Coursera - IBM

Applied Data Science Capstone Project

The Battle of Neighborhoods in Munich, Germany

by Xingyuan Tan

1. Introduction

Munich is the capital and most populous city of Bavaria. With a population of 1,558,395 inhabitants as of July 31, 2020, it is the third-largest city in Germany, after Berlin and Hamburg, and thus the largest which does not constitute its own state, as well as the 11th-largest city in the European Union. The city's metropolitan region is home to 6 million people.
Today, Munich is a global centre of art, science, technology, finance, publishing, culture, innovation, education, business, and tourism and enjoys a very high standard and quality of living, reaching first in Germany and third worldwide according to the 2018 Mercer survey,and being rated the world's most liveable city by the Monocle's Quality of Life Survey 2018. According to the Globalization and World Rankings Research Institute, Munich is considered an alpha-world city, as of 2015.It is one of the most prosperous and fastest growing cities in Germany.

2. Business Problem

As Munich is becoming a very attractive city to live in, one individual wants to start his own business in the city by opening a restaurant. However the city is too big for him to decide which neighborhood would be the best choice for a start-up.

3.Data

The postal code data is coming from Muechen.de, being published via https://www.muenchen.de/int/en/living/postal-codes.html; and the geo data is retrieving from GeoNames.org and being stored in git-hub for further usage. Since two data contains different part of data, the combination is being done in the project.

3.1 Neighborhoods Data

```
url = 'https://www.muenchen.de/int/en/living/postal-codes.html'
munich_post_list = pd.read_html(url)
munich_post = munich_post_list[0]
munich_post
```

	District	Postal Code
0	Allach-Untermenzing	80995, 80997, 80999, 81247, 81249
1	Altstadt-Lehel	80331, 80333, 80335, 80336, 80469, 80538, 80539
2	Au-Haidhausen	81541, 81543, 81667, 81669, 81671, 81675, 81677
3	Aubing-Lochhausen-Langwied	81243, 81245, 81249
4	Berg am Laim	81671, 81673, 81735, 81825
5	Bogenhausen	81675, 81677, 81679, 81925, 81927, 81929
6	Feldmoching-Hasenbergl	80933, 80935, 80995
7	Hadern	80689, 81375, 81377
8	Laim	80686, 80687, 80689
9	Ludwigsvorstadt-Isarvorstadt	80335, 80336, 80337, 80469
10	Maxvorstadt	80333, 80335, 80539, 80636, 80797, 80798, 8079...
11	Milbertshofen-Am Hart	80807, 80809, 80937, 80939

3.2 Geo Data (from GeoNames.Org)

Get PostalCode and Geoinfo

```
df = pd.read_csv('https://raw.githubusercontent.com/zauberware/postal-codes-json-xml-csv/master/data/DE/zipcodes.de.csv') #get
ing geocode from mentioned csv
df.head()
```

	country_code	zipcode	place	state	state_code	province	province_code	community	community_code	latitude	longitude
0	DE	1945	Grünewald	Brandenburg	BB	NaN	0	Landkreis Oberspreewald-Lausitz	12066	51.4000	14.0000
1	DE	1945	Lindenau	Brandenburg	BB	NaN	0	Landkreis Oberspreewald-Lausitz	12066	51.4000	13.7333
2	DE	1945	Hohenbocka	Brandenburg	BB	NaN	0	Landkreis Oberspreewald-Lausitz	12066	51.4310	14.0098
3	DE	1945	Schwarzbach	Brandenburg	BB	NaN	0	Landkreis Oberspreewald-Lausitz	12066	51.4500	13.9333
4	DE	1945	Guteborn	Brandenburg	BB	NaN	0	Landkreis Oberspreewald-Lausitz	12066	51.4167	13.9333

```
munich = df[df['community'] == 'München'].reset_index(drop=True)
munich.head()
```

	country_code	zipcode	place	state	state_code	province	province_code	community	community_code	latitude	longitude
0	DE	80331	München	Bayern	BY	Upper Bavaria	91	München	9162	48.1345	11.5710
1	DE	80333	München	Bayern	BY	Upper Bavaria	91	München	9162	48.1452	11.5668
2	DE	80335	München	Bayern	BY	Upper Bavaria	91	München	9162	48.1427	11.5552
3	DE	80336	München	Bayern	BY	Upper Bavaria	91	München	9162	48.1345	11.5590
4	DE	80337	München	Bayern	BY	Upper Bavaria	91	München	9162	48.1224	11.5449

3.3 Data Combination

```
latitudes = [] # Initializing the latitude array
longitudes = [] # Initializing the longitude array
postal_codes = muc['Postal Code']
```

```
#Mapping postal code between two data frames
for postal_code in postal_codes :
    lat = df[df['zipcode']== int(postal_code)]['latitude'].iloc[0]
    lon = df[df['zipcode'] == int(postal_code)]['longitude'].iloc[0]

    latitudes.append(lat)
    longitudes.append(lon)
```

```
muc['Latitude'] = latitudes
muc['Longitude'] = longitudes
```

```
muc.head()
```

	District	Postal Code	Latitude	Longitude
0	Allach-Untermenzing	80995	48.1976	11.5181
1	Allach-Untermenzing	80997	48.1834	11.4784
2	Allach-Untermenzing	80999	48.1853	11.4643
3	Allach-Untermenzing	81247	48.1662	11.4673
4	Allach-Untermenzing	81249	48.1500	11.5833

3.4 Venue Data

```
print(muc_venues.shape)
muc_venues.head()
```

```
(2414, 7)
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Allach-Untermenzing	48.1976	11.5181	Indian Village	48.197932	11.524083	Indian Restaurant
1	Allach-Untermenzing	48.1976	11.5181	EDEKA Fratzl	48.197393	11.524481	Supermarket
2	Allach-Untermenzing	48.1976	11.5181	Pizza Fasanerie	48.197816	11.524199	Pizza Place
3	Allach-Untermenzing	48.1976	11.5181	H Fasanerie Bahnhof	48.196895	11.524417	Bus Stop
4	Allach-Untermenzing	48.1834	11.4784	NORMA	48.183488	11.478840	Supermarket

4. Methodology

4.1 Folium

```
# Munich latitude and longitude using Google search
muc_lat = 48.69668
muc_lng = 13.46314

# Creates map of Munich using latitude and longitude values
map_munich = folium.Map(location=[muc_lat, muc_lng], zoom_start=10)

# Add markers to map
for lat, lng, borough in zip(muc['Latitude'], muc['Longitude'], muc['District']):
    label = '{}'.format( borough)
    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_munich)

map_munich
```

4.2 OneHot Encoding

```
# one hot encoding
muc_onehot = pd.get_dummies(muc_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
muc_onehot['Neighborhood'] = muc_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [muc_onehot.columns[-1]] + list(muc_onehot.columns[:-1])
muc_onehot = muc_onehot[fixed_columns]

muc_onehot.head()
```

4.3 K-Means Clustering

```
# set number of clusters
kclusters = 5

muc_grouped_clustering = muc_grouped.drop('Neighborhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(muc_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

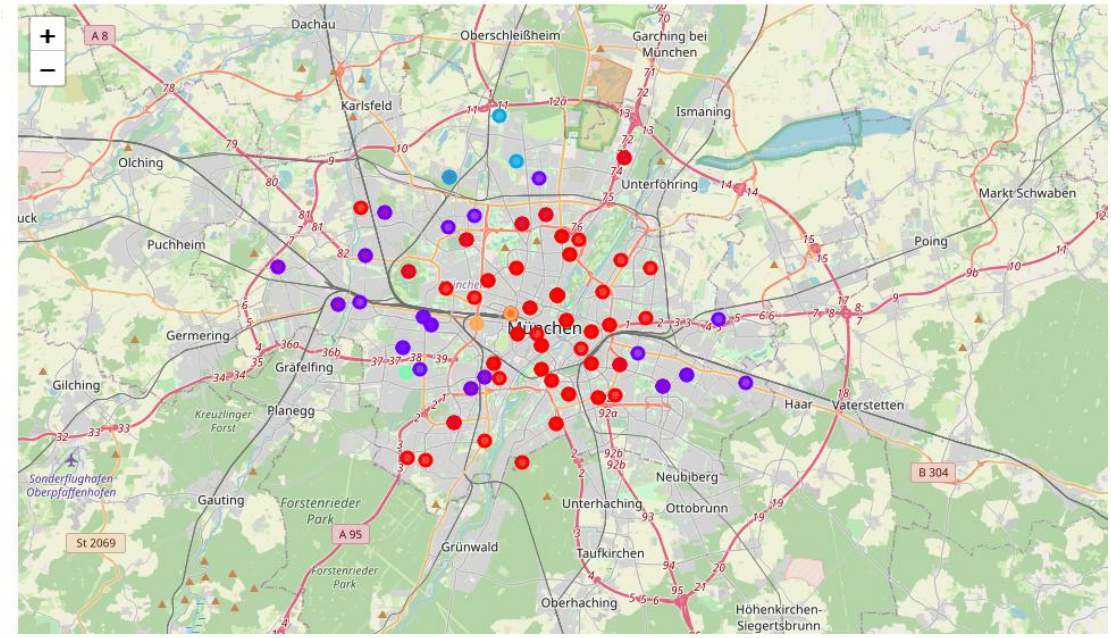
```
array([0, 0, 0, 0, 1, 0, 2, 3, 1, 0], dtype=int32)
```

```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels00', kmeans.labels_)

muc_merged = muc

# merge manhattan_grouped with manhattan_data to add latitude/longitude for each neighborhood
muc_merged = muc_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='District')

muc_merged.head() # check the last columns!
```

5. Results

The neighborhoods are divided into n clusters where n is the number of clusters found using the optimal approach. The clustered neighborhoods are visualized using different colors so as to make them distinguishable.

```
map_clusters
```



6. Discussion

The result shows that in the Altstadt (old city) of Munich, the most common venues are ice-cream shops, musemums, and Restaurant, which reflects the reality that it is currently one of the most popular tourist place in the city. In cluster2, the most common venues changes from restaurant to Supermarket, which indicates that those are more liveing areas for the citizens. Outer thatn cluster2, the most common venues change to Bus Stop and Hotel, which gives the sign that those places are where most companies are located so there're more visits in Bus Stop (for working) and Hotel (business trips). The analyse around München Haupbahnhof isn't very well as expected though, which could be due to less of information and limited area.
The critical part of the project is where to find the geo data and how to combine different source of data together for further analysis.

7. Conclusion

So now the neighborhoods in Munich are being clustered. For users who want to open a restaurant, it is better to open it in the inner city or near to the living communities, where the potential visits could be higher due to tourism and the citizen people. However, the analysis is only based on single source, there may have other factors which could affect the decision as well.

Cluster 1

```
muc_merged.loc[muc_merged['Cluster Labels00'] == 0, muc_merged.columns[[1] + list(range(5, muc_merged.shape[1]))]]
```

	Postal Code	Cluster Labels01	Cluster Labels	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	80995	0	0	Italian Restaurant	Café	Bakery	Bus Stop	Supermarket	Bar	Sporting Goods Shop	German Restaurant	Plaza	Ice Cream Shop
1	80997	0	0	Italian Restaurant	Café	Bakery	Bus Stop	Supermarket	Bar	Sporting Goods Shop	German Restaurant	Plaza	Ice Cream Shop
2	80999	0	0	Italian Restaurant	Café	Bakery	Bus Stop	Supermarket	Bar	Sporting Goods Shop	German Restaurant	Plaza	Ice Cream Shop
3	81247	0	0	Italian Restaurant	Café	Bakery	Bus Stop	Supermarket	Bar	Sporting Goods Shop	German Restaurant	Plaza	Ice Cream Shop
4	81249	0	0	Italian Restaurant	Café	Bakery	Bus Stop	Supermarket	Bar	Sporting Goods Shop	German Restaurant	Plaza	Ice Cream Shop
5	80331	0	0	Café	Hotel	Italian Restaurant	Plaza	Bar	Middle Eastern Restaurant	Cocktail Bar	Art Museum	Burger Joint	Restaurant

Cluster 2

```
muc_merged.loc[muc_merged['Cluster Labels00'] == 1, muc_merged.columns[[1] + list(range(5, muc_merged.shape[1]))]]
```

	Postal Code	Cluster Labels01	Cluster Labels	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
22	81671	1	1	Supermarket	Bakery	Drugstore	Hotel	Metro Station	Bus Stop	Asian Restaurant	Italian Restaurant	Soccer Field	Bavarian Restaurant
23	81673	1	1	Supermarket	Bakery	Drugstore	Hotel	Metro Station	Bus Stop	Asian Restaurant	Italian Restaurant	Soccer Field	Bavarian Restaurant
24	81735	1	1	Supermarket	Bakery	Drugstore	Hotel	Metro Station	Bus Stop	Asian Restaurant	Italian Restaurant	Soccer Field	Bavarian Restaurant
25	81825	1	1	Supermarket	Bakery	Drugstore	Hotel	Metro Station	Bus Stop	Asian Restaurant	Italian Restaurant	Soccer Field	Bavarian Restaurant

Cluster 3

```
muc_merged.loc[muc_merged['Cluster Labels00'] == 2, muc_merged.columns[[1] + list(range(5, muc_merged.shape[1]))]]
```

	Postal Code	Cluster Labels01	Cluster Labels	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
32	80933	2	2	Bus Stop	Supermarket	Indian Restaurant	Pizza Place	Gastropub	Business Service	Food & Drink Shop	Beer Garden	Bakery	Greek Restaurant
33	80935	2	2	Bus Stop	Supermarket	Indian Restaurant	Pizza Place	Gastropub	Business Service	Food & Drink Shop	Beer Garden	Bakery	Greek Restaurant
34	80995	2	2	Bus Stop	Supermarket	Indian Restaurant	Pizza Place	Gastropub	Business Service	Food & Drink Shop	Beer Garden	Bakery	Greek Restaurant