How similar are Indonesian Embassies based on their location

boy.setiawan
April 4th, 2021

1. Introduction

1.1. Problem Description

An Embassy is a representative of a country in other countries, their existence helps to indicate a relation among countries and serve as a way to communicate or strengthen the ties. Their location follows a strict and complex requirements both from the country it comes from and the country it resides, usually in a special diplomatic compound or district. Despite all of the careful planning and requirements, the decision to establish an embassy could have come from other necessities such as certain neighborhood/district/area, near to and close from certain amneties, places that could support the embassy mission etc.

The knowledge of how a certain embassies are similar or different could **help give a bigger view to Indonesian Foreign Affairs Officials** to understand the general environment their embassies are located. Should a certain embassies need to be treated differently, do embassies with certain criteria experience the same or different stress level of working for their staffs, do certain embassies experience certain disturbance etc.

The task to find and group embassies into their own similar group could be a daunting task. From forming the basic data, to gathering and analyzing the data must be done without bias and subjectivity. The end result must be able to give an insight of what the embassies really are so they can be interpreted and understand as a whole new information or to strengthen the already known facts.

1.2. Data Description

In order to accomplish the analysis, we will need data on Indonesian Embassies abroad and their latitude and longitude so we can combine it with data about venues surrounding the embassies which we intend to explore from FourSquare API to give a better understanding about the embassy's neighborhood, venues and places.

1.3. Solution

Our solution is to cluster the embassies and see the differences and similarities between them based on the data gathered about venues surrounding the embassies.

1.4. Methodology

A. Data Gathering

1. List of Indonesian Embassies Abroad

Apparently to get data about the list of Indonesian Embassies abroad is going to be a bit difficult, this is because the Ministry of Foreign Affairs website doesn't display the data to be easily analyzable. But with the help of google and wikipedia we find a page that we could extract the data that we need as a starting point.

https://id.wikipedia.org/wiki/Kedutaan_besar_Republik_Indonesia

	Country	Capital
0	Afganistan\n	Kabul [1]\n
1	Afrika Selatan\n	Pretoria [2]\n
2	Aljazair\n	Algiers\n
3	Amerika Serikat\n	Washington, D.C. [3]\n
4	Arab Saudi\n	Riyadh\n
5	Argentina\n	Buenos Aires [4]\n
6	Australia\n	Canberra [5]\n
7	Austria\n	Wina\n
8	Azerbaijan\n	Baku\n
9	Bahrain\n	Manama [6]\n
10	Bangladesh\n	Dhaka [7]\n

After scraping the page with BeautifulSoup we manage to import the data in a Data Frame as so. There are 98 Indonesian Embassies according to our data. As we can see the data needs to be cleaned since the scrapping process leave us with some characters we need to eliminate.

Clean the data to avoid problems later

1. Get rid of the '\n'
2. Get rid of the []

```
df_ID_embassies['Country'] = df_ID_embassies['Country'].replace(f'(\n)', '',regex=True)
df_ID_embassies['Capital'] = df_ID_embassies['Capital'].replace(f'(\n)', '',regex=True)
df_ID_embassies['Capital'] = df_ID_embassies['Capital'].replace(f'(\[\d+\])', '',regex=True) #inside [] with one more digits
```

After we clean the Data Frame we need to change some values and drop some rows from the Data Frame. This is because some of the data used pronunciation of capitals in Indonesian language, others not so common name and data that does not interest us such as Indonesian representatives for the United Nations.

The final tally is 93 Indonesian Embassies abroad.

2. The Embassies Address

It would be nice if everything was served on a plate but in this task, we need to have some wits up our sleeves. What we need are longitudes and latitudes of the embassies, as with the problem above it is not readily available to us. But there is hope since the address is available for us to extract from the Ministry of Foreign Affairs website for they have a common URL that we can explore.

https://kemlu.go.id/CAPITAL

with this information we need to iterate this URL with our Data Frame and do another scrapping to get the all the addresses.

apparently the address from scrapping each embassy website for the address needs a make over.

```
embassies_address['Address'] = embassies_address['Address'].str.strip()
embassies_address['Address'] = embassies_address['Address'].str.replace('^ +', '_',regex=True)
embassies_address['Address'] = embassies_address['Address'].str.replace(' +$', '_',regex=True)
embassies_address['Address'] = embassies_address['Address'].replace(r'\\n',' ', regex=True)
embassies_address['Address'] = embassies_address['Address'].replace(to_replace=[r"\\t|\\n|\\r", "\t|\n|\r"], value=["",""], 
embassies_address['Address'] = embassies_address['Address'].replace(r"(?i)[^0-9a-z!?.;,@' -]",'',regex=True)
embassies_address
```

	Address
0	Malalai Watt, Shah-re-Naw, Ministry of Interio...
1	Embassy of the Republic of Indonesia949 Franci...
2	Embassy of the Republic of Indonesia-61, Avenu...
3	
4	Diplomatic Quarter, P.O. Box 94343 - Riyadh 11693
5	Mariscal Ramon Castilla 2901, 1425 Capital Fed...
6	Embassy of the Republic of IndonesiaAddress 8 ...
7	Embassy of the Republic of Indonesia in Vienna...

```
df_ID_embassies
```

	Country	Capital	Address
0	Afganistan	Kabul	Shah-re-Naw Ministry of Interior Street Kabul
1	Afrika Selatan	Pretoria	949 Francis Baard Street Hatfield. Pretoria
2	Aljazair	Algiers	Avenue Souidani Boudjemaa 61 Algiers
3	Amerika Serikat	Washington	2020 Massachusetts Avenue NW. Washington DC
4	Arab Saudi	Riyadh	Diplomatic Quarter. Riyadh
5	Argentina	Buenos Aires	Mariscal Ramon Castilla 2901. Buenos Aires
6	Australia	Canberra	8 Darwin Avenue Yarralumla. Canberra
7	Austria	Wina	Gustav Tschermakgasse 5-7 Vienna
8	Azerbaijan	Baku	Azer Aliyev 3 Nasimi Baku
9	Bahrain	Manama	Villa 2113 Road 2432 Manama
10	Bangladesh	Dhaka	Road No 53 Plot No 14 Gulshan Dhaka
11	Belanda	Den Haag	Tobias Asserlaan 8 Den Haag
12	Belgia	Brussels	Boulevardde la Woluwe 38 Brussels
13	Bosnia dan Herzegovina	Sarajevo	Splitska 9. Sarajevo
14	Brasil	Brasilia	SES Avenida Das Nacoes Quadra 805 Brasilia-DF
15	Britania Raya	London	30 Great Peter Street. London
16	Brunei	Bandar Seri Begawan	Jalan Kebangsaan Kampung Kawasan Diplomatik Mu...
17	Bulgaria	Sofia	Simeonovsko Shosse Sofia

After looking at the result we need to take a decision to save the data frame and edit the address manually since the format is not unison. Not a very satisfying task but something we need to take since creating regex rules does not look like a viable solution for this problem. After that we integrate it with our Data Frame and voila a list of Indonesian Embassies abroad with their address.

3. The Embassies Geo Location

```python
from geopy.geocoders import Nominatim
from geopy.extra.rate_limiter import RateLimiter
```

```python
geocoder = Nominatim(user_agent='embassies')
geocode = RateLimiter(geocoder.geocode, min_delay_seconds=3, return_value_on_exception=None)
```

Get the embassies addresses

```python
address = df_ID_embassies['Address'].values
```

Loop all the addresses

```python
long_and_lat = []
for addr in address:
    print(addr)
    location = geocode(addr)
    long_and_lat.append(location)
```

As our initial data requirement is to have a list of embassies with their geolocation, we need to translate the address into latitude and longitude coordinate. To do that we use a common library in python called geocoder and loop through the address.

And then format the result in a Data Frame and combine with our embassies Data Frame.

	Country	Capital	Address	latitude	longitude
0	Afganistan	Kabul	Shah-re-Naw Ministry of Interior Street Kabul	0.000000	0.000000
1	Afrika Selatan	Pretoria	949 Francis Baard Street Hatfield. Pretoria	-25.745801	28.240627
2	Aljazair	Algiers	Avenue Souidani Boudjemaa 61 Algiers	0.000000	0.000000
3	Amerika Serikat	Washington	2020 Massachusetts Avenue NW. Washington DC	38.910279	-77.046149
4	Arab Saudi	Riyadh	Diplomatic Quarter. Riyadh	24.677103	46.625145
5	Argentina	Buenos Aires	Mariscal Ramon Castilla 2901. Buenos Aires	-34.579190	-58.399681
6	Australia	Canberra	8 Darwin Avenue Yarralumla. Canberra	-35.303568	149.115401
7	Austria	Wina	Gustav Tschermakgasse 5-7 Vienna	0.000000	0.000000
8	Azerbaijan	Baku	Azer Aliyev 3 Nasimi Baku	0.000000	0.000000
9	Bahrain	Manama	Villa 2113 Road 2432 Manama	26.222771	50.588948
10	Bangladesh	Dhaka	Road No 53 Plot No 14 Gulshan Dhaka	0.000000	0.000000

Unfortunately, there are some embassies without geolocation information.

```python
embassies_without_geolocation = df_ID_embassies[df_ID_embassies['latitude'] == 0]
embassies_without_geolocation
```

	Country	Capital	Address	latitude	longitude
0	Afganistan	Kabul	Shah-re-Naw Ministry of Interior Street Kabul	0.0	0.0
2	Aljazair	Algiers	Avenue Souidani Boudjemaa 61 Algiers	0.0	0.0
7	Austria	Wina	Gustav Tschermakgasse 5-7 Vienna	0.0	0.0
8	Azerbaijan	Baku	Azer Aliyev 3 Nasimi Baku	0.0	0.0
10	Bangladesh	Dhaka	Road No 53 Plot No 14 Gulshan Dhaka	0.0	0.0
12	Belgia	Brussels	Boulevardde la Woluwe 38 Brussels	0.0	0.0
14	Brasil	Brasilia	SES Avenida Das Nacoes Quadra 805 Brasilia-DF	0.0	0.0
16	Brunei	Bandar Seri Begawan	Jalan Kebangsaan Kampung Kawasan Diplomatik Mu...	0.0	0.0
17	Bulgaria	Sofia	Simeonovsko Shosse Sofia	0.0	0.0
21	Ekuador	Quito	CALLE QUITEO LIBRE E15 QUITO	0.0	0.0

In this case, 40 embassies are still without geolocation information.

```python
import http.client, urllib.parse
import json

conn = http.client.HTTPConnection('api.positionstack.com')
access_key = ''
geolocation = []

for data in address_list:
    #print(address_list[data])
    for key in address_list[data]:
        #print(address_list[data][key])
        params = urllib.parse.urlencode({
        'access_key': access_key,
        'query': address_list[data][key],
        #'region': 'kabul',
        'limit': 1,
        })

        conn.request('GET', '/v1/forward?{}'.format(params))
        res = conn.getresponse()
        api_data = res.read()
        result = json.loads(api_data.decode('utf-8'))
        try:
            geolocation.append({key:(result['data'][0]['latitude'],result['data'][0]['longitude'])})
        except:
            geolocation.append({key:(0,0)})

geolocation
```

To overcome this obstacle and after googling about free forward geocoding, we stumble open positionstack.com which offer a diverse and rich API in their service. After following the instruction and documentation, we translate the remaining address to get their latitude and longitude.

And do the process all over again and see if there is any embassy without geolocation information, which there are and decide to do it manually since the number is quite small.

```python
embassies_without_geolocation = df_ID_embassies[df_ID_embassies['latitude'] == 0]
embassies_without_geolocation
```

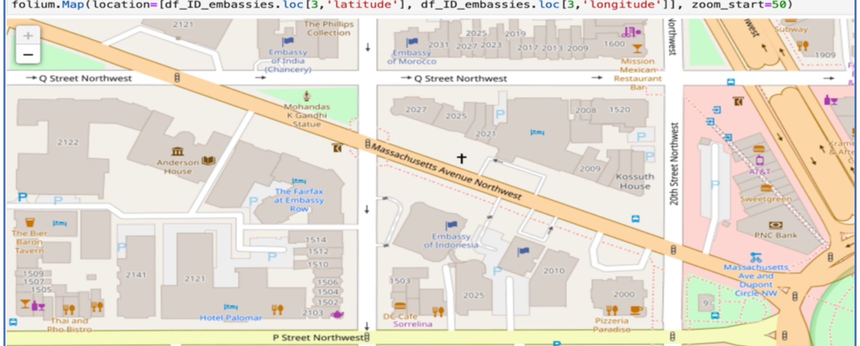
	Country	Capital	Address	latitude	longitude
0	Afganistan	Kabul	Shah-re-Naw Ministry of Interior Street Kabul	0.0	0.0
8	Azerbaijan	Baku	Azer Aliyev 3 Nasimi Baku	0.0	0.0
10	Bangladesh	Dhaka	Road No 53 Plot No 14 Gulshan Dhaka	0.0	0.0
28	Irak	Baghdad	Salhiya Hay Al-I'lam 220 Zukak 5 Baghdad	0.0	0.0
33	Kamboja	Phnom Penh	Street 268 Preah Suramarit Boulevard Phnom Penh	0.0	0.0
35	Kazakhstan	Astana	Saraishyk Street Diplomatic town. Nur-Sultan	0.0	0.0
39	Korea Utara	Pyongyang	Munsudong Taedonggang Distric Pyongyang	0.0	0.0
42	Kuwait	Kuwait City	Daiya Block 1 Rashed Ahmed Al-Roumi Street	0.0	0.0
44	Lebanon	Beirut	Presidential Palace Avenue Rue 68 Sector 3 Beirut	0.0	0.0
45	Libya	Tripoli	Hay Al Karamah Qobri Taariq Al Sari Tripoli	0.0	0.0
55	Oman	Muscat	Al-Shatty Qurum Building Way 3015 Muscat	0.0	0.0
60	Polandia	Warsawa	ulica Estoska 3 Warsawa	0.0	0.0
74	Suriah	Damascus	al-Madina al-Munawara Street Block 270A Buildi...	0.0	0.0

Let's see if we get our geolocation data right with displaying a sample.

```python
import folium #Library for maps
```

let's see if the coordinates is correct for embassy in washington

```python
folium.Map(location=[df_ID_embassies.loc[3,'latitude'], df_ID_embassies.loc[3,'longitude']], zoom_start=50)
```



4. The Embassies Venues

```python
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    index = 1
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(index, name)
        index += 1

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

    return(nearby_venues)
```

Now let's get venues data surrounding the embassies from FourSquare.

```python
embassies_venues.shape
```

(2329, 7)

there are 2329 venues for the embassies

```python
embassies_venues.head()
```

	Embassy	Embassy Latitude	Embassy Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Kabul	34.531210	69.179090	Kabul Star Hotel	34.531202	69.177159	Hotel
1	Kabul	34.531210	69.179090	Green Beans Coffee	34.529924	69.179781	Coffee Shop
2	Kabul	34.531210	69.179090	Bukhara Restaurant	34.529789	69.179680	Afghan Restaurant
3	Kabul	34.531210	69.179090	Spinneys Supermarket	34.533478	69.178270	Supermarket
4	Pretoria	-25.745801	28.240627	Royal Danish Icecream	-25.742076	28.242174	Ice Cream Shop

Save the Data

```python
embassies_venues.to_csv('ID_Embassies_with_venues.csv')
```

Which resulted in an astonishing 2.329 venues which we can work on.

B. Data Exploration

```python
embassies_venues.head(10)
```

	Embassy	Embassy Latitude	Embassy Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Kabul	34.531210	69.179090	Kabul Star Hotel	34.531202	69.177159	Hotel
1	Kabul	34.531210	69.179090	Green Beans Coffee	34.529924	69.179781	Coffee Shop
2	Kabul	34.531210	69.179090	Bukhara Restaurant	34.529789	69.179680	Afghan Restaurant
3	Kabul	34.531210	69.179090	Spinneys Supermarket	34.533478	69.178270	Supermarket
4	Pretoria	-25.745801	28.240627	Royal Danish Icecream	-25.742076	28.242174	Ice Cream Shop
5	Pretoria	-25.745801	28.240627	Namaskar Indian Restaurant	-25.742667	28.242235	Indian Restaurant
6	Pretoria	-25.745801	28.240627	Gautrain Hatfield Station	-25.747655	28.237484	Train Station
7	Pretoria	-25.745801	28.240627	Steers	-25.744297	28.245228	Fast Food Restaurant
8	Pretoria	-25.745801	28.240627	KFC Gordon Road	-25.743100	28.242200	Fried Chicken Joint
9	Pretoria	-25.745801	28.240627	Dros	-25.744867	28.236806	Pub

To get a better understanding of the data, we need to explore our data. Let's see what kind of data we have exactly. We can see that each embassy has venues and their categories. The one we will be using is the value in the Venue Category, because this value will be available across embassies. Let's see how many categories there are in our Data Frame.

```python
embassies_venues.groupby('Embassy').count().sort_values(by='Venue Category',ascending=False).head(100)
```

Embassy	Embassy Latitude	Embassy Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Manila	100	100	100	100	100	100
Mexico City	100	100	100	100	100	100
Washington	95	95	95	95	95	95
Moscow	93	93	93	93	93	93
Seoul	87	87	87	87	87	87
Rome	84	84	84	84	84	84
Berlin	71	71	71	71	71	71
Santiago	62	62	62	62	62	62
London	61	61	61	61	61	61
Paramaribo	2	2	2	2	2	2
Doha	2	2	2	2	2	2
Beirut	2	2	2	2	2	2
Ottawa	1	1	1	1	1	1
Addis Ababa	1	1	1	1	1	1
Abu Dhabi	1	1	1	1	1	1

We can see that that the biggest venues are in Manila, Mexico City and Washington. And the least venues are in Ottawa, Addis Ababa and Abu Dhabi.

Let's see how many categories we have in the Data Frame.

How many categories of venues are there

```python
print('There are {} uniques categories.'.format(len(embassies_venues['Venue Category'].unique())))
```

```
There are 325 uniques categories.
```

let's see what are they

```python
venues = embassies_venues['Venue Category'].unique()
venues
```

```
array(['Hotel', 'Coffee Shop', 'Afghan Restaurant', 'Supermarket',
       'Ice Cream Shop', 'Indian Restaurant', 'Train Station',
       'Fast Food Restaurant', 'Fried Chicken Joint', 'Pub',
       'Gas Station', 'Food Truck', 'Moroccan Restaurant',
       'Metro Station', 'Restaurant', 'Farmers Market',
       'Asian Restaurant', 'Salad Place', 'Art Museum', 'Bookstore',
       'Comic Shop', 'Greek Restaurant', 'Furniture / Home Store', 'Park',
       'Pizza Place', 'Social Club', 'Gym', 'Middle Eastern Restaurant',
       'Massage Studio', 'Veterinarian', 'Café', 'Sandwich Place',
       'Event Space', 'Japanese Restaurant', 'Hotel Bar', 'Fountain',
       'History Museum', 'Italian Restaurant', 'Bakery',
```

There are 325 categories and if we see some are generally the same. A Restaurant might have specialty but basically it is a restaurant, so we have to gather identical categories into a more general category so we can have a better model as a result.

```python
restaurants = [data for data in venues if "Restaurant" in data]

for restaurant in restaurants:
    embassies_venues['Venue Category'] = embassies_venues['Venue Category'].replace(f'(^.*{restaurant}.*$)', "Restaurant",regex=True)
```

We do this repeatedly for the categories we want until we are satisfied that there no more ambiguous data in the Data Frame.

```python
print('There are {} uniques categories.'.format(len(embassies_venues['Venue Category'].unique())))
```

```
There are 143 uniques categories.
```

In this case from 325 we managed to shrink it to 143 categories. For the last exploration let's find out if there is any embassy with no venues at all.

```python
embassies_with_venues = embassies_venues['Embassy'].unique()

embassies_with_no_venues = df_ID_embassies[~df_ID_embassies['Capital'].isin(embassies_with_venues)]
embassies_with_no_venues
```

	Country	Capital	Address	latitude	longitude
17	Bulgaria	Sofia	Simeonovsko Shosse Sofia	42.834585	24.221365
39	Korea Utara	Pyongyang	Munsudong Taedonggang Distric Pyongyang	39.024030	125.786960
45	Libya	Tripoli	Hay Al Karamah Qobri Taariq Al Sari Tripoli	32.839090	13.082160
83	Tunisia	Tunis	Rue 15 du Lac Mlaren Les berges Tunis	33.687264	9.007775

Apparently, there are 4 embassies with no venues data. So, we will disregard these embassies in the final result.

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

# add embassy column back to dataframe
embassy_onehot['Embassy'] = embassies_venues['Embassy']

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

embassy_onehot.head(10)
```

	Embassy	ATM	Accessories Store	Airport	Amphitheater	Antique Shop	Aquarium	Arts & Crafts Store	Arts & Entertainment	Athletics & Sports	...	Theme Park	Tourist Information Center	Toy / Game Store	Track	Trail	Train Station
0	Kabul	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
1	Kabul	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
2	Kabul	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
3	Kabul	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
4	Pretoria	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
5	Pretoria	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
6	Pretoria	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1
7	Pretoria	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
8	Pretoria	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
9	Pretoria	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

10 rows × 144 columns

```python
embassy_onehot.shape
```

```
(2329, 144)
```

The next step is to make the categories into their own column in the Data Frame.

Which resulted in 2.329 rows and 144 columns. In the current state our Data Frame can be used to make a cluster.

If we looked back, our first data consist of 93 embassies and after exploring FourSquare we end up with 4 embassies with no venues data. So, if our calculation is correct, we need to sum up 2.329 rows into 89 rows. We do that by grouping the Data Frame by the embassy column and count the mean of each column in the group.

```python
embassy_grouped = embassy_onehot.groupby('Embassy').mean().reset_index()
embassy_grouped.head()
```

	Embassy	ATM	Accessories Store	Airport	Amphitheater	Antique Shop	Aquarium	Arts & Crafts Store	Arts & Entertainment	Athletics & Sports	...	Theme Park	Tourist Information Center	Toy / Game Store	Track	Trail	Train Station	Tram Station
0	Abu Dhabi	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	Abuja	0.0	0.0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	Addis Ababa	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	Algiers	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	Amman	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows x 144 columns

```python
embassy_grouped.shape
```

```
(89, 144)
```

```python
def return_most_common_venues(embassy, num_top_venues):
    temp = embassy
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp = temp.round({'freq': 2})
    temp = temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues)
    dummy = temp[temp['freq'] == 0]
    temp.loc[dummy.index,'venue']= np.nan
    return temp['venue'].tolist()
```

```python
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Embassy']
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
embassies_venues_sorted = pd.DataFrame(columns=columns)
embassies_venues_sorted['Embassy'] = embassy_grouped['Embassy']

for ind in np.arange(embassy_grouped.shape[0]):
    embassies_venues_sorted.iloc[ind, 1:] = return_most_common_venues(embassy_grouped.iloc[ind, :].T.reset_index(), num_top_venues)

embassies_venues_sorted
```

Now let's find out what are the most common venues in each embassy.

	Embassy	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	Abu Dhabi	Grocery Store	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Abuja	Hotel	Restaurant	Grocery Store	Food Shop	Lounge	Shopping Mall	Food Court	Café	Supermarket	Multiplex
2	Addis Ababa	Food Shop	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Algiers	Restaurant	Metro Station	Food Shop	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Amman	Food Shop	Restaurant	Grocery Store	Café	Intersection	Supermarket	Salon	Spa	Drink Shop	Hotel
5	Ankara	Restaurant	Food Shop	Café	Drink Shop	Bar	Grocery Store	Park	Meyhane	Multiplex	Scenic Lookout
6	Astana	Restaurant	Drink Shop	Bar	Spa	Café	Bookstore	Hotel	Stationery Store	Food Shop	NaN
7	Athena	Food Shop	Restaurant	Café	Hotel	Drink Shop	Supermarket	Grocery Store	Gym	Theater	Club
8	Baghdad	Hostel	Hotel	Bus Station	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	Baku	Restaurant	Hotel	Gym	Club	Lounge	Food Shop	Spa	Food Court	Park	Pet Store
10	Bandar Seri Begawan	Restaurant	Drink Shop	Hotel	Park	Café	Garden	Shopping Mall	Museum	Food Shop	Pedestrian Plaza

```
embassies_venues_sorted['1st Most Common Venue'].value_counts()

Restaurant           50
Food Shop            16
Grocery Store         5
Hotel                 4
Park                  2
Bus Stop              1
Landmark              1
Other Repair Shop     1
Airport               1
Lake                  1
Hobby Shop            1
Hostel                1
Plaza                 1
Gym                   1
Train Station         1
Market                1
Café                  1
Name: 1st Most Common Venue, dtype: int64
```

We can see from the Data Frame that Restaurant comes up a lot on the 1st Most Common Venue. Let's count how many of them.

We can see that 50 embassies have Restaurant as their 1st most common venue and Food Shop in second.

```
embassies_venues_sorted['2nd Most Common Venue'].value_counts()

Restaurant                17
Food Shop                 17
Drink Shop                 9
Café                       8
Hotel                      7
Grocery Store              4
Bar                        3
Lounge                     2
Bus Stop                   2
Museum                     2
Business Service           1
Big Box Store              1
Metro Station              1
Salon                      1
Food Court                 1
Electronics Store          1
Pier                       1
Pharmacy                   1
Arts & Entertainment       1
Clothing Store             1
Gym                        1
Hostel                     1
Plaza                      1
Bridge                     1
Dog Run                    1
Supermarket                1
Name: 2nd Most Common Venue, dtype: int64
```

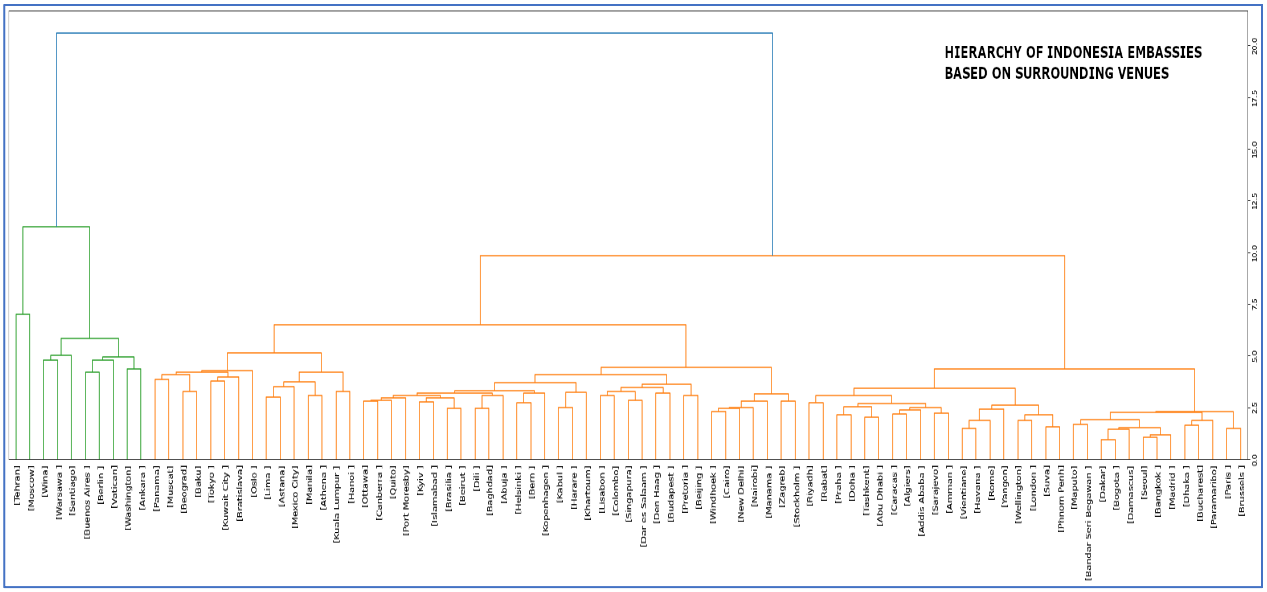
What's the most 2nd Most Common Venue?. Apparently is the same as before with Restaurant and Food Shop as the top two.

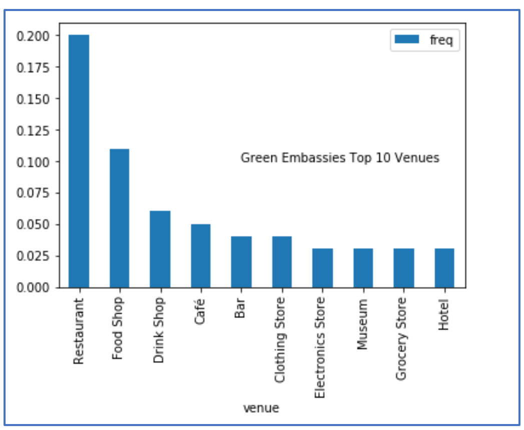
```
embassies_venues_sorted['3rd Most Common Venue'].value_counts()

Hotel                    13
Food Shop                10
Café                      8
Grocery Store             8
Restaurant                7
Bar                       5
Bus Station               4
Park                      4
Drink Shop                3
Shopping Mall             2
Gym                       2
Plaza                     2
Museum                    2
Playground                1
Pedestrian Plaza          1
Brewery                   1
Photography Studio        1
Arts & Entertainment      1
Shopping Plaza            1
Electronics Store         1
Pharmacy                  1
Gas Station               1
Antique Shop              1
Name: 3rd Most Common Venue, dtype: int64
```

What's the most 3rd Most Common Venue?. Looks like Hotel comes up with 13 embassies have it as their 3rd Most Common Venue.

C. Clustering

Before we begin clustering, we must answer the question “do we know how many clusters the data have?”. Based on the problem domain and your knowledge of the data, usually we can answer the question. But if we are clueless about the cluster in the data, there are tools that can help us. First let’s see if our data have any hierarchy in them and built our cluster in a hierarchy.





We can see that the embassy in Wina and Warsawa is very close in venues with the embassy in Santiago. Combine they are very close with the embassies in Buenos Aires, Berlin, Vatican, Washington and Ankara. What’s interesting in the hierarchy, the embassies in the orange hierarchy will eventually resemble the embassies in the green hierarchy. Let’s combine the embassies in the green hierarchy and see what their top venues are to get a better understanding of the hierarchy. We can see that most of them are something that we are expected from the data and nothing is out of the ordinary.

Now let’s cluster our embassies with a question like ‘an embassy is in a group all together if there are at least 7 embassies with the same value, if not then they are an outlier?’. We can do that with the help of DBSCAN which clusters based on the data and the parameters we wanted and also produce a set of outlier data that doesn’t belong to any of the generated clusters. We can see the result below.

	Embassy	Address	latitude	longitude	colors
0	Abu Dhabi	Sultan Bin Zayed Street Str 32 Abu Dhabi	24.365906	54.582223	#8000ff
2	Addis Ababa	Egypt Street Mekanissa Road Woreda 05 Addis A...	9.018947	38.746032	#8000ff
8	Baghdad	Salhiya Hay Al-I'lam 220 Zukak 5 Baghdad	33.319590	44.386390	#8000ff
13	Beirut	Presidential Palace Avenue Rue 68 Sector 3 Beirut	33.845390	35.541880	#8000ff
18	Brasilia	SES Avenida Das Nacoes Quadra 805 Brasilia-DF	-16.793428	-49.295322	#8000ff
25	Canberra	8 Darwin Avenue Yarralumla. Canberra	-35.303568	149.115401	#8000ff
33	Dili	Rua Karketu Mota-Ain No 02 Dili	-8.550615	125.569168	#8000ff
34	Doha	Al Salmiya Street Zone 66 Street 943 Onaiza	25.333074	51.511092	#8000ff
38	Helsinki	Kuusisaarentie 3. Helsinki	60.187281	24.868059	#8000ff
39	Islamabad	Diplomatic Enclave I Street 5 Islamabad	33.721480	73.043290	#8000ff
59	Ottawa	55 Parkdale Avenue. Ottawa	45.410317	-75.734033	#8000ff
64	Port Moresby	Sir John Giuse Drive Lot 12 Section 410 Port M...	-9.434881	147.208705	#8000ff
67	Quito	CALLE QUITEO LIBRE E15 QUITO	-0.229850	-78.524950	#8000ff

The embassies in the list are outliers, they are the embassies whose values are very far apart from the rest of the data and doesn't have at least 7 embassies with similar data.

	Embassy	Address	latitude	longitude	colors
1	Abuja	Katsina Ala Crescent 10 Abuja	9.068530	7.483750	#2c7ef7
9	Baku	Azer Aliyev 3 Nasimi Baku	40.395740	49.821620	#2c7ef7
15	Berlin	Lehrter 16 Berlin	52.524636	13.369861	#2c7ef7
23	Buenos Aires	Mariscal Ramon Castilla 2901. Buenos Aires	-34.579190	-58.399681	#2c7ef7
41	Khartoum	Street 60 Block No 12 Al Riyadh Area Khartoum	15.551770	32.532410	#2c7ef7
42	Kopenhagen	Alle 1 Hellerup Copenhagen	55.722238	12.559591	#2c7ef7
45	Kyiv	17 Universytetska Street. Kyiv	50.419465	30.480809	#2c7ef7
46	Lima	Avenida Las Flores 334-336 San Isidro. Lima	-12.096272	-77.047005	#2c7ef7
50	Manama	Villa 2113 Road 2432 Manama	26.222771	50.588948	#2c7ef7
54	Moscow	Novokuznetskaya Ulitsa No 12 Moscow	55.741469	37.615561	#2c7ef7
61	Paramaribo	Van Brussellaan 3 Paramaribo	5.824594	-55.193191	#2c7ef7
65	Praha	Nad Budankami II 7. Praha	50.071131	14.372931	#2c7ef7
77	Tashkent	YahyoGulomov Street 73 Tashkent	41.264650	69.216270	#2c7ef7
78	Tehran	Ghaemmagham 180 Tehran	35.694390	51.421510	#2c7ef7
79	Tokyo	4 Chome--1 Yotsuya Shinjuku City.Tokyo	35.680587	139.720589	#2c7ef7
80	Vatican	Via Marocco 1000144. Roma	41.820634	12.466099	#2c7ef7
88	Zagreb	Ulica Medveak 56 Zagreb	45.806026	15.976218	#2c7ef7

These are the first cluster, as we can see some of the embassies from the green hierarchy are here like Moscow, Tehran, Berlin and Buenos Aires belong in this cluster.

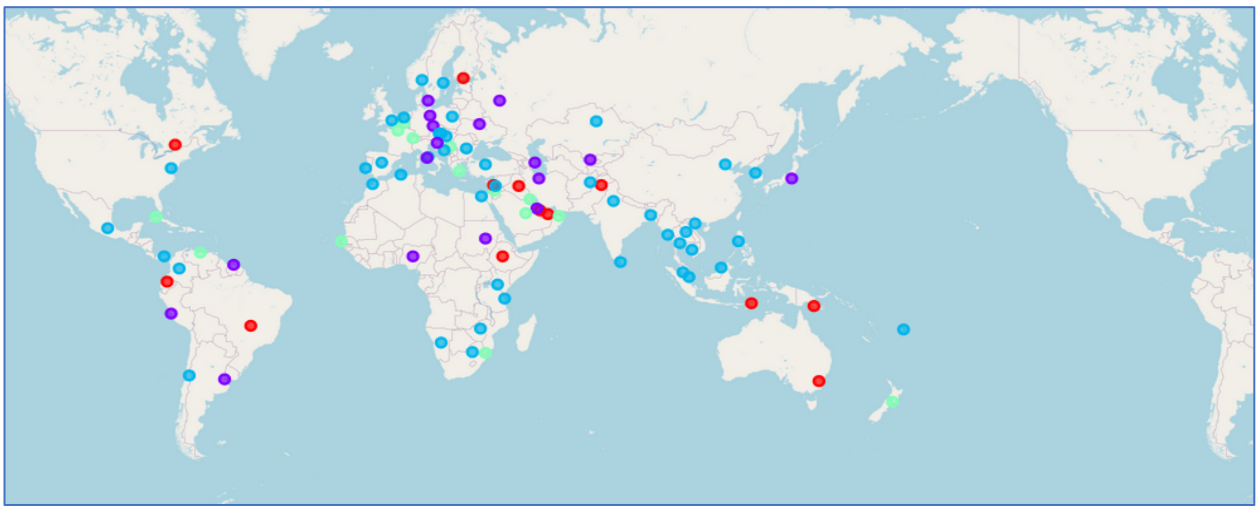
	Embassy	Address	latitude	longitude	colors
3	Algiers	Avenue Souidani Boudjemaa 61 Algiers	36.743950	3.083627	#2adddd
5	Ankara	Prof Dr Aziz Sancar 10 Ankara	39.885977	32.858080	#2adddd
6	Astana	Saraishyk Street Diplomatic town. Nur-Sultan	51.134260	71.425820	#2adddd
10	Bandar Seri Begawan	Jalan Kebangsaan Kampung Kawasan Diplomatik Mu...	4.889737	114.941695	#2adddd
11	Bangkok	Petchburi Road 600 Bangkok	13.755181	100.526715	#2adddd
12	Beijing	Dong Zhi Men Wai Da Jie No 4 Chaoyang District...	39.939696	116.438361	#2adddd
17	Bogota	Calle 70 Bogota	4.704367	-74.123511	#2adddd
19	Bratislava	Brnianska 31. Bratislava	48.164030	17.085551	#2adddd
21	Bucharest	19 Aleea Alexandru Sector 1. Bucharest	44.456520	26.089180	#2adddd
22	Budapest	Varosligeti fasor 26. Budapest	47.509719	19.076236	#2adddd
24	Cairo	Aisha El Taymouria Street 13 Garden City Cairo	30.079694	31.323437	#2adddd
27	Colombo	Sarana Road 400/50 Colombo	6.910436	79.892990	#2adddd
29	Damascus	al-Madina al-Munawara Street Block 270A Buildi...	33.497650	36.251010	#2adddd
30	Dar es Salaam	299 Ali Hassan Mwinyi Road. Dar es Salaam	-6.796460	39.281544	#2adddd
31	Den Haag	Tobias Asserlaan 8 Den Haag	52.086144	4.288699	#2adddd
32	Dhaka	Road No 53 Plot No 14 Gulshan Dhaka	23.796150	90.412780	#2adddd
35	Hanoi	50 Ngo Quyen Street. Hanoi	21.026050	105.855536	#2adddd
36	Harare	Duthie Avenue 3 Harare	-17.796650	31.046799	#2adddd
40	Kabul	Shah-re-Naw Ministry of Interior Street Kabul	34.531210	69.179090	#2adddd
43	Kuala Lumpur	Jalan Tun Razak 233 Kualalumpur	3.146757	101.721745	#2adddd
47	Lisabon	Avenida Dom Vasco da Gama no 40 Lisbon	38.699396	-9.224974	#2adddd
48	London	30 Great Peter Street. London	51.496893	-0.129560	#2adddd
49	Madrid	Calle de Agastia No 65. Madrid	40.444792	-3.650197	#2adddd
51	Manila	Salcedo Street 185 Manila	14.554002	121.015910	#2adddd
53	Mexico City	Julio Verne No 27 Mexico City	19.427980	-99.197181	#2adddd
56	Nairobi	Menengai Rd Upper Hill. Nairobi	-1.300961	36.811411	#2adddd
57	New Delhi	50-A Kautilya Marg Chanakyapuri. New Delhi	28.604079	77.189826	#2adddd
58	Oslo	Fritzners gate 12. Oslo	59.916488	10.704865	#2adddd
60	Panama	Casa no 15 y Ricardo Arango Urbanizacion Obarr...	8.984590	-79.520240	#2adddd
63	Phnom Penh	Street 268 Preah Suramarit Boulevard Phnom Penh	11.557290	104.930180	#2adddd
66	Pretoria	949 Francis Baard Street Hatfield. Pretoria	-25.745801	28.240627	#2adddd
68	Rabat	Rue Beni Boufrah 63 Rabat	34.013250	-6.832550	#2adddd
70	Rome	Via Campania 55. Roma	41.910390	12.493463	#2adddd
71	Santiago	Avenida Las Urbinas 160 Providencia. Santiago	-33.422172	-70.612054	#2adddd
72	Sarajevo	Splitska 9. Sarajevo	43.850707	18.403550	#2adddd
73	Seoul	380 Yeouidaebang-ro Yeongdeungpo-gu. Seoul	37.518528	126.930767	#2adddd
74	Singapura	7 Chatsworth Road. Singapore	1.300208	103.821990	#2adddd
75	Stockholm	Kungsbroplan 1. Stockholm	59.331717	18.049393	#2adddd
76	Suva	Marama Building 91 Gordon Street Fiji	-18.144302	178.426665	#2adddd
81	Vientiane	Kaysone Phomvihane Avenue. Vientiane	17.978149	102.627226	#2adddd
82	Warsawa	ulica Estoska 3 Warsawa	52.236640	21.048470	#2adddd
83	Washington	2020 Massachusetts Avenue NW. Washington DC	38.910279	-77.046149	#2adddd
85	Wina	Gustav Tschermakgasse 5-7 Vienna	48.198674	16.348388	#2adddd
86	Windhoek	103 Nelson Mandela Avenue. Windhoek	-22.571877	17.103135	#2adddd
87	Yangon	Pyiudaungsu Yeiktha Road 100 Yangon	16.805280	96.156110	#2adddd

And this is the second cluster of our embassies where the rest of the green hierarchy belongs such as Ankara, Washington, Wina, Warsawa, Vatican and Santiago.

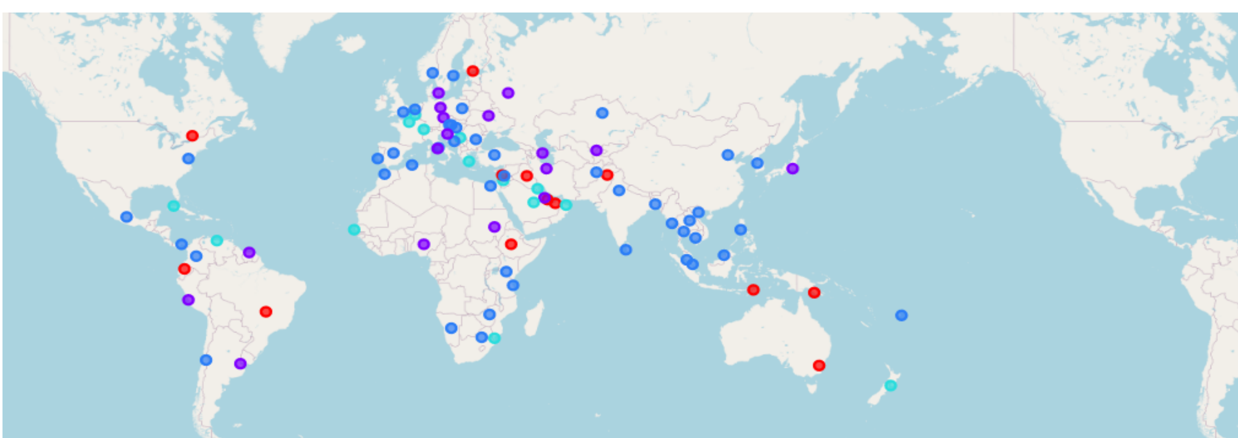
	Embassy	Address	latitude	longitude	colors
4	Amman	Ali Seedo Al-Kurdi Street 13 Amman	31.947359	35.873896	#80ffb4
7	Athena	Marathonodromon Street 15452 Athens	37.959118	23.715552	#80ffb4
14	Beograd	Bulevar Kneza Aleksandra Karadjordjevica No 18...	44.794199	20.448512	#80ffb4
16	Bern	Elfenauweg 51. Bern	46.937197	7.466099	#80ffb4
20	Brussels	Boulevardde la Woluwe 38 Brussels	50.843183	4.371755	#80ffb4
26	Caracas	Avenida El Paseo Caracas	10.452583	-66.879828	#80ffb4
28	Dakar	Avenue Cheikh Anta Diop BP. DAKAR	14.698848	-17.468628	#80ffb4
37	Havana	5ta Avenida 1607 Miramar. La Habana	23.114127	-82.433124	#80ffb4
44	Kuwait City	Daiya Block 1 Rashed Ahmed Al-Roumi Street	29.354400	48.009860	#80ffb4
52	Maputo	Streets No 141 Sommerschield Maputo	-25.965530	32.583220	#80ffb4
55	Muscat	Al-Shatty Qurum Building Way 3015 Muscat	23.605810	58.450130	#80ffb4
62	Paris	47-49 rue Cortambert. Paris	48.861150	2.279244	#80ffb4
69	Riyadh	Diplomatic Quarter. Riyadh	24.677103	46.625145	#80ffb4
84	Wellington	70 Glen Road Kelburn .Wellington	-41.288417	174.763033	#80ffb4

This is a cluster of other embassies that have similarities between them and form their own hierarchy.

Let's visualize our clusters and the outliers together.



What if we have a certain number of clusters in mind, let's say we want to cluster our data into 4 clusters based on a certain knowledge previously known to us. We can use KMeans and see what the results are and visualize our new clusters. As we can see in general the result is quite consistent with the previous cluster.



1.5. Discussion

The result of our cluster shows there are some similarities between the Indonesian Embassies. We based the similarities on venues around the embassies, although we could also based it on any other data according to the problem we are trying to solve. The conclusion of the result of an unlabeled data cluster like we have, usually ends out in the domain knowledge of our intended research in this case the staff on Indonesian Foreign Ministry. But the general audience also can enjoy the result with knowing that their embassies in Australia, Timor Leste and Papua Nugini are almost similar based on the venues around them compare with the embassies in south east asia. Majorities of the embassies in the south and east asia are alike in both clusters.

Another improvements we need to consider is to get get data about public venues such as embassy, government building, public services which usually common around an embassy. Unfortunately we couldn’t get the data without resorting to paid services, which is beyond our reach at the moment.