Capstone Project - The Battle of Neighborhoods - Warsaw

February 3, 2020

1 Capstone Project - The Battle of Neighborhoods - Warsaw

1.0.1 Applied Data Science Capstone by IBM/Coursera

Warsaw

1.1 Table of contents

1.2 ## Introduction

1.2.1 Background

Prices of flats in Poland go up faster than inflation, according to the report of money.pl website. Rising apartment prices on the market effectively obscure another problem - the increase in rental prices. This trend affects students, young workers without their own flats or economic immigrants. According to the analysis of experts at Rynekpotny.pl, the increases reached even 23%. Despite everything, life in Warsaw tempts many young people.

1.2.2 Business Problem

The capital is mainly attracting to itself those who are focused on making dizzying careers or artists and creative people. The heart of the city is the City Centre, which is vibrant with life at any time of day or night. It is one of eighteen districts, but each of them has different advantages. In this scenario, machine learning tools should be used to assist people coming to Warsaw to make wise and effective decisions. As a result, the business problem is:

- **How can we help people moving to the capital to choose the right location to rent an flat in Warsaw?**

In order to solve this business problem, we intend to merge Warsaw districts into a cluster in order to recommend facilities. We will recommend facilities according to the amenities and necessary equipment of the surrounding facilities such as: **Café**, **Restaurant**, **Park**.

1.3 ## Data

To consider the objective stated above, we can list the below data sources used for the analysis.

- **Districts of Warsaw** Wikipedia page was scraped to pull out the necessary information;
- **Coordinate data** for each Districts of Warsaw obtained through Nominatim search engine for OpenStreetMap data;

In order to investigate and target recommended locations in different locations depending on the presence of facilities and necessary objects, we will access the data through the **FourSquare API** and arrange it as a data frame for visualization. By combining data about districts in Warsaw and data about amenities and essential facilities surrounding such properties from the FourSquare API, we will be able to recommend an appropriate location.

1.4 ## Methodology

The Methodology section will describe the main elements of the analysis and prediction system. The methodological part consists of four stages:

1. Data Preparation
2. Visualization and Data Exploration
3. Data preparation and Preprocessing
4. Modeling

1.4.1 Data Preparation

Scrape the Wikipedia page and gathering data into a Pandas dataframe To start with our analysis, we used the BeautifulSoup package to transform the data in the table on the Wikipedia page into the below pandas dataframe. Subsequently, we transform the data into a pandas dataframe.

[1]:
```
import urllib.request, urllib.parse, urllib.error
import pandas as pd

# !conda install -c anaconda beautifulsoup4
from bs4 import BeautifulSoup
```

[2]:
```
url ="https://en.wikipedia.org/wiki/Districts_of_Warsaw"
html = urllib.request.urlopen(url).read()

warsaw_dist_wiki = BeautifulSoup(html, 'html.parser')
```

[3]:
```
warsaw_data = pd.DataFrame({'District' : [''], 'Neighborhood' : ['']})

warsaw_dist_wiki = BeautifulSoup(html, 'html.parser')
wiki_table = warsaw_dist_wiki.findAll('table')

wiki_neighborhood = []
```

```
for td in wiki_table[1].find_all('td'):
    wiki_neighborhood.append(td)

warsaw_districts =[]
for th in wiki_table[1].find_all('th'):
    warsaw_districts.append(th.text.strip())
print(f'Warsaw is divided into {len(warsaw_districts)} districts, each one with␣
 ↪its own administrative body.')

print('Each of the districts is customarily subdivided into several␣
 ↪neighbourhoods:')
k = 0
for i in range(len(warsaw_districts)):
    k=k-1
    for j in wiki_neighborhood[i].find_all('li'):
        k=k+1
        warsaw_data.loc[i+k] = [warsaw_districts[i], j.text.strip()]

    temp_loc = len(wiki_neighborhood[i].find_all('li'))
    print(f'- {warsaw_districts[i]} - {temp_loc} ')
```

```
Warsaw is divided into 18 districts, each one with its own administrative body.
Each of the districts is customarily subdivided into several neighbourhoods:
- Bemowo - 10
- Biaoka - 11
- Bielany - 14
- Mokotów - 12
- Ochota - 4
- Praga-Poudnie - 6
- Praga-Pónoc - 4
- Rembertów - 3
- ródmiecie - 8
- Targówek - 7
- Ursus - 5
- Ursynów - 14
- Wawer - 12
- Wesoa - 6
- Wilanów - 8
- Wochy - 8
- Wola - 8
- oliborz - 3
```

[4]:
```
print('The dataframe has {} districts and {} neighborhoods.'.format(
        len(warsaw_data['District'].unique()),
        warsaw_data.shape[0]
    )
)
```

```
warsaw_data.head()
```

```
The dataframe has 18 districts and 143 neighborhoods.
```

[4]:

```
  District      Neighborhood
0   Bemowo  Bemowo Lotnisko
1   Bemowo       Boernerowo
2   Bemowo         Chrzanów
3   Bemowo        Fort Bema
4   Bemowo     Fort Radiowo
```

Use geopy library to get the latitude and longitude values of Warsaw Localities After we have built a dataframe of Warsaw localities along with the district name and neighborhood name, in order to utilize the Foursquare location data, we need to get the latitude and the longitude coordinates of each neighborhood. It possible to export data to a csv file for easier loading later.

[5]:

```
warsaw_data['Latitude'] = ''
warsaw_data['Longitude'] = ''
warsaw_data.head()
```

[5]:

```
  District      Neighborhood Latitude Longitude
0   Bemowo  Bemowo Lotnisko
1   Bemowo       Boernerowo
2   Bemowo         Chrzanów
3   Bemowo        Fort Bema
4   Bemowo     Fort Radiowo
```

[6]:

```
# !conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim
```

```
import time

geolocator = Nominatim(user_agent="to_explorer")

for id in range(len(warsaw_data)):
    address = warsaw_data['Localities'][id] + ', Warsaw, Poland'
    location = geolocator.geocode(address)
    warsaw_data['Latitude'][id] = location.latitude
    warsaw_data['Longitude'][id] = location.longitude
    time.sleep(1)

warsaw_data.to_csv('warsaw_localities.csv', index = None, header=True)
```

[7]:

```
warsaw_data = pd.read_csv('warsaw_localities.csv', header=0)
warsaw_data.head()
```

[7]:

	District	Neighborhood	Latitude	Longitude
0	Bemowo	Bemowo Lotnisko	52.261261	20.910737
1	Bemowo	Boernerowo	52.262390	20.901451
2	Bemowo	Chrzanów	52.216759	20.882969
3	Bemowo	Fort Bema	52.256562	20.938620
4	Bemowo	Fort Radiowo	52.257211	20.891900

Utilizing Foursquare API to explore the neighborhoods Foursquare is the most trusted, independent location data platform for understanding how people move through the real world. We have used, as a part of the assignment, the Foursquare API to retrieve information about the popular spots for each neighborhoods of Warsaw. The recommended location needs to have many eating and shopping venues nearby. Convenient public transport is also required.

Foursquare credentials are defined in hidden cell bellow.

[8]:
```
CLIENT_ID = '5D53WUUPEJYZNSE3AF2SZCCPZE1RAW32JLQJE1JNIFJSOOVH' # your␣
↪Foursquare ID
CLIENT_SECRET = '252PIMDDHM2QA4ETIPRACOMCLFZIAWWMMNVT3QGAY3MS3JXX' # your␣
↪Foursquare Secret
VERSION = '20180605' # Foursquare API version
```

[9]:
```
from pandas.io.json import json_normalize # tranform JSON file into a pandas␣
↪dataframe
import requests
```

Create a nearby venues function for all the neighborhoods in Warsaw

[10]:
```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
#         print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?
↪&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
```

```
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item␣
↪in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']
    return(nearby_venues)
```

We chose 100 popular places for each neighborhoods within a 10 km radius.

[11]:
```
radius = 6000 # define radius
LIMIT  = 200  # limit of number of venues returned by Foursquare API
```

Create a new dataframe called for the venues of Warsaw

[12]:
```
warsaw_venues = getNearbyVenues(names=warsaw_data['Neighborhood'],
                                   latitudes=warsaw_data['Latitude'],
                                   longitudes=warsaw_data['Longitude']
                                  )
print('Import completed')
```

```
Import completed
```

Below is the data frame obtained from the JSON file returned by Foursquare.

[13]:
```
print('Total {} of venues are found'.format(len(warsaw_venues)))
warsaw_venues.head()
```

```
Total 1427 of venues are found
```

[13]:
```
      Neighborhood  Neighborhood Latitude  Neighborhood Longitude  \
0  Bemowo Lotnisko              52.261261               20.910737
1  Bemowo Lotnisko              52.261261               20.910737
2  Bemowo Lotnisko              52.261261               20.910737
3  Bemowo Lotnisko              52.261261               20.910737
4  Bemowo Lotnisko              52.261261               20.910737

                Venue  Venue Latitude  Venue Longitude   Venue Category
0           Goldwings       52.260579        20.910778    Flight School
1      Hostel Kingroom       52.264355        20.912432           Hostel
```

```
2  Dach Nacipanej Vistuli       52.259773        20.915648  Airport Service
3                   Garae      52.258142        20.914198      Beer Garden
4                 Place4Us      52.263905        20.915367            Hotel
```

1.4.2 Visualization and Data Exploration

Generating a map of Warsaw and plotting the Neighborhood data on it

[14]:
```
address = 'Warsaw, Poland'

geolocator = Nominatim(user_agent="to_explorer") # GeocoderTimedOut: Service␣
↪timed out
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude

# latitude = 52.2337172
# longitude = 21.07141112883227
print('The geograpical coordinate of {} are {}, {}.'.format(address, latitude,␣
↪longitude))
```

```
The geograpical coordinate of Warsaw, Poland are 52.2337172, 21.07141112883227.
```

[15]:
```
#!conda install -c conda-forge folium=0.5.0 --yes
import folium
```

[16]:
```
map_warsaw = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, district in zip(warsaw_data['Latitude'],␣
↪warsaw_data['Longitude'], warsaw_data['Neighborhood']):
    label = '{}'.format(district)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=7,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_warsaw)

map_warsaw
```

[16]:
```
<folium.folium.Map at 0x1d30c3832e8>
```

Numbers of venues for each neighborhood

[17]:
```
map_warsaw = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, district in zip(warsaw_venues['Venue Latitude'],␣
↪warsaw_venues['Venue Longitude'], warsaw_venues['Venue Category']):
    label = '{}'.format(district)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_warsaw)

map_warsaw
```

[17]:
```
<folium.folium.Map at 0x1d30c5e9b38>
```

[18]:
```
warsaw_venues.groupby('Neighborhood').count().head()
```

[18]:
```
                     Neighborhood Latitude  Neighborhood Longitude  Venue  \
Neighborhood
Anin                                    13                      13     13
Bemowo Lotnisko                          5                       5      5
Biaoka Dworska                        1                       1      1
Boernerowo                               4                       4      4
Bródno                                   9                       9      9

                     Venue Latitude  Venue Longitude  Venue Category
Neighborhood
Anin                             13               13              13
Bemowo Lotnisko                   5                5               5
Biaoka Dworska                 1               1               1
Boernerowo                        4                4               4
Bródno                            9                9               9
```

Numbers of unique categories can be curated from all the returned venues

[19]:
```
print('There are {} uniques categories.'.format(len(warsaw_venues['Venue␣
↪Category'].unique())))
```

```
There are 234 uniques categories.
```

Examples of Neighborhood meeting the Venue Category: **Café**

[20]:
```
warsaw_venues[warsaw_venues['Venue Category']=='Café'].head()
```

[20]:
```
     Neighborhood  Neighborhood Latitude  Neighborhood Longitude  \
16      Fort Bema              52.256562               20.938620
25          Górce              52.245431               20.913714
63        Kobiaka              52.354573               21.043018
80      Tarchomin              52.318028               20.954304
138     Sodowiec               52.276825               20.960235

                    Venue  Venue Latitude  Venue Longitude Venue Category
16   Cafe Jurta Forty Bema       52.257985        20.935512           Café
25                 CieKawa       52.242059        20.913374           Café
63           Cafe Karolinka      52.355282        21.038461           Café
80                Carmelia       52.321284        20.955756           Café
138     COSTA Stare Bielany      52.275127        20.961906           Café
```

1.4.3 Data preparation and Preprocessing

The number of objects found in the category discussed in the business problem

[21]:
```python
# selected_category = ['Café', 'Park', 'Pizza Place', 'Restaurant']
selected_category = ['Café', 'Restaurant', 'Park']
```

Numbers of place in selected categories

[22]:
```python
for cat in selected_category:
    print(cat, warsaw_venues[warsaw_venues['Venue Category'].str.contains(cat)].
 ↪shape[0])
```

```
Café 86
Restaurant 334
Park 51
```

Select only districts with interesting objects

[23]:
```python
selected_category = warsaw_venues[warsaw_venues['Venue Category'].
 ↪isin(selected_category)]
```

[24]:
```python
selected_category.head()
```

[24]:
```
    Neighborhood  Neighborhood Latitude  Neighborhood Longitude  \
15     Fort Bema              52.256562               20.938620
16     Fort Bema              52.256562               20.938620
25         Górce              52.245431               20.913714
35         Groty              52.250801               20.873235
37         Groty              52.250801               20.873235

                    Venue  Venue Latitude  Venue Longitude Venue Category
15               Fort Bema       52.256450        20.936549           Park
16   Cafe Jurta Forty Bema       52.257985        20.935512           Café
25                 CieKawa       52.242059        20.913374           Café
35              Krasnodwór       52.253985        20.873138     Restaurant
```

```
37  Lasy Miejskie Warszawy       52.253175        20.867496            Park
```

[25]:
```python
warsaw_onehot = pd.get_dummies(selected_category[['Venue Category']],␣
↪prefix="", prefix_sep="")
warsaw_onehot['Neighborhood'] = selected_category['Neighborhood']

fixed_columns = [warsaw_onehot.columns[-1]] + list(warsaw_onehot.columns[:-1])
warsaw_onehot = warsaw_onehot[fixed_columns]

warsaw_onehot.head()
```

[25]:

	Neighborhood	Café	Park	Restaurant
15	Fort Bema	0	1	0
16	Fort Bema	1	0	0
25	Górce	1	0	0
35	Groty	0	0	1
37	Groty	0	1	0

[26]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
```

[27]:
```python
warsaw_grouped = warsaw_onehot.groupby('Neighborhood').mean().reset_index()
warsaw_grouped.head()
```

[27]:

	Neighborhood	Café	Park	Restaurant
0	Anin	0.5	0.5	0.0
1	Bródno	0.0	1.0	0.0
2	Bródno Podgrodzie	1.0	0.0	0.0
3	Bonia Wilanowskie	0.0	1.0	0.0
4	Czerniaków	1.0	0.0	0.0

[28]:
```python
warsaw_grouped = warsaw_grouped[warsaw_grouped.loc[:]!=0].dropna()
warsaw_grouped
```

[28]:

	Neighborhood	Café	Park	Restaurant
8	Grochów	0.600000	0.200000	0.200000
25	Natolin	0.200000	0.400000	0.400000
31	Powile	0.600000	0.200000	0.200000
42	Stara Praga	0.250000	0.250000	0.500000
46	Stary Mokotów	0.600000	0.200000	0.200000
47	Stary oliborz	0.500000	0.250000	0.250000
62	ródmiecie Poudniowe	0.750000	0.083333	0.166667
63	ródmiecie Pónocne	0.444444	0.222222	0.333333

[29]:
```python
num_top_venues = 5

for hood in warsaw_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = warsaw_grouped[warsaw_grouped['Neighborhood'] == hood].T.
↪reset_index()
    temp.columns = ['venue','freq']
```

```
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
```

```
----Grochów----
----Natolin----
----Powile----
----Stara Praga----
----Stary Mokotów----
----Stary oliborz----
----ródmiecie Poudniowe----
----ródmiecie Pónocne----
```

[30]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

[31]:
```
import numpy as np
```

[32]:
```
num_top_venues = 2

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = warsaw_grouped['Neighborhood']

for ind in np.arange(warsaw_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] =␣
 ↪return_most_common_venues(warsaw_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted
```

[32]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue
8	Grochów	Café	Restaurant
25	Natolin	Restaurant	Park
31	Powile	Café	Restaurant
42	Stara Praga	Restaurant	Park
46	Stary Mokotów	Café	Restaurant

```
47         Stary oliborz                     Café            Restaurant
62  ródmiecie Poudniowe                    Café            Restaurant
63    ródmiecie Pónocne                    Café            Restaurant
```

[33]:
```python
selected_neighborhood = warsaw_grouped['Neighborhood'].values
target_warsaw_venues = warsaw_venues[warsaw_venues['Neighborhood'].
 ↪isin(selected_neighborhood)]
target_warsaw_venues.groupby('Neighborhood').count()
```

[33]:
```
                        Neighborhood Latitude  Neighborhood Longitude  Venue  \
Neighborhood
Grochów                                    24                      24     24
Natolin                                    27                      27     27
Powile                                     46                      46     46
Stara Praga                                35                      35     35
Stary Mokotów                              29                      29     29
Stary oliborz                              24                      24     24
ródmiecie Poudniowe                       100                     100    100
ródmiecie Pónocne                          88                      88     88

                        Venue Latitude  Venue Longitude  Venue Category
Neighborhood
Grochów                             24               24              24
Natolin                             27               27              27
Powile                              46               46              46
Stara Praga                         35               35              35
Stary Mokotów                       29               29              29
Stary oliborz                       24               24              24
ródmiecie Poudniowe                100              100             100
ródmiecie Pónocne                   88               88              88
```

[34]:
```python
warsaw_onehot = pd.get_dummies(target_warsaw_venues[['Venue Category']],␣
 ↪prefix="", prefix_sep="")
warsaw_onehot['Neighborhood'] = target_warsaw_venues['Neighborhood']

fixed_columns = [warsaw_onehot.columns[-1]] + list(warsaw_onehot.columns[:-1])
warsaw_onehot = warsaw_onehot[fixed_columns]

warsaw_onehot.head()
```

[34]:
```
      Neighborhood  American Restaurant  Art Gallery  Arts & Crafts Store  \
235  Stary Mokotów                    0            0                    0
236  Stary Mokotów                    0            0                    0
237  Stary Mokotów                    0            0                    0
238  Stary Mokotów                    0            0                    0
239  Stary Mokotów                    0            0                    0

     Asian Restaurant  Bakery  Bank  Bar  Beach Bar  Beer Bar  …  Tiki Bar  \
235                 0       0     0    0          0         0  …         0
```

```
236                   0          0      0     0           0          0  ...            0
237                   0          0      0     0           0          0  ...            0
238                   0          0      0     0           0          0  ...            0
239                   0          0      0     1           0          0  ...            0

     Turkish Restaurant  Vegetarian / Vegan Restaurant  Video Store  \
235                   0                              0            0
236                   0                              0            0
237                   0                              0            0
238                   0                              0            0
239                   0                              0            0

     Vietnamese Restaurant  Whisky Bar  Wine Bar  Wine Shop  Yoga Studio  \
235                      0           0         0          0            0
236                      0           0         0          0            0
237                      0           0         0          0            0
238                      0           0         0          0            0
239                      0           0         0          0            0

     Zoo Exhibit
235            0
236            0
237            0
238            0
239            0

[5 rows x 120 columns]
```

[35]:
```python
warsaw_grouped = warsaw_onehot.groupby('Neighborhood').mean().reset_index()
warsaw_grouped
```

[35]:
```
            Neighborhood  American Restaurant  Art Gallery  \
0                Grochów             0.000000     0.000000
1                Natolin             0.000000     0.000000
2                Powile             0.000000     0.000000
3            Stara Praga             0.000000     0.000000
4          Stary Mokotów             0.000000     0.000000
5          Stary oliborz             0.000000     0.000000
6  ródmiecie Poudniowe             0.000000     0.000000
7    ródmiecie Pónocne             0.011364     0.011364

   Arts & Crafts Store  Asian Restaurant    Bakery      Bank       Bar  \
0             0.000000          0.000000  0.000000  0.000000  0.000000
1             0.000000          0.000000  0.000000  0.000000  0.000000
2             0.000000          0.043478  0.021739  0.000000  0.043478
3             0.000000          0.000000  0.000000  0.000000  0.028571
4             0.000000          0.000000  0.103448  0.000000  0.034483
5             0.000000          0.041667  0.000000  0.000000  0.000000
```

```
6             0.000000          0.000000  0.020000  0.000000  0.020000
7             0.011364          0.011364  0.011364  0.011364  0.011364

   Beach Bar  Beer Bar  ...  Tiki Bar  Turkish Restaurant  \
0   0.000000  0.041667  ...      0.00            0.000000
1   0.000000  0.037037  ...      0.00            0.000000
2   0.021739  0.000000  ...      0.00            0.000000
3   0.000000  0.000000  ...      0.00            0.000000
4   0.000000  0.000000  ...      0.00            0.000000
5   0.000000  0.000000  ...      0.00            0.000000
6   0.000000  0.000000  ...      0.01            0.000000
7   0.000000  0.034091  ...      0.00            0.011364

   Vegetarian / Vegan Restaurant  Video Store  Vietnamese Restaurant  \
0                       0.000000     0.000000               0.000000
1                       0.000000     0.037037               0.000000
2                       0.000000     0.000000               0.021739
3                       0.028571     0.000000               0.000000
4                       0.034483     0.000000               0.034483
5                       0.000000     0.000000               0.000000
6                       0.080000     0.000000               0.010000
7                       0.022727     0.000000               0.000000

   Whisky Bar  Wine Bar  Wine Shop  Yoga Studio  Zoo Exhibit
0    0.000000  0.000000       0.00         0.00     0.000000
1    0.000000  0.000000       0.00         0.00     0.000000
2    0.000000  0.000000       0.00         0.00     0.000000
3    0.000000  0.000000       0.00         0.00     0.028571
4    0.000000  0.000000       0.00         0.00     0.000000
5    0.000000  0.041667       0.00         0.00     0.000000
6    0.000000  0.020000       0.01         0.01     0.000000
7    0.011364  0.011364       0.00         0.00     0.000000

[8 rows x 120 columns]
```

[36]:
```
num_top_venues = 10

for hood in warsaw_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = warsaw_grouped[warsaw_grouped['Neighborhood'] == hood].T.
 ↪reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
```

```
----Grochów----
----Natolin----
```

```
----Powile----
----Stara Praga----
----Stary Mokotów----
----Stary oliborz----
----ródmiecie Poudniowe----
----ródmiecie Pónocne----
```

[37]:
```
num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = warsaw_grouped['Neighborhood']

for ind in np.arange(warsaw_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] =␣
 ↪return_most_common_venues(warsaw_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted
```

[37]:
```
              Neighborhood 1st Most Common Venue  \
0                  Grochów                  Café
1                  Natolin     Sushi Restaurant
2                   Powile          Pizza Place
3              Stara Praga                 Diner
4            Stary Mokotów                Bakery
5             Stary oliborz                 Café
6   ródmiecie Poudniowe                   Café
7     ródmiecie Pónocne             Nightclub

           2nd Most Common Venue        3rd Most Common Venue  \
0                   Dessert Shop                  Bus Station
1                     Restaurant                         Park
2                           Café  Eastern European Restaurant
3                     Restaurant                        Hotel
4                           Café               Ice Cream Shop
5                Thai Restaurant           Polish Restaurant
6  Vegetarian / Vegan Restaurant                  Coffee Shop
7                    Coffee Shop                  Cocktail Bar
```

```
  4th Most Common Venue      5th Most Common Venue 6th Most Common Venue  \
0           Supermarket                Pizza Place  Fast Food Restaurant
1           Coffee Shop         Indian Restaurant    Italian Restaurant
2       Asian Restaurant                       Pub     Polish Restaurant
3           Coffee Shop  Middle Eastern Restaurant                  Road
4     Italian Restaurant         Convenience Store           Coffee Shop
5           Coffee Shop                      Plaza          Burger Joint
6          Cocktail Bar         Italian Restaurant     Sushi Restaurant
7                  Café                      Hotel    Italian Restaurant

  7th Most Common Venue 8th Most Common Venue 9th Most Common Venue  \
0           Flea Market            Restaurant           Coffee Shop
1        Sandwich Place                  Café     Convenience Store
2                   Bar    Italian Restaurant        Science Museum
3            Public Art                 Plaza                  Park
4          Dessert Shop           Pizza Place         Movie Theater
5            Restaurant            Public Art            Playground
6                Hostel                Bistro                 Plaza
7            Restaurant              Beer Bar     Polish Restaurant

      10th Most Common Venue
0         Mexican Restaurant
1      General Entertainment
2                 Restaurant
3              Movie Theater
4  Eastern European Restaurant
5             Breakfast Spot
6                      Hotel
7          Greek Restaurant
```

[38]:
```
from sklearn.cluster import KMeans
```

[39]:
```
kclusters = 4

warsaw_grouped_clustering = warsaw_grouped.drop('Neighborhood', 1)

kmeans = KMeans(n_clusters=kclusters, random_state=0).
↪fit(warsaw_grouped_clustering)
kmeans.labels_[0:10]
```

[39]:
```
array([3, 2, 1, 1, 0, 1, 1, 1])
```

[40]:
```
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

warsaw_merged = warsaw_data[warsaw_data['Neighborhood'].isin([hood for hood in␣
↪warsaw_grouped['Neighborhood']])]
warsaw_merged = warsaw_merged.join(neighborhoods_venues_sorted.
↪set_index('Neighborhood'), on='Neighborhood')
```

```
warsaw_merged # check the last columns!
```

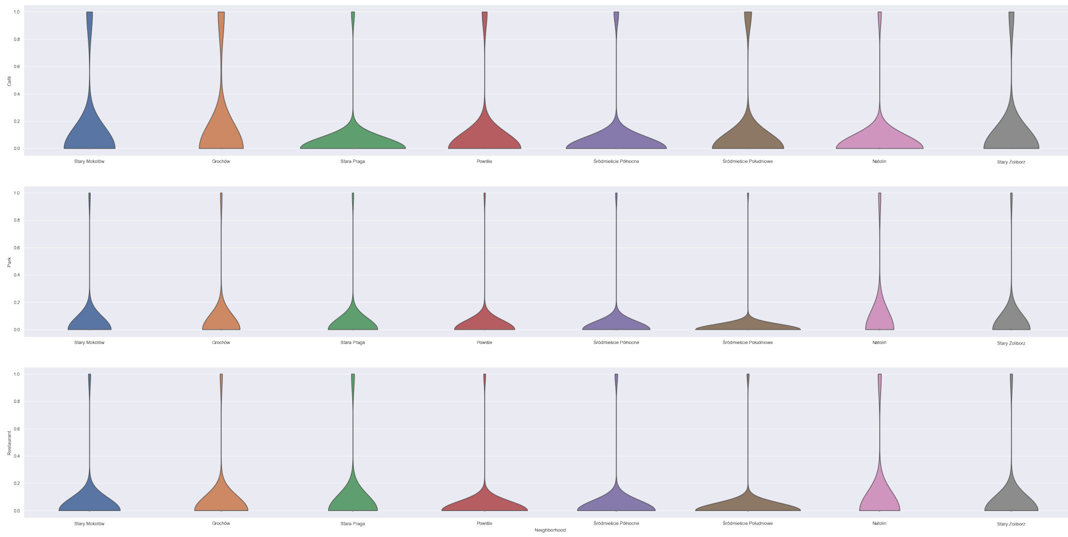
[40]:
```
          District            Neighborhood   Latitude  Longitude  \
43         Mokotów          Stary Mokotów  52.205272  21.011551
53   Praga-Poudnie                Grochów  52.246707  21.084637
59     Praga-Pónoc            Stara Praga  52.250981  21.033605
66      ródmiecie                  Powile  52.238055  21.029351
69      ródmiecie     ródmiecie Pónocne  52.236806  21.009433
70      ródmiecie   ródmiecie Poudniowe  52.222253  21.015700
89         Ursynów                Natolin  52.141101  21.056435
141       oliborz           Stary oliborz  52.266810  20.992990

     Cluster Labels 1st Most Common Venue          2nd Most Common Venue  \
43                0                Bakery                           Café
53                3                  Café                   Dessert Shop
59                1                 Diner                     Restaurant
66                1           Pizza Place                           Café
69                1             Nightclub                    Coffee Shop
70                1                  Café  Vegetarian / Vegan Restaurant
89                2     Sushi Restaurant                     Restaurant
141               1                  Café                Thai Restaurant

          3rd Most Common Venue 4th Most Common Venue  \
43               Ice Cream Shop    Italian Restaurant
53                  Bus Station           Supermarket
59                        Hotel           Coffee Shop
66   Eastern European Restaurant      Asian Restaurant
69                  Cocktail Bar                 Café
70                  Coffee Shop          Cocktail Bar
89                         Park           Coffee Shop
141           Polish Restaurant           Coffee Shop

        5th Most Common Venue 6th Most Common Venue 7th Most Common Venue  \
43          Convenience Store           Coffee Shop          Dessert Shop
53                Pizza Place  Fast Food Restaurant           Flea Market
59   Middle Eastern Restaurant                  Road            Public Art
66                        Pub     Polish Restaurant                   Bar
69                      Hotel    Italian Restaurant            Restaurant
70         Italian Restaurant      Sushi Restaurant                Hostel
89          Indian Restaurant    Italian Restaurant        Sandwich Place
141                     Plaza          Burger Joint            Restaurant

    8th Most Common Venue 9th Most Common Venue       10th Most Common Venue
43            Pizza Place         Movie Theater  Eastern European Restaurant
53             Restaurant           Coffee Shop           Mexican Restaurant
59                  Plaza                  Park                Movie Theater
```

```
66      Italian Restaurant        Science Museum                  Restaurant
69                Beer Bar    Polish Restaurant          Greek Restaurant
70                  Bistro                Plaza                       Hotel
89                    Café    Convenience Store     General Entertainment
141             Public Art           Playground             Breakfast Spot
```

1.5 Analysis

[41]:
```
fig = plt.figure(figsize=(50,25))
sns.set(font_scale=1.1)

ax = plt.subplot(3,1,1)
sns.violinplot(x="Neighborhood", y="Café", data=warsaw_onehot, cut=0);
plt.xlabel("")

ax = plt.subplot(3,1,2)
sns.violinplot(x="Neighborhood", y="Park", data=warsaw_onehot, cut=0);
plt.xlabel("")

# plt.subplot(4,1,3)
# sns.violinplot(x="Neighborhood", y="Pizza Place", data=warsaw_onehot, cut=0);

plt.subplot(3,1,3)
sns.violinplot(x="Neighborhood", y="Restaurant", data=warsaw_onehot, cut=0);

ax.text(-1.0, 3.1, 'Frequency distribution for the top 3 venue categories for␣
↪each neighborhood', fontsize=60)
plt.savefig ("Distribution_Frequency_Venues_3_categories_clothing.png", dpi=240)
plt.show()
```

Frequency distribution for the top 3 venue categories for each neighborhood



```
[42]: import matplotlib.cm as cm
      import matplotlib.colors as colors
```

```
[47]: # create map
      map_clusters = folium.Map(location=[latitude, longitude], zoom_start=12)

      # set color scheme for the clusters
      x = np.arange(kclusters)
      ys = [i + x + (i*x)**2 for i in range(kclusters)]
      colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
      rainbow = [colors.rgb2hex(i) for i in colors_array]

      # add markers to the map
      markers_colors = []
      for lat, lon, poi, cluster in zip(warsaw_merged['Latitude'], warsaw_merged['Longitude'], warsaw_merged['Neighborhood'], warsaw_merged['Cluster Labels']):
          label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)

          if np.isnan(cluster):
              cluster = np.nan_to_num(cluster)

          folium.CircleMarker(
              [lat, lon],
              radius=30,
              popup=label,
              color=rainbow[int(cluster)-1],
```

```
        fill=True,
        fill_color=rainbow[int(cluster)-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

[47]: <folium.folium.Map at 0x1d313e1d128>

Examine Cluster 0

[44]:
```
warsaw_merged.loc[warsaw_merged['Cluster Labels'] == 0, warsaw_merged.
 ↪columns[[1] + list(range(5, warsaw_merged.shape[1]))]]
```

[44]:
```
     Neighborhood 1st Most Common Venue 2nd Most Common Venue  \
43  Stary Mokotów                Bakery                  Café

   3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
43        Ice Cream Shop    Italian Restaurant     Convenience Store

   6th Most Common Venue 7th Most Common Venue 8th Most Common Venue  \
43           Coffee Shop          Dessert Shop           Pizza Place

   9th Most Common Venue       10th Most Common Venue
43         Movie Theater  Eastern European Restaurant
```

Examine Cluster 1

[45]:
```
warsaw_merged.loc[warsaw_merged['Cluster Labels'] == 1, warsaw_merged.
 ↪columns[[1] + list(range(5, warsaw_merged.shape[1]))]]
```

[45]:
```
               Neighborhood 1st Most Common Venue  \
59              Stara Praga                 Diner
66                   Powile           Pizza Place
69      ródmiecie Pónocne             Nightclub
70    ródmiecie Poudniowe                  Café
141           Stary oliborz                  Café

             2nd Most Common Venue        3rd Most Common Venue  \
59                      Restaurant                        Hotel
66                            Café  Eastern European Restaurant
69                     Coffee Shop                  Cocktail Bar
70    Vegetarian / Vegan Restaurant                  Coffee Shop
141                Thai Restaurant            Polish Restaurant

    4th Most Common Venue      5th Most Common Venue 6th Most Common Venue  \
59            Coffee Shop  Middle Eastern Restaurant                  Road
66       Asian Restaurant                        Pub     Polish Restaurant
69                   Café                      Hotel    Italian Restaurant
70           Cocktail Bar         Italian Restaurant       Sushi Restaurant
```

```
141           Coffee Shop                          Plaza          Burger Joint

    7th Most Common Venue 8th Most Common Venue 9th Most Common Venue  \
59             Public Art                 Plaza                  Park
66                    Bar   Italian Restaurant        Science Museum
69             Restaurant              Beer Bar     Polish Restaurant
70                 Hostel                Bistro                 Plaza
141            Restaurant            Public Art            Playground

    10th Most Common Venue
59           Movie Theater
66              Restaurant
69        Greek Restaurant
70                   Hotel
141         Breakfast Spot
```

Examine Cluster 2

[48]:
```python
warsaw_merged.loc[warsaw_merged['Cluster Labels'] == 2, warsaw_merged.
 ↪columns[[1] + list(range(5, warsaw_merged.shape[1]))]]
```

[48]:
```
   Neighborhood 1st Most Common Venue 2nd Most Common Venue  \
89      Natolin      Sushi Restaurant            Restaurant

   3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
89                  Park           Coffee Shop     Indian Restaurant

   6th Most Common Venue 7th Most Common Venue 8th Most Common Venue  \
89    Italian Restaurant        Sandwich Place                  Café

   9th Most Common Venue 10th Most Common Venue
89     Convenience Store  General Entertainment
```

Examine Cluster 3

[49]:
```python
warsaw_merged.loc[warsaw_merged['Cluster Labels'] == 3, warsaw_merged.
 ↪columns[[1] + list(range(5, warsaw_merged.shape[1]))]]
```

[49]:
```
   Neighborhood 1st Most Common Venue 2nd Most Common Venue  \
53      Grochów                  Café          Dessert Shop

   3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
53           Bus Station           Supermarket           Pizza Place

   6th Most Common Venue 7th Most Common Venue 8th Most Common Venue  \
53  Fast Food Restaurant           Flea Market            Restaurant

   9th Most Common Venue 10th Most Common Venue
53           Coffee Shop     Mexican Restaurant
```

1.6 Results and Discussion

I think it is no surprise that all these districts are very centrally located in the circular layout of Warsaw. Locations meeting the criteria of popular places would usually be in central locations in many cities around the world. From this visualization it is clear that on a practical level, without data on the basis of which decisions could be made, the circle of 103 locations is very large. We have significantly narrowed the search area from 8 potential districts to 5, which should respond to the business problem.

We have drawn conclusions from the data, creating location recommendations, but that's the point. There is no right or wrong answer or conclusion for the task. The task of analyzing the data here is to guide the course of selecting the location of the apartment to narrow the search to only a few main areas that best fit the criteria.

Moreover, FourSquare is not popular in Warsaw, the data maybe out-dated or unreliable, the report should gather more data from other location data source such as Google Place API.

1.7 Conclusion

Different applications of this analysis are available based on a different methodology and possibly different data sources. The stakeholder problem has been resolved. The stakeholder wants to find the best place to live in Warsaw, and the "best location" factors are based on the number of places in the food, cafe and park category around the location. Machine learning technique based on content filtering is the most appropriate method to solve the problem. Eight destination locations may not be a good choice, but I can quickly choose other locations and issue a recommendation again.

[ ]:

```

```