Week 7 Report

A Clustering Model Based on Japan's housing transaction data

37237256 薛丁銘

1. Background

Similar to last week, I decide to do a cluster model by myself. This time I have got a dataset of Japan's housing transaction. Then I clustered the transaction into 10 clusters, showing the 10 most frequent housing transaction areas in Japan.

2. Dataset

Two datasets are involved in this task. The first one is the detailed housing transaction data in Japan for the last four month with the machi code of the houses as shown in Figure 2.1. In order to cluster the locations, another dataset with latitudes/longitudes of Japan machi code was prepared as shown in Figure 2.2.

```
No,種類,地域,市区町村コード,都道府県名,市区町村名,地区名,最寄駅：名称,最寄駅：距離（分）,取引価格（総額）,坪単価,間取り,面積（㎡）,取引価格（㎡単価）,
112922,宅地(土地),住宅地,8443,茨城県,稲敷郡阿見町,本郷,荒川沖,18,12000000,140000.0,,280,43000.0,長方形,20.3,,,,,,西,町道,6.0,第１種中高
81610,宅地(土地と建物),住宅地,14153,神奈川県,相模原市南区,上鶴間,東林間,8,28000000,,,200,,ほぼ台形,13.3,180,平成15年,木造,住宅,,西,市道,15.
13427,宅地(土地と建物),住宅地,41202,佐賀県,唐津市,和多田大土井,鬼塚,11,3000000,,,200,,ほぼ長方形,13.0,100,昭和56年,軽量鉄骨造,住宅,,北東,私
216941,宅地(土地),住宅地,11222,埼玉県,越谷市,相模町,三郷(埼玉),9,31000000,480000.0,,210,150000.0,長方形,8.0,,,,,,南,区画街路,6.0,第１種
253716,宅地(土地),住宅地,14205,神奈川県,藤沢市,鵠沼石上,石上,3,29000000,740000.0,,130,220000.0,ほぼ整形,,,,,,,北西,市道,4.0,第１種住居地
137608,中古マンション等,,27128,大阪府,大阪市中央区,常盤町,谷町四丁目,4,25000000,,１ＬＤＫ,30,,,,,平成29年,ＲＣ,住宅,住宅,,,,商業地域,80.0,800
145105,林地,,28219,兵庫県,三田市,藍本,,,780000,,,5000㎡以上,,,,,,,,,,,,,,,2014年第３四半期,,
46832,農地,,46217,鹿児島県,曽於市,末吉町諏訪方,,,870000,,,2900,,,,,,,,,,,,,,,2017年第１四半期,,
86187,農地,,4212,宮城県,登米市,米山町,,,200000,,,1000,,,,,,,,,,,,,,,2012年第１四半期,,
8401,中古マンション等,,26106,京都府,京都市下京区,醒泉学区,五条(京都市営),10,18000000,,２ＬＤＫ,50,,,,,平成5年,ＳＲＣ,住宅,住宅,,,,商業地域,80.
128579,宅地(土地と建物),住宅地,13108,東京都,江東区,白河,清澄白河,2,26000000,,,40,,ほぼ長方形,5.5,105,昭和55年,鉄骨造,住宅,,東,区道,6.0,商業
106157,農地,,40213,福岡県,行橋市,大字袋迫,,,1200000,,,165,,,,,,,,,,,,,,,2010年第４四半期,,
40974,宅地(土地と建物),住宅地,27143,大阪府,堺市東区,日置荘西町,初芝,6,7000000,,,65,,ほぼ長方形,6.7,110,昭和54年,木造,住宅、駐車場,住宅,南西,
213873,林地,,1453,北海道,上川郡東神楽町,字八千代ケ岡,,,1600000,,,5000㎡以上,,,,,,,,,,,,,,,2017年第４四半期,,
199969,宅地(土地),住宅地,28202,兵庫県,尼崎市,戸ノ内町,園田,26,32000000,490000.0,,210,150000.0,ほぼ長方形,15.4,,,,,,北,市道,8.0,第１種住
19856,宅地(土地),宅地見込地,2203,青森県,八戸市,大字市川町,陸奥市川,1H?1H30,3000000,34000.0,,300,10000.0,ほぼ整形,,,,,,,接面道路無,,,第１
39614,宅地(土地),住宅地,22214,静岡県,藤枝市,藤枝,藤枝,30分?60分,1200000,4400.0,,900,1300.0,ほぼ整形,28.0,,,,,倉庫,南,私道,3.0,第１種住居
20254,林地,,9213,栃木県,那須塩原市,石林,,,3000000,,,1000,,,,,,,,,,,,,,,2020年第４四半期,,
22035,宅地(土地と建物),住宅地,13106,東京都,台東区,浅草,浅草(つくばＥＸＰ),9,65000000,,,80,,台形,5.7,85,平成21年,木造,住宅,住宅,南東,私道,3.0,
507012,宅地(土地と建物),住宅地,13211,東京都,小平市,花小金井南町,花小金井,8,30000000,,,195,,不整形,7.2,85,平成14年,木造,住宅,,北,市道,4.0,第
59867,宅地(土地と建物),住宅地,26211,京都府,京田辺市,花住坂,松井山手,12,35000000,,,175,,長方形,11.0,125,昭和64年,軽量鉄骨造,住宅,,南,市道,6.
44508,中古マンション等,,43101,熊本県,熊本市中央区,新町,上熊本(ＪＲ・熊本電鉄),22,22000000,,３ＬＤＫ,80,,,,,平成12年,ＳＲＣ,住宅,住宅,,,,商業地域
109366,中古マンション等,,27102,大阪府,大阪市都島区,友渕町,都島,18,17000000,,２ＬＤＫ,70,,,,,平成4年,ＳＲＣ,住宅,,,,,第１種住居地域,80.0,200.0
13210,宅地(土地),住宅地,9205,栃木県,鹿沼市,花岡町,新鹿沼,7,8000000,120000.0,,220,36000.0,長方形,13.0,,,,,,北,区画街路,6.0,第２種中高層住
64175,宅地(土地),住宅地,40107,福岡県,北九州市小倉南区,朽網東,朽網,10,7000000,71000.0,,320,22000.0,袋地等,,,,,,住宅,南西,市道,2.4,第１種低
39791,宅地(土地と建物),住宅地,7204,福島県,いわき市,中央台高久,いわき,1H30?2H,31000000,,,280,,長方形,14.0,155,平成24年,軽量鉄骨造,住宅,,西,市
82972,宅地(土地と建物),商業地,20205,長野県,飯田市,上殿岡,切石,28,120000000,,,930,,ほぼ台形,19.0,165,平成6年,鉄骨造,その他,,南東,市道,5.0,近
11617,農地,,15108,新潟県,新潟市西蒲区,越前浜,,,120000,,,450,,,,,,,,,,,,,,,2022年第１四半期,,
195564,林地,,23221,愛知県,新城市,作手高里,,,4500000,,,2700,,,,,,,,,,,,,,,2015年第２四半期,,
10666,農地,,24204,三重県,松阪市,伊勢寺町,,,300000,,,910,,,,,,,,,,,,,,,2021年第１四半期,,
```

Figure 2.1 The Housing transaction data

```
コード,都道府県,市区町村,市区町村2,緯度,経度
11011,北海道,札幌市,中央区,43.05546,141.340956
11029,北海道,札幌市,北区,43.09085,141.340831
11037,北海道,札幌市,東区,43.076069,141.363722
11045,北海道,札幌市,白石区,43.047687,141.405078
11053,北海道,札幌市,豊平区,43.031291,141.380106
11061,北海道,札幌市,南区,42.990031,141.353497
11070,北海道,札幌市,西区,43.07447,141.300889
11088,北海道,札幌市,厚別区,43.036408,141.474789
11096,北海道,札幌市,手稲区,43.121944,141.245632
11100,北海道,札幌市,清田区,42.999636,141.44383
```

Figure 2.2 The latitude/longitude data

3. Clustering Model

In the first part I conducted the data preprocess including loading the data and merge the transaction data with the location data. There was something wrong with the length of the machi code so I had to convert the formation. The preprocess code is as shown in Figure 3.1.

```
recommendation.py U    region_clustering.py ×
school > Introduction to Machine Learning > Week07_exercise > region_clustering.py > ...

# Reading house price data and latitude_longitude data
house_pricing_data = pd.read_csv("school\Introduction to Machine Learning\Week07_exercise\data\house_pricing_data.csv")
latitude_longitude = pd.read_csv("school\Introduction to Machine Learning\Week07_exercise\data\latitude_longitude.csv")
# print(house_pricing_data.columns)
# print(latitude_longitude.columns)


house_machi_code = pd.DataFrame(house_pricing_data['市区町村コード'])
latitude_longitude_reference = latitude_longitude[['コード', '緯度', '経度']]

house_machi_code.dropna(inplace = True)
latitude_longitude_reference.dropna(inplace = True)

datacheck_house_machi_code = pd.DataFrame(house_machi_code).map(lambda x : len(str(x)))
datacheck_latitude_longitude_reference = pd.DataFrame(latitude_longitude_reference['コード']).map(lambda x : len(str(x)))

print(datacheck_house_machi_code.value_counts())
print(datacheck_latitude_longitude_reference.value_counts())

code2lng = {}
code2lat = {}
for i in range(len(latitude_longitude_reference)):
    code = str(latitude_longitude_reference['コード'].iloc[i])
    if (len(code) == 5):
        code = code[0:4]
    else:
        code = code[0:5]
    code2lng.update({int(code) : latitude_longitude_reference["経度"].iloc[i]})
    code2lat.update({int(code) : latitude_longitude_reference["緯度"].iloc[i]})

house_machi_code['latitude'] = house_machi_code['市区町村コード'].map(code2lat)
house_machi_code['longitude'] = house_machi_code['市区町村コード'].map(code2lng)

house_location = []
for i in range(len(house_machi_code)):
    lat = house_machi_code.iloc[i, 1]
    long = house_machi_code.iloc[i, 2]
    house_location.append((lat, long))

print(len(house_location))

# # plot the house positions
# house_machi_code.plot(x = 'latitude', y = 'longitude', kind = 'scatter', label = 'House location', marker = '+', color = 'black')
# plt.title("Scatterplot of house locations")
# plt.xlabel("Latitude")
# plt.ylabel("Longitude")
# plt.show()

location_array = house_machi_code[['latitude', 'longitude']].to_numpy()

```

Figure 3.1 Preprocess of the data

After the preprocessing, I drew I map of the transaction data. The distribution of the transaction data looks like the map of Japan as shown in Figure 3.2. Then I used kmeans to cluster the housing transaction data into 10 clusters. I also drew another map of the clustering result with different colors. The result map is shown in Figure 3.3 and the clustering code is shown in Figure 3.4.

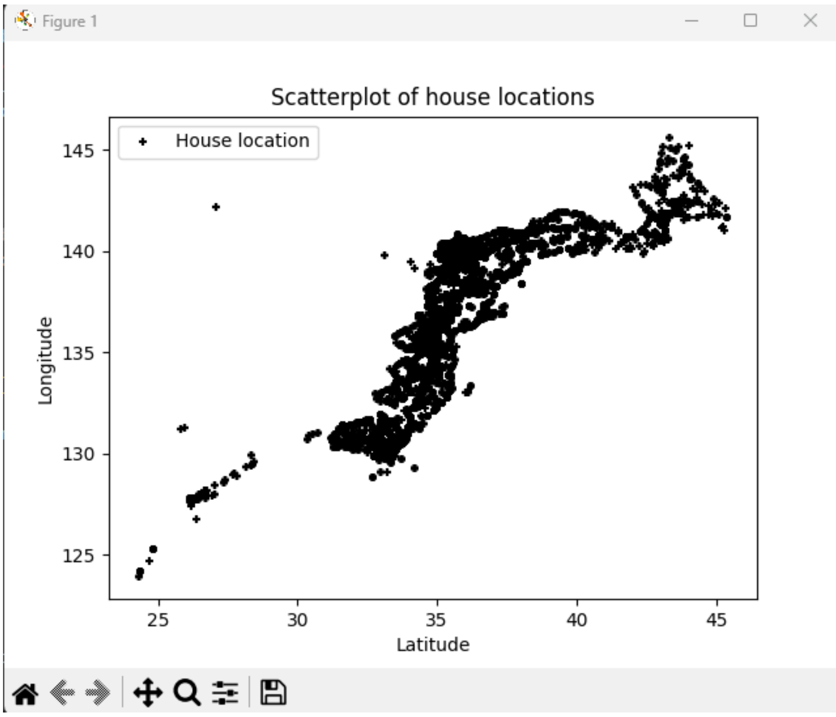

Figure 3.2 Shape of the transaction data

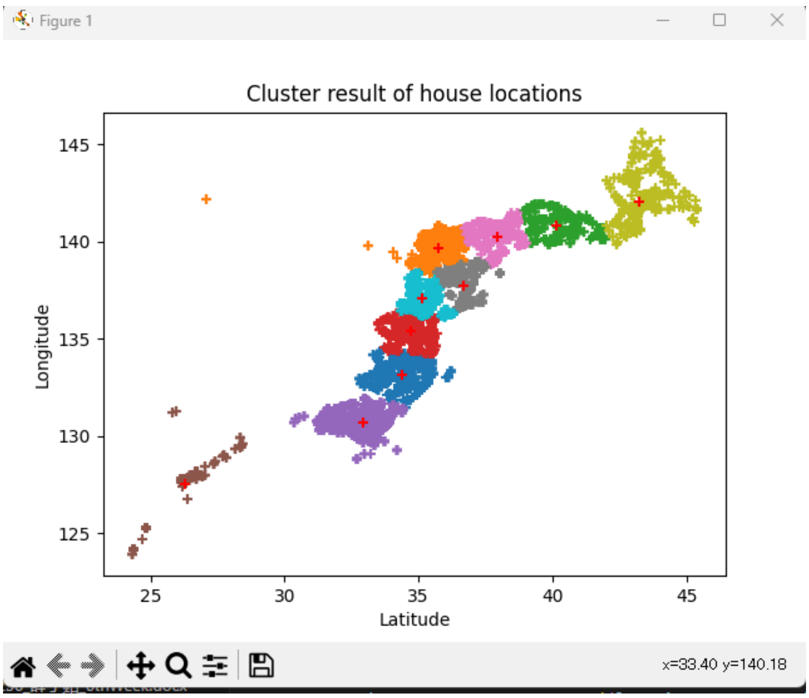

Figure 3.3 Shape of the transaction data

```python
kmeans = KMeans(n_clusters=10, random_state = 42)
kmeans.fit(location_array)

centers = kmeans.cluster_centers_
print(centers)

predicted_labels = kmeans.labels_
print(len(predicted_labels))

merged_array = np.concatenate((location_array, predicted_labels.reshape(-1, 1)), axis =1)

print(merged_array)

labels = np.unique(merged_array[:, 2])
split_house_array = [merged_array[merged_array[:,2] == label] for label in labels]
label_count = 0
for array in split_house_array:
    plt.scatter(x = array[:,0], y = array[:,1], label = 'house cluster ' + str(label_count), marker = '+')
    label_count += 1
plt.scatter(x = centers[:,0], y = centers[:,1], label = 'house location', marker = '+', color = 'red')
plt.title("Cluster result of house locations")
plt.xlabel("Latitude")
plt.ylabel("Longitude")
plt.show()
```

Figure 3.4 The clustering code

Finaly I ran the evaluation code as shown in Figure 3.5 and the results is shown in Figure 3.6. There are no other cluster methods for comparing so the clustering evaluation is just for reference. The evaluation took longer time than I expected. I guess it was because the dataset was too large.

```python
## Evaluation
silhousette = metrics.silhouette_score(location_array, predicted_labels)
print("Silhousette Score: ", silhousette)

calinski = metrics.calinski_harabasz_score(location_array, predicted_labels)
print("Calinski-Harabasz: ", calinski)

Davies = metrics.davies_bouldin_score(location_array, predicted_labels)
print("Davies-Bouldin: ", Davies)
```

Figure 3.5 The evaluation code

```
Silhousette Score:  0.5978859918214449
Calinski-Harabasz:  362853.02294252854
Davies-Bouldin:  0.6126386250450064
PS D:\code>
```

Figure 3.6 The evaluation result