Applied Data Science Capstone

The Battle of Neighborhoods

Opening convenience store in Japan’s Prefectures

Rita

Contents

Introduction 2
Data 2
Methodology 2
Step 1: Data Collection and preparation – List of Japanese prefectures 2
Step 2: Data Collection and preparation – The longitude and latitude of each prefectures 4
Step 3: Visualization the neighborhoods of Japan 5
Step 4: Data Collection and Preparation – Venue and Venue Categories in Japan 5
Step 5: One Hot Encoding the venue categories 6
Step 6: Find out top 10 Venues in the neighborhoods 8
Step 7: Using K-Means model for clustering 8
Result 9
Cluster 1 10
Cluster 2 11
Cluster 3 11
Cluster 4 11
Cluster 5 12
Discussion 12
Conclusion 12
References 13

Introduction

A convenience store is a small retail shop that sells a range of daily item, such as coffee, cup noodles, snack foods, newspaper and magazines. It is very easy to find a convenience store in Asian countries and it is unsurprised that you can find a convenience store in a lot of streets in Japan. Apart from the Japanese, the tourist love going to convenience store to buy the food and drinks. In 2019, the contribution of travel and tourism to GDP for Japan was 7.5 % with over 31 million people travelled to Japan and 8.7% increased by comparing with to 2018.

With a growth of the number of tourist and the needs of the citizens in Japan, the local entrepreneurs should carefully consider which prefectures would be beneficial to open some new convenience stores that meet the demand for both tourist and citizens.

The aims of this project are to find out the relationship between population density and the number of convenience stores and also find out which prefectures should be considered to open additional convenience stores that would meet the future demands.

Data

The data used in this report are collected from below sources.

1. The list of Japanese prefectures by area was captured in the website https://en.wikipedia.org/wiki/List_of_Japanese_prefectures_by_area

2. The longitude and latitude of each prefectures was obtained by a geocoding service in Python.

3. The Foursquare API was used to get the venues and venue categories in each prefecture in Japan.

Data Analysis would be performed to find out the relationship between population density and the number of convenience stores and to find out which prefectures should be considered to open additional convenience stores that would meet the future demands of the citizens and tourists.

Methodology

Python with the package Pandas, Numpy, Requests, matplotlib, folium, geopy, sklearn would be the tools used in the project.

Step 1: Data Collection and preparation – List of Japanese prefectures

The list of Japanese prefectures would be extracted from the website https://en.wikipedia.org/wiki/List_of_Japanese_prefectures_by_area by the package pandas and requests.

```
In [1]: import pandas as pd
        import requests
```

```
In [2]: url = "https://en.wikipedia.org/wiki/List_of_Japanese_prefectures_by_area"
        wiki_url = requests.get(url)
        wiki_url

Out[2]: <Response [200]>
```

```
In [3]: data = pd.read_html(wiki_url.text)
        data

Out[3]: [    Rank Prefecture Japanese EstimatedArea(km²)[2] DeterminedArea(km²)[3]  \
             Rank Prefecture Japanese EstimatedArea(km²)[2] DeterminedArea(km²)[3]
         0    1.0   Hokkaido      北海道              83424.31               83424.31
         1    2.0      Iwate      岩手県              15275.01               15275.01
         2    3.0  Fukushima      福島県              13783.74               13783.74
         3    4.0     Nagano      長野県              13561.56               13104.29
         4    5.0    Niigata      新潟県              12584.10               10363.99
         5    6.0      Akita      秋田県              11637.54               11637.54
         6    7.0       Gifu      岐阜県              10621.29                9768.57
         7    8.0     Aomori      青森県               9645.59                9645.59
         8    9.0   Yamagata      山形県               9323.15                6651.83
         9   10.0  Kagoshima     鹿児島県               9186.94                9042.65
         10  11.0  Hiroshima      広島県               8479.45                8479.45
         11  12.0      Hyogo      兵庫県               8400.96                8400.96
         12  13.0   Shizuoka      静岡県               7777.42                7252.60
         13  14.0   Miyazaki      宮崎県               7735.31                6794.24
         14  15.0   Kumamoto      熊本県               7409.35                7272.41
         15  16.0     Miyagi      宮城県               7282.22                6858.62
         16  17.0    Okayama      岡山県               7114.50                7010.92
```

After observing the data, the first row of the data with the first level of index for the column name would be extracted.

```
In [5]: data = data[0]
        data.columns

Out[5]: MultiIndex([(                  'Rank',                   'Rank'),
                    (            'Prefecture',             'Prefecture'),
                    (              'Japanese',               'Japanese'),
                    (  'EstimatedArea(km²)[2]',   'EstimatedArea(km²)[2]'),
                    ( 'DeterminedArea(km²)[3]',  'DeterminedArea(km²)[3]'),
                    ('UndeterminedArea(km²)[4]', 'UndeterminedArea(km²)[4]'),
                    (            'Population',             'Population'),
                    (          'Density(/km²)',           'Density(/km²)'),
                    (       'Inhabitable Area',                  '(km²)'),
                    (       'Inhabitable Area',           'Density(/km²)'),
                    (          'ForestArea(%)',           'ForestArea(%)'),
                    (               'Capital',                'Capital')],
                   )
```

```
In [6]: data.columns.get_level_values(0)

Out[6]: Index(['Rank', 'Prefecture', 'Japanese', 'EstimatedArea(km²)[2]',
               'DeterminedArea(km²)[3]', 'UndeterminedArea(km²)[4]', 'Population',
               'Density(/km²)', 'Inhabitable Area', 'Inhabitable Area',
               'ForestArea(%)', 'Capital'],
              dtype='object')
```

```
In [7]: data.columns = data.columns.get_level_values(0)
```

The unused column would be dropped and renamed as below so that only the useful column, such as Prefecture, Estimated Area, Population and Density, would be kept in the dataframe.

```
In [8]: df = data
        df.drop(columns=["Rank", "Japanese", "DeterminedArea(km²)[3]", "UndeterminedArea(km²)[4]",
                       "Inhabitable Area","ForestArea(%)","Capital"], inplace = True)
        df.columns = ["Prefecture", "EstimatedArea", "Population", "Density"]
        df
```

Out[8]:

	Prefecture	EstimatedArea	Population	Density
0	Hokkaido	83424.31	5381733	64.51
1	Iwate	15275.01	1279594	83.77
2	Fukushima	13783.74	1914039	138.86
3	Nagano	13561.56	2098804	154.76
4	Niigata	12584.10	2304264	183.11
5	Akita	11637.54	1023119	87.92

The last row for "Japan" would be dropped as it is the grand total in the table, which would be unused in the project.

```
In [9]: df = df[0:46]
        df
```

Out[9]:

	Prefecture	EstimatedArea	Population	Density
0	Hokkaido	83424.31	5381733	64.51
1	Iwate	15275.01	1279594	83.77
2	Fukushima	13783.74	1914039	138.86
3	Nagano	13561.56	2098804	154.76
4	Niigata	12584.10	2304264	183.11
5	Akita	11637.54	1023119	87.92

The data types would be checked to ensure the column Estimated Area, Population and Density are in the number format.

```
In [10]: df.dtypes

Out[10]: Prefecture        object
         EstimatedArea    float64
         Population         int64
         Density          float64
         dtype: object
```

Step 2: Data Collection and preparation – The longitude and latitude of each prefectures

The longitude and latitude of each prefectures would be captured by python. For example, the coordinates of Japan are 36.5748441, 139.2394179.

```
In [11]: import geocoder
         from geopy.geocoders import Nominatim
         from tqdm import tqdm
         from geopy.extra.rate_limiter import RateLimiter
         import numpy as np

In [12]: address = 'Japan'
         geolocator = Nominatim(user_agent="Mozilla/76.0")
         location = geolocator.geocode(address)
         latitude = location.latitude
         longitude = location.longitude
         print('The coordinates of Japan are {}, {}.'.format(latitude, longitude))

         The coordinates of Japan are 36.5748441, 139.2394179.
```

A for-loop would be used to get all the longitude and latitude of each prefectures and therefore, the coordinates would be added into the dataframe. Part of the result would be shown as below:

```
In [15]: for index in df.index:
             if(coords[index]!=None):
                 df.at[index, 'Latitude'] = coords[index].latitude
                 df.at[index, 'Longitude'] = coords[index].longitude

         df
```

Out[15]:

	Prefecture	EstimatedArea	Population	Density	Latitude	Longitude
0	Hokkaido	83424.31	5381733	64.51	43.451983	142.819783
1	Iwate	15275.01	1279594	83.77	39.972417	141.212442
2	Fukushima	13783.74	1914039	138.86	37.754540	140.459214
3	Nagano	13561.56	2098804	154.76	36.114395	138.031902
4	Niigata	12584.10	2304264	183.11	37.645228	138.766912
5	Akita	11637.54	1023119	87.92	39.689880	140.342608
6	Gifu	10621.29	2031903	191.30	35.786745	137.046078
7	Aomori	9645.59	1308265	135.63	40.886943	140.590121
8	Yamagata	9323.15	1123891	120.55	38.255339	140.340021
9	Kagoshima	9186.94	1648177	179.40	31.572494	130.527873
10	Hiroshima	8479.45	2843990	335.40	34.391606	132.451816

Step 3: Visualization the neighborhoods of Japan

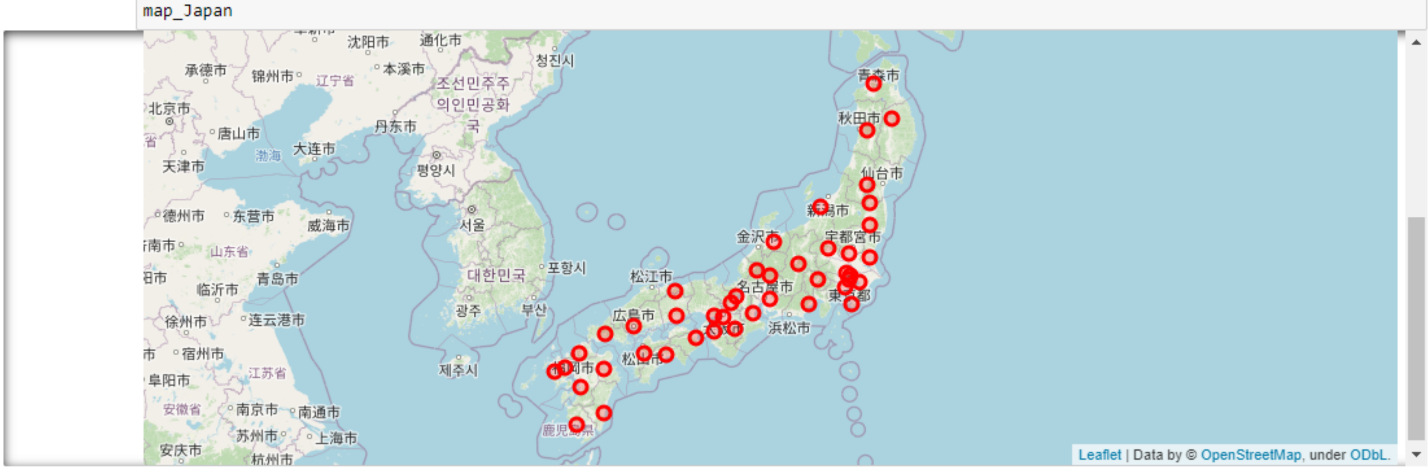
A map is plotted based on the coordinates of each Prefecture by using the packages folium.

```
In [16]: import folium
```

```
In [17]: # Creating the map of Toronto
         map_Japan = folium.Map(location=[latitude, longitude], zoom_start=11)

         # adding markers to map
         for latitude, longitude, Prefecture in zip(df['Latitude'], df['Longitude'], df['Prefecture']):
             label = '{}'.format(Prefecture)
             label = folium.Popup(label, parse_html=True)
             folium.CircleMarker(
                 [latitude, longitude],
                 radius=5,
                 popup=label,
                 color='red',
                 fill=True
                 ).add_to(map_Japan)

         map_Japan
```



Step 4: Data Collection and Preparation – Venue and Venue Categories in Japan

After visualizing the coordinates of prefectures in Japan, it is important to find out the characteristics of the neighborhoods and the common venue and venue categories. A function using Foursquare would be created to collect the information of the venue and venue categories of the neighborhoods within 500m radius as below:

```
In [81]: 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={}'.format(
                     CLIENT_ID,
                     CLIENT_SECRET,
                     VERSION,
                     lat,
                     lng,
                     radius
                     )

                 # 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']['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 = ['Prefecture',
                           'Prefecture Latitude',
                           'Prefecture Longitude',
                           'Venue',
                           'Venue Category']

             return(nearby_venues)
```

Part of the result would be shown as below:

```
In [22]: venues_in_japan.head()
```

Out[22]:

	Prefecture	Prefecture Latitude	Prefecture Longitude	Venue	Venue Category
0	Iwate	39.972417	141.212442	道の駅 石神の丘	Rest Area
1	Iwate	39.972417	141.212442	レストラン石神の丘	Restaurant
2	Iwate	39.972417	141.212442	沼宮内交差点	Intersection
3	Iwate	39.972417	141.212442	石神の丘交差点	Intersection
4	Iwate	39.972417	141.212442	石神の丘美術館	Art Museum

Step 5: One Hot Encoding the venue categories

To find out the difference between the venue categories and the top 10 common venue as the similarity used in the project, the method One Hot Encoding would be used to turn the categorical variables into the numeric variables.

In [25]:
```python
japan_venue_cat = pd.get_dummies(venues_in_japan[['Venue Category']], prefix="", prefix_sep="")
japan_venue_cat
```

Out[25]:

	American Restaurant	Arcade	Art Museum	Arts & Crafts Store	Asian Restaurant	BBQ Joint	Bakery	Bar	Bath House	Beach	...	Thrift / Vintage Store	Tourist Information Center	Toy / Game Store	Train Station	Turkish Restaurant	Udon Restaurant	V / R
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
4	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
623	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	
624	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
625	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
626	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
627	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	

628 rows × 138 columns

The neighborhood would then be added into the dataframe.

In [91]:
```python
japan_venue_cat['Prefecture'] = venues_in_japan['Prefecture']

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

japan_venue_cat.head()
```

Out[91]:

	Prefecture	American Restaurant	Arcade	Art Museum	Arts & Crafts Store	Asian Restaurant	BBQ Joint	Bakery	Bar	Bath House	...	Thrift / Vintage Store	Tourist Information Center	Toy / Game Store	Train Station	Turkish Restaurant	Udon Restaurant
0	Iwate	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
1	Iwate	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
2	Iwate	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
3	Iwate	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
4	Iwate	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0

5 rows × 138 columns

The mean of the categories in each neighborhood would be calculated by grouping the Prefecture

In [92]:
```python
japan_grouped = japan_venue_cat.groupby('Prefecture').mean().reset_index()
japan_grouped.head()
```

Out[92]:

	Prefecture	American Restaurant	Arcade	Art Museum	Arts & Crafts Store	Asian Restaurant	BBQ Joint	Bakery	Bar	Bath House	...	Thrift / Vintage Store	Tourist Information Center	Toy / Game Store	Train Station	Turkish Restaurai
0	Aichi	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.000000	...	0.0	0.000000	0.0	0.000000	0.00000
1	Chiba	0.0	0.066667	0.0	0.0	0.0	0.000000	0.000000	0.033333	0.000000	...	0.0	0.000000	0.0	0.000000	0.00000
2	Fukushima	0.0	0.033333	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.033333	...	0.0	0.000000	0.0	0.066667	0.00000
3	Hiroshima	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.000000	...	0.0	0.033333	0.0	0.000000	0.0333
4	Hyogo	0.0	0.000000	0.0	0.0	0.0	0.033333	0.033333	0.000000	0.000000	...	0.0	0.000000	0.0	0.033333	0.00000

5 rows × 138 columns

Step 6: Find out top 10 Venues in the neighborhoods

A function to get the top 10 venue categories in the neighborhoods would be created as below:

In [28]:
```
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]
```

In [29]:
```
import numpy as np
```

In [30]:
```
num_top_venues = 10

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

columns = ['Prefecture']
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['Prefecture'] = japan_grouped['Prefecture']

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

neighborhoods_venues_sorted.head()
```

Out[30]:

	Prefecture	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	Aichi	Park	Yoshoku Restaurant	Flower Shop	Fishing Store	Fish Market	Fast Food Restaurant	Electronics Store	Dumpling Restaurant	Drugstore	Cupcake Shop
1	Chiba	Ramen Restaurant	Coffee Shop	Café	Arcade	Electronics Store	Sushi Restaurant	Fast Food Restaurant	Burger Joint	Chinese Restaurant	Rock Club
2	Fukushima	Ramen Restaurant	Hotel	Convenience Store	Train Station	Chinese Restaurant	Bowling Alley	Donut Shop	Plaza	Hobby Shop	Dumpling Restaurant
3	Hiroshima	Café	Okonomiyaki Restaurant	Seafood Restaurant	Hotel	Ramen Restaurant	Park	Memorial Site	Sculpture Garden	Electronics Store	Soba Restaurant
4	Hyogo	Convenience Store	Intersection	Ramen Restaurant	Chinese Restaurant	Restaurant	Coffee Shop	Donburi Restaurant	Japanese Restaurant	Discount Store	Grocery Store

Step 7: Using K-Means model for clustering

K-Means model for clustering would be used to cluster the similar neighborhoods based on the similarity. In this project, number of clusters = 5 would be used for data analysis.

In [31]:
```python
from sklearn.cluster import KMeans
```

In [32]:
```python
k_num_clusters = 5

japan_grouped_clustering = japan_grouped.drop('Prefecture', 1)

kmeans = KMeans(n_clusters=k_num_clusters, random_state=0).fit(japan_grouped_clustering)
kmeans
```

Out[32]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=0, tol=0.0001, verbose=0)
```

In [33]:
```python
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
```

In [34]:
```python
japan_merged = df
japan_merged = japan_merged.join(neighborhoods_venues_sorted.set_index('Prefecture'), on='Prefecture')
japan_merged.head()
```

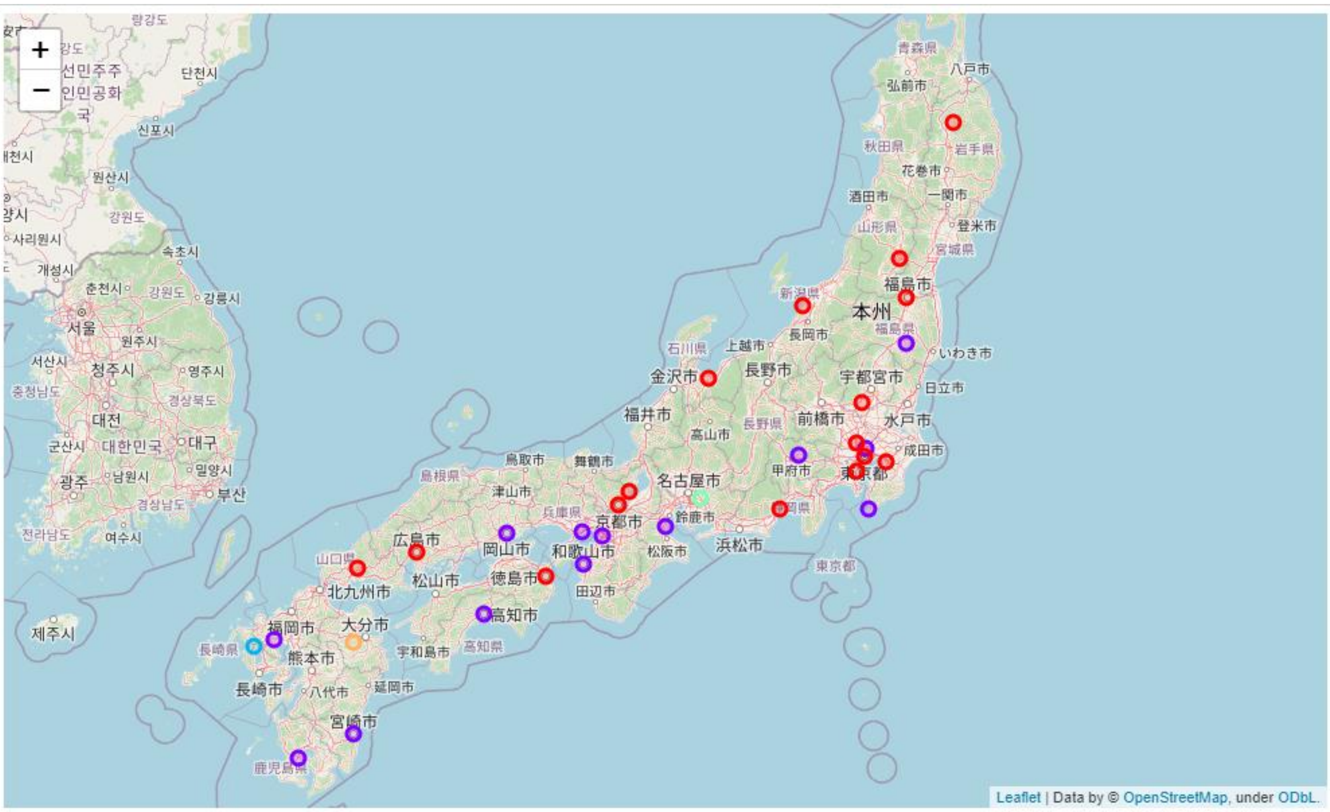
Out[34]:

	Prefecture	EstimatedArea	Population	Density	Latitude	Longitude	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 M Comr Venu
0	Hokkaido	83424.31	5381733	64.51	43.451983	142.819783	NaN	NaN	NaN	NaN	NaN	NaN	NaN	I
1	Iwate	15275.01	1279594	83.77	39.972417	141.212442	0.0	Intersection	Art Museum	Rest Area	Restaurant	Yoshoku Restaurant	Drugstore	Fisl S
2	Fukushima	13783.74	1914039	138.86	37.754540	140.459214	0.0	Ramen Restaurant	Hotel	Convenience Store	Train Station	Chinese Restaurant	Bowling Alley	Di S
3	Nagano	13561.56	2098804	154.76	36.114395	138.031902	NaN	NaN	NaN	NaN	NaN	NaN	NaN	I
4	Niigata	12584.10	2304264	183.11	37.645228	138.766912	0.0	Fish Market	Japanese Restaurant	Park	Seafood Restaurant	Noodle House	Marine Terminal	Fisl S

Result

The result cluster would be visualized in the below map:

Out[37]:



Cluster 1

```
In [55]: c[japan_merged_nonan['Cluster Labels'] == 0, japan_merged_nonan.columns[[0] + [3] + list(range(6, japan_merged_nonan.shape[1]))]]
```

Out[55]:

Prefecture	Density	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
Iwate	83.77	0.0	Intersection	Art Museum	Rest Area	Restaurant	Yoshoku Restaurant	Drugstore	Fishing Store	Fish Market	Fast Food Restaurant	Electronics Store
Fukushima	138.86	0.0	Ramen Restaurant	Hotel	Convenience Store	Train Station	Chinese Restaurant	Bowling Alley	Donut Shop	Plaza	Hobby Shop	Dumpling Restaurant
Niigata	183.11	0.0	Fish Market	Japanese Restaurant	Park	Seafood Restaurant	Noodle House	Marine Terminal	Fishing Store	Ramen Restaurant	Diner	Convenience Store
Yamagata	120.55	0.0	Convenience Store	Ramen Restaurant	Sake Bar	Coffee Shop	Dessert Shop	Soba Restaurant	Bookstore	Intersection	Fast Food Restaurant	Miscellaneous Shop
Hiroshima	335.40	0.0	Café	Okonomiyaki Restaurant	Seafood Restaurant	Hotel	Ramen Restaurant	Park	Memorial Site	Sculpture Garden	Electronics Store	Soba Restaurant
Shizuoka	475.78	0.0	Convenience Store	Historic Site	Concert Hall	Ramen Restaurant	Bookstore	Park	Donut Shop	Coffee Shop	Outdoor Sculpture	Noodle House
Tochigi	308.09	0.0	Museum	Ramen Restaurant	Boat or Ferry	Convenience Store	Department Store	Coffee Shop	Restaurant	River	Café	Fast Food Restaurant
Yamaguchi	229.82	0.0	Café	Bakery	Restaurant	Japanese Restaurant	Park	Diner	Italian Restaurant	Mountain	Arcade	Clothing Store
Chiba	1206.49	0.0	Ramen Restaurant	Coffee Shop	Café	Arcade	Electronics Store	Sushi Restaurant	Fast Food Restaurant	Burger Joint	Chinese Restaurant	Rock Club
Kyoto	565.97	0.0	Japanese Restaurant	Hotel	Convenience Store	Japanese Curry Restaurant	Chinese Restaurant	Concert Hall	Soba Restaurant	Historic Site	Gourmet Shop	French Restaurant
Toyama	251.04	0.0	Japanese Restaurant	Hotel	Park	Sake Bar	Convenience Store	Discount Store	Scenic Lookout	Chinese Restaurant	Seafood Restaurant	Kushikatsu Restaurant
Tokushima	182.25	0.0	Convenience Store	Park	Bed & Breakfast	Ramen Restaurant	Chinese Restaurant	Restaurant	Historic Site	Cupcake Shop	Okonomiyaki Restaurant	Burger Joint
Shiga	351.70	0.0	Platform	Bakery	Yakitori Restaurant	Bus Station	Train Station	Park	Food & Drink Shop	Beach	Japanese Restaurant	Yoshoku Restaurant
Saitama	1913.38	0.0	Convenience Store	Noodle House	Park	Coffee Shop	Pizza Place	Food & Drink Shop	Japanese Family Restaurant	Soba Restaurant	Udon Restaurant	Donburi Restaurant
Kanagawa	3777.67	0.0	Ramen Restaurant	Convenience Store	Park	Japanese Restaurant	Garden Center	Chinese Restaurant	Rental Car Location	Donburi Restaurant	Sake Bar	Smoke Shop
Tokyo	6168.74	0.0	Historic Site	Park	Hotel Bar	Paper / Office Supplies Store	Mediterranean Restaurant	Lounge	Japanese Restaurant	Italian Restaurant	Hotel	Garden

Cluster 2

In [54]:
```
japan_merged_nonan.loc[japan_merged_nonan['Cluster Labels'] == 1, japan_merged_nonan.columns[[0] + [3] + list(range(6, japan_merg
```

Out[54]:

	Prefecture	Density	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
9	Kagoshima	179.40	1.0	Convenience Store	Mobile Phone Shop	Chinese Restaurant	Pharmacy	Candy Store	Dumpling Restaurant	Fishing Store	Fish Market	Fast Food Restaurant	Electronics Store
11	Hyogo	658.83	1.0	Convenience Store	Intersection	Ramen Restaurant	Chinese Restaurant	Restaurant	Coffee Shop	Donburi Restaurant	Japanese Restaurant	Discount Store	Grocery Store
13	Miyazaki	142.73	1.0	Convenience Store	Hotel	Ramen Restaurant	Café	Bed & Breakfast	Parking	Bus Stop	Japanese Restaurant	Udon Restaurant	Donburi Restaurant
15	Miyagi	320.49	1.0	Convenience Store	Historic Site	Café	Gourmet Shop	Yoshoku Restaurant	Dumpling Restaurant	Fishing Store	Fish Market	Fast Food Restaurant	Electronics Store
16	Okayama	270.09	1.0	Convenience Store	Ramen Restaurant	Noodle House	Diner	Coffee Shop	BBQ Joint	Intersection	Park	Soba Restaurant	Yakitori Restaurant
17	Kochi	102.52	1.0	Convenience Store	Hotel	Noodle House	Japanese Restaurant	Souvenir Shop	Tourist Information Center	Café	Supermarket	Deli / Bodega	Toy / Game Store
18	Shimane	103.51	1.0	Convenience Store	Intersection	Japanese Restaurant	Supermarket	Fishing Store	BBQ Joint	Bus Stop	Japanese Family Restaurant	Drugstore	Sake Bar
24	Mie	314.47	1.0	Convenience Store	Fast Food Restaurant	Ice Cream Shop	Ramen Restaurant	Discount Store	Japanese Restaurant	Japanese Curry Restaurant	Supermarket	Shopping Mall	Sake Bar
29	Wakayama	203.95	1.0	Convenience Store	Ramen Restaurant	Bed & Breakfast	Yoshoku Restaurant	Discount Store	Pachinko Parlor	Drugstore	Nightclub	Shopping Mall	Japanese Restaurant
31	Yamanashi	186.98	1.0	Convenience Store	Sake Bar	Japanese Restaurant	Chinese Restaurant	Supermarket	Udon Restaurant	Ramen Restaurant	Deli / Bodega	Donut Shop	Fast Food Restaurant
34	Ishikawa	275.68	1.0	Convenience Store	Ramen Restaurant	Drugstore	Furniture / Home Store	Supermarket	Clothing Store	Fish Market	Fast Food Restaurant	Electronics Store	Dumpling Restaurant
41	Saga	341.23	1.0	Convenience Store	Japanese Family Restaurant	Supermarket	Takoyaki Place	Ramen Restaurant	Harbor / Marina	Gym / Fitness Center	Cupcake Shop	Deli / Bodega	Department Store
45	Osaka	4639.80	1.0	Convenience Store	Grocery Store	Intersection	Japanese Curry Restaurant	Fruit & Vegetable Store	Fast Food Restaurant	Chinese Restaurant	Donut Shop	Bike Shop	Japanese Restaurant

Cluster 3

In [49]:
```
japan_merged_nonan.loc[japan_merged_nonan['Cluster Labels'] == 2, japan_merged_nonan.columns[[0] + [3] + list(range(6, japan_merg
```

Out[49]:

	Prefecture	Density	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
36	Nagasaki	333.29	2.0	Convenience Store	Yoshoku Restaurant	Drugstore	Flower Shop	Fishing Store	Fish Market	Fast Food Restaurant	Electronics Store	Dumpling Restaurant	Donut Shop
43	Okinawa	628.45	2.0	Convenience Store	Grocery Store	Yoshoku Restaurant	Flower Shop	Fishing Store	Fish Market	Fast Food Restaurant	Electronics Store	Dumpling Restaurant	Drugstore

Cluster 4

In [47]:
```
:[japan_merged_nonan['Cluster Labels'] == 3, japan_merged_nonan.columns[[0] + [3] + list(range(6, japan_merged_nonan.shape[1]))]]
```

Out[47]:

	Prefecture	Density	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
26	Aichi	1446.72	3.0	Park	Yoshoku Restaurant	Flower Shop	Fishing Store	Fish Market	Fast Food Restaurant	Electronics Store	Dumpling Restaurant	Drugstore	Cupcake Shop

Cluster 5

```
In [48]: japan_merged_nonan.loc[japan_merged_nonan['Cluster Labels'] == 4, japan_merged_nonan.columns[[0] + [3] + list(range(6, japan_merg
```

Out[48]:

	Prefecture	Density	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
21	Oita	183.94	4.0	Flower Shop	Japanese Restaurant	Yoshoku Restaurant	Drugstore	Fishing Store	Fish Market	Fast Food Restaurant	Electronics Store	Dumpling Restaurant	Donut Shop

Discussion

Cluster 1

It is a cluster with a lot of restaurant and hotel. However, it seems the convenience stores is not very common in this cluster. As convenience store may have the similar characteristics as restaurant. Opening some convenience stores maybe beneficial to the citizens or tourists as they can buy the day-use items, food or drinks in the convenience stores. Therefore, a lot of people may go into the convenience stores to buy food and drinks.

Cluster 2

It is a cluster with sufficient convenience store, in which there are famous prefecture in Japan, such as Osaka and Kiyazaki. It is undoubted that this are a lot of citizens and tourists in the cluster. Therefore, there are already enough convenience stores in this cluster. It is not recommended to open some new convenience stores in this cluster as there maybe a lot of competitors in this clusters.

Cluster 3

As similar as cluster 2, it is a cluster with plenty of convenience store. With the same reason as cluster 2, it is also not suggested to open some new convenience stores in this cluster.

Cluster 4

It is a cluster with a lot of park and restaurant. It seems it maybe beneficial to open a convenience store in this cluster as the citizens may go to buy drinks or food after playing in the park.

Cluster 5

It is a cluster with flower shop as the top common venue. It seems that it may not be beneficial to open a convenience store as the citizen may not need to buy drinks or food after shopping in a flower shop.

To combine all the result, it is recommended to open some new convenience stores in cluster 1 rather than cluster 4. It is because there are a lot of different restaurant and hotel in cluster 1. If opening some new convenience stores in cluster 1, it can attract both tourist and citizens. However, in cluster 4, it has only a lot of park and restaurants. If opening some new convenience stores in cluster 4, it can only attract some of the citizens.

If investigating the cluster 1 deeply, the top prefecture for opening a new convenience stores would be Fukushima. As the population density would be low and it has a lot of restaurant, hotel and train station, which can attract both tourist and citizens to go into the convenience stores to buy day-used item and food or drinks.

Conclusion

The aims of this project were to find out the relationship between population density and the number of convenience stores and also find out which prefectures should be considered to open additional convenience stores that would meet the future demands. In this project, the data for the list of japan prefecture and their coordinates were captured from the internet and then a K-Means clustering method was used to find out 5 cluster among the list of Japan prefectures. It is recommended to open a new convenience stores in Cluster 1, especially in the Fukushima as the stores can serve both tourist and citizens.

References

1. Japan Contribution of travel and tourism to GDP (% of GDP), 1995-2019. (n.d.). Retrieved November 11, 2020, from https://knoema.com/atlas/Japan/topics/Tourism/Travel-and-Tourism-Total-Contribution-to-GDP/Contribution-of-travel-and-tourism-to-GDP-percent-of-GDP

2. Tourism in Japan. (2020, September 04). Retrieved November 11, 2020, from https://en.wikipedia.org/wiki/Tourism_in_Japan