make the process of finding the best hyperparameter simpler. Using *GridSearchCV* (that includes cross-validation as well) we generated the following parameters.

'alpha': [0.001,0.1,1, 10, 100, 1000, 10000, 100000, 100000]

We created a ridge regions object as follows:

```
Ridge(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=None,
    normalize=False, random_state=None, solver='auto', tol=0.001)
```

Then, we created a ridge grid search object using a cross-validation parameter cv as 4.

```
Grid1 = GridSearchCV(RR, parameters1,cv=4)
```
We fitted the model with following sklearn command: Grid1.fit(x_train, y_train) and generated following GridSearchCV parameters:

```
GridSearchCV(cv=4, error_score='raise-deprecating',
        estimator=Ridge(alpha=1.0, copy_X=True, fit_intercept=True,
                max_iter=None, normalize=False, random_state=None,
                solver='auto', tol=0.001),
        iid='warn', n_jobs=None,
        param_grid=[{'alpha': [0.001, 0.1, 1, 10, 100, 1000, 10000, 100000,
                        100000]}],
        pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
        scoring=None, verbose=0)
```

**3.3.2. Ridge regression model results**

This object finds the best parameter values on the validation data. We can obtain the estimator with the best parameters and assign it to the variable BestRR. Best RR value is below:

```
Ridge(alpha=10, copy_X=True, fit_intercept=True, max_iter=None, normalize=False,
    random_state=None, solver='auto', tol=0.001)
```

Then, we applied the model using Grid1.best_estimator_ and obtained the following result.

*The R-square:* 0.3470253669057909

## 4. Discussion and Conclusion

When we compare the multiple linear regression model with the ridge regression model, the multiple linear regression model seems favorable(R-Square values are 0.43 and 0.35 respectively). On the other hand, in reality, cross-validated results should be compared to one another. So if we compare the results of the cross-validated multiple linear regression model with the results of the Ridge regression model, which implicitly includes cross-validated results( with the GridsearchCV), Ridge regression model is favorable model. Because R-Square values are 0.31 and 0.35 respectively. We know that these values are not so promising. One of the reasons is the limited data. Almost half of the restaurant avenue data have not rating score which is the dependent variable in this project.

In addition, this project can be extended by calculating the distances from people's gathering places to each restaurants and be added to the models. Prediction models can be repeated with a new target-dependent variable such as some combination of ‘Likes’ and ‘Rating’. Besides, as a target variable, revenues of restaurants can be used if available.

This capstone project gave us very valuable skills such as; using machine learning algorithms, working with real life data, scraping web data, manipulating and visualizing geo-locational data, using Folium Leaflet Map library, using APIs and exploring open data sources etc. In conclusion, thank you IBM and Coursera for this valuable certificate program.

Prepared by Sancak Özdemir
20.04.2020