Opening a new mall in the city of Strasbourg

1.Introduction

1.1 Background

People like to do several activities during the weekends or during the holidays like shopping, eating at restaurants with friends. Several shopping malls particulary in the city of Strasbourg allow them to realize their dreams by offering them a wide range of activities as eating in a restaurant. For an investor who wants to meet the customer's needs, opening a mall in a place where customer traffic is important is important. That will determine whether he is going to have a return on investment or go bankrupt. Therefore, doing web scraping in order to leverage data by cleaning, exploring and clustering the neighborhoods is crucial.

1.2 Business Problem

Data that might contribute to determine the best place to open a mall should include the list of neighborhoods in Strasbourg, Latitude and Longitude coordinates of these neighborhoods and venue data particularly the venue related to shopping malls. This project aims to determine where will a stakeholder open a mall in order to raise his turnover and make a perk

1.3 Targeted Audience

This project particularly concerns those who want to become shareholders or who want to open a shopping mall in the best places on the neighborhoods in Strasbourg

2. Data acquisition and wrangling

2.1 Data sources

https://fr.wikipedia.org/wiki/Liste_des_quartiers_de_Strasbourg is a wikipedia page that have all the neighborhoods in Strasbourg. There are 15 neighborhoods in Strasbourg. We will scrape the Wikipedia page and wrangle the data, clean it, and then read it into a *pandas* dataframe. Many data science skills are required to lead this project as :

•Using web scraping allows us to extract the data with the help of libraries such as BeautifulSoup4 and pandas. Then we will clean the data

•Importing Nominatim library from Python Geocoder package in order to transform addresses of the neighborhoods into latitude and longitude coordinates. This will be useful for visualize the data

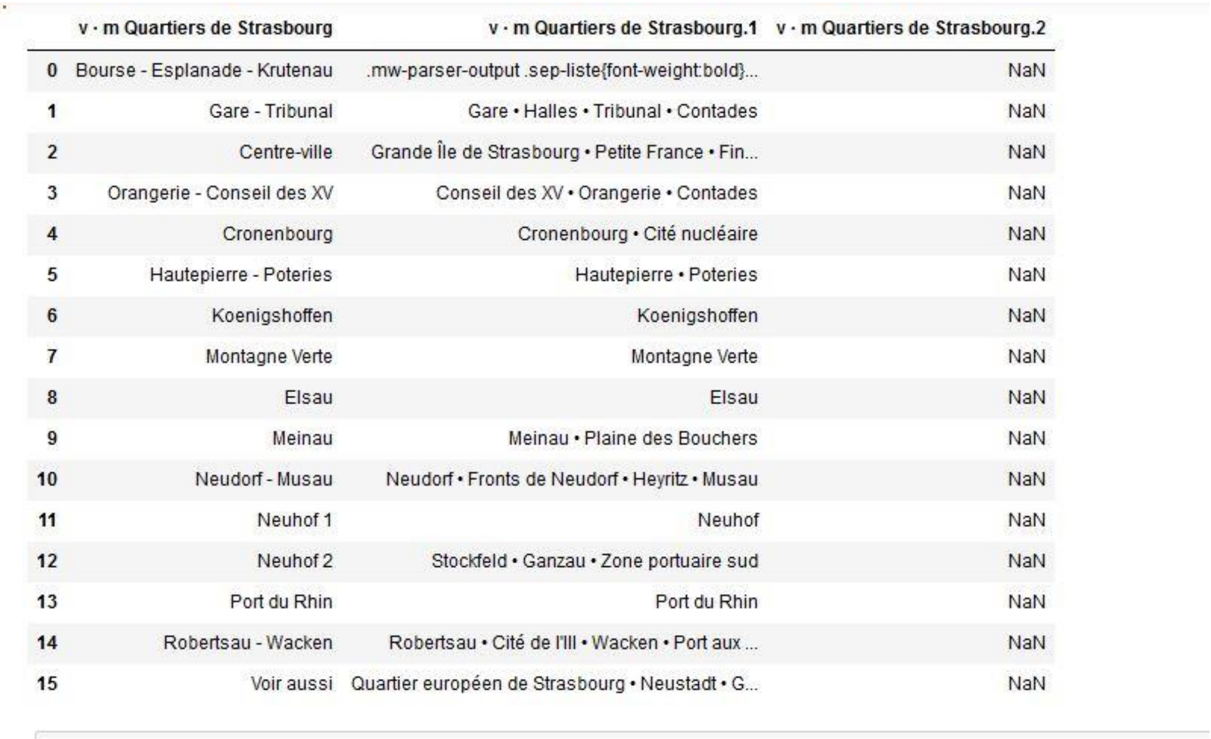
•Using Foursquare API in order to get the venue data of those neighborhoods. Foursquare provides many categories of the venue data but we are interested in Shopping mall category.

•Doing Machine Learning by using K-means clustering

•Visualize the data by using the folium library

2.2 Data Cleaning

After I had scrapped data from the Wikipedia page, I transformed it into a dataframe by using the BeautifulSoup4 package. I obtained the dataframe below

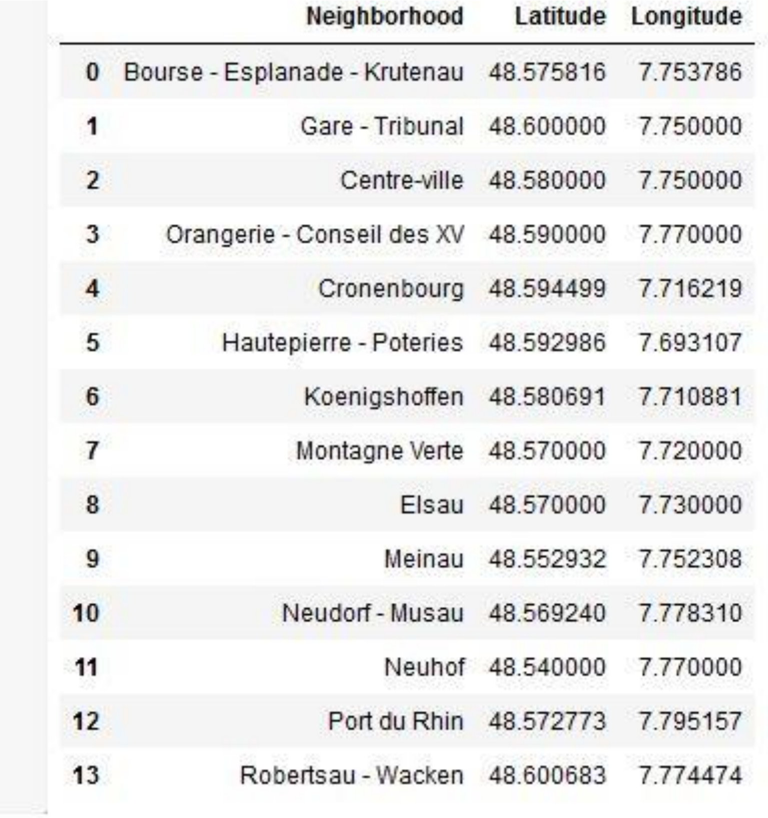


	v · m Quartiers de Strasbourg	v · m Quartiers de Strasbourg.1	v · m Quartiers de Strasbourg.2
0	Bourse - Esplanade - Krutenau	.mw-parser-output .sep-liste{font-weight:bold}...	NaN
1	Gare - Tribunal	Gare • Halles • Tribunal • Contades	NaN
2	Centre-ville	Grande Île de Strasbourg • Petite France • Fin...	NaN
3	Orangerie - Conseil des XV	Conseil des XV • Orangerie • Contades	NaN
4	Cronenbourg	Cronenbourg • Cité nucléaire	NaN
5	Hautepierre - Poteries	Hautepierre • Poteries	NaN
6	Koenigshoffen	Koenigshoffen	NaN
7	Montagne Verte	Montagne Verte	NaN
8	Elsau	Elsau	NaN
9	Meinau	Meinau • Plaine des Bouchers	NaN
10	Neudorf - Musau	Neudorf • Fronts de Neudorf • Heyritz • Musau	NaN
11	Neuhof 1	Neuhof	NaN
12	Neuhof 2	Stockfeld • Ganzau • Zone portuaire sud	NaN
13	Port du Rhin	Port du Rhin	NaN
14	Robertsau - Wacken	Robertsau • Cité de l'Ill • Wacken • Port aux ...	NaN
15	Voir aussi	Quartier européen de Strasbourg • Neustadt • G...	NaN

Figure1: Neighborhoods in Strasbourg

I cleaned the dataframe by dropping the unnecessary columns as «v . m Quartiers de Strasbourg.1» and «v . m Quartiers de Strasbourg.2». Then I renamed the column «v . m Quartiers de Strasbourg» by the name «Neighborhood».

After I created another dataframe df1 in order to get the coordinates of all neighborhoods. The module Nominatim from the library geopy, which allows to transform address into latitude and longitudes was used. After I merge the two dataframes into one dataframe. The result is in the image below



	Neighborhood	Latitude	Longitude
0	Bourse - Esplanade - Krutenau	48.575816	7.753786
1	Gare - Tribunal	48.600000	7.750000
2	Centre-ville	48.580000	7.750000
3	Orangerie - Conseil des XV	48.590000	7.770000
4	Cronenbourg	48.594499	7.716219
5	Hautepierre - Poteries	48.592986	7.693107
6	Koenigshoffen	48.580691	7.710881
7	Montagne Verte	48.570000	7.720000
8	Elsau	48.570000	7.730000
9	Meinau	48.552932	7.752308
10	Neudorf - Musau	48.569240	7.778310
11	Neuhof	48.540000	7.770000
12	Port du Rhin	48.572773	7.795157
13	Robertsau - Wacken	48.600683	7.774474

Figure 2: DataFrame cleaned

3-Methodology

First, I visualize the map of Strasbourg and I add the neighborhoods superimposed on top by using the folium library. This was possible by grouping the markers into different clusters. To implement this, we start off by instantiating a *MarkerCluster* object and adding all the data points in the dataframe to this object. We obtained this map below

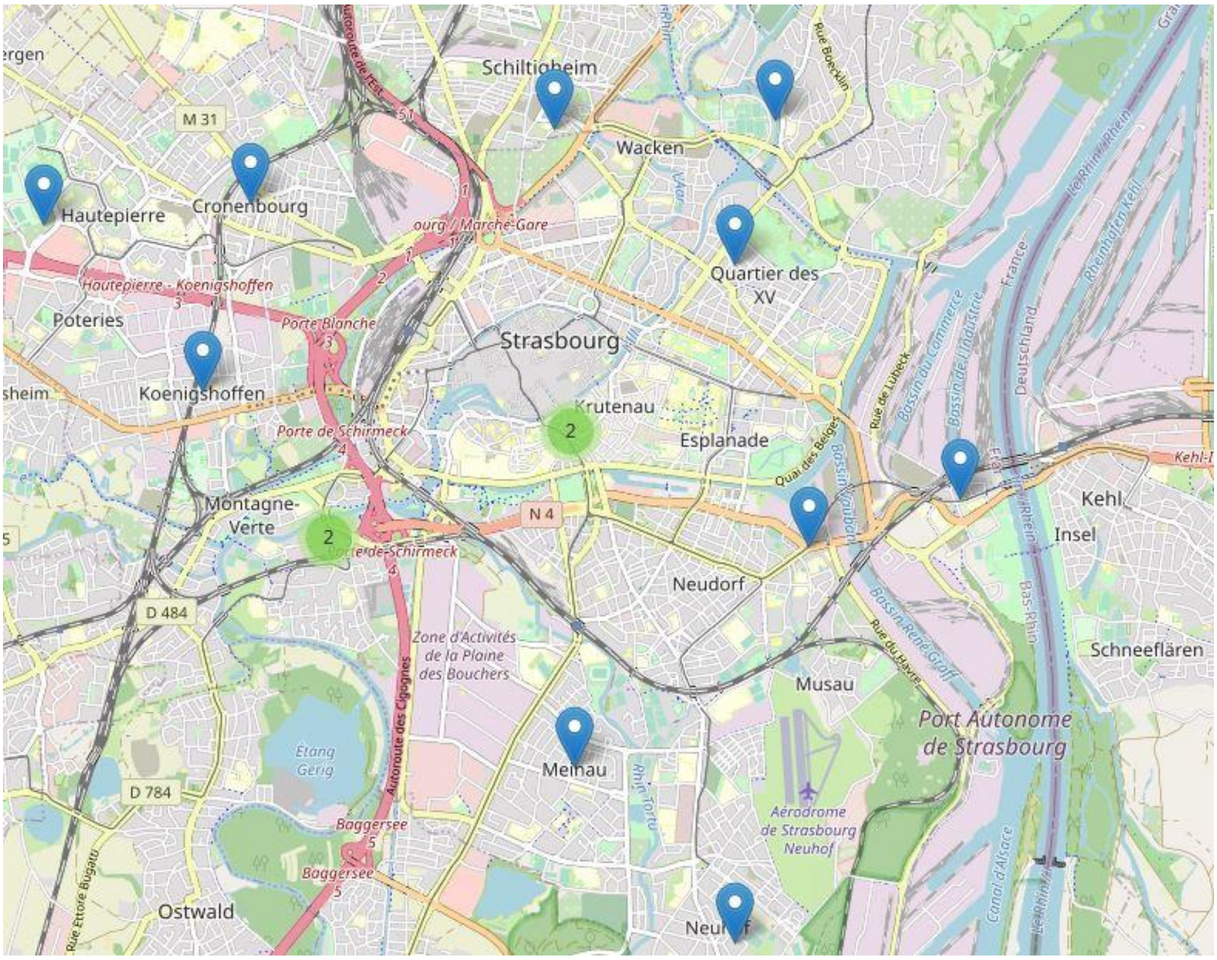
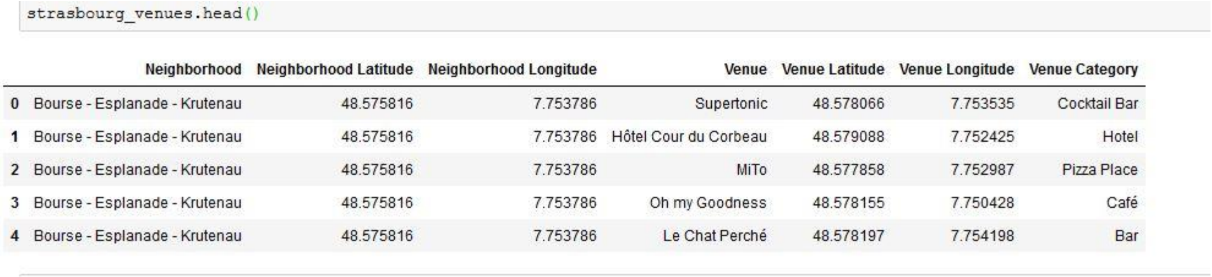


Figure 3 : map Strasbourg

Secondly I used the Foursquare API in order to get the venue name, venue latitude, venue longitude and venue category of each neighborhoods. I designed the limit as 100 venues and the radius 750 meters of each neighborhoods. The head of the result can be seen below



```
strasbourg_venues.head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Bourse - Esplanade - Krutenau	48.575816	7.753786	Supertonic	48.578066	7.753535	Cocktail Bar
1	Bourse - Esplanade - Krutenau	48.575816	7.753786	Hôtel Cour du Corbeau	48.579088	7.752425	Hotel
2	Bourse - Esplanade - Krutenau	48.575816	7.753786	MiTo	48.577858	7.752987	Pizza Place
3	Bourse - Esplanade - Krutenau	48.575816	7.753786	Oh my Goodness	48.578155	7.750428	Café
4	Bourse - Esplanade - Krutenau	48.575816	7.753786	Le Chat Perché	48.578197	7.754198	Bar

Figure4 : data