Coursera Capstone Project

Table of contents

1. Introduction: Business Problem
2. Data
3. Methodology
4. Results
5. Conclusion
6.

1. Introduction : Business Problem

1.1 Description of the problem

The business problem we are currently posing is: This project would specifically help Business people planning to start Restaurants, Hotels, etc. in Jakarta, Indonesia.

The Foursquare API is used to access the venues in the neighborhoods. Since, it returns less venues in the neighborhoods, we would be analysing areas for which countable number of venues are obtained. Then they are clustered based on their venues using Data Science Techniques. Here the k-means clustering algorithm is used to achieve the task. The optimal number of clusters can be obtained using silhouette score. Folium visualization library can be used to visualize the clusters superimposed on the map of Jakarta city. These clusters can be analyzed to help small scale business owners select a suitable location for their need such as Hotels, Shopping Malls, Restaurants, or Coffee shops.

1.2 Discussion of the problem

Jakarta , officially the Special Capital Region of Jakarta, is the capital of Indonesia. It lies on the northwest coast of Java (the world's most populous island). Jakarta is the centre of the economy, culture and politics of Indonesia. It has province level status which had a population of 10,562,088 as of 2020. Although Jakarta extends over only 699.5 square kilometres (270.1 sq mi), and thus has the smallest area of any Indonesian province, its metropolitan area covers 6,392 square kilometres (2,468 sq mi), and is the world's second-most populous urban area, after Tokyo. It has a population of about 35.934 million as of 2020.

Jakarta's business opportunities, and its ability to offer a potentially higher standard of living than is available in other parts of the country, have attracted migrants from across the Indonesian archipelago, making it a melting pot of numerous cultures.

So, how could we leverage Foursquare location data and machine learning to help us make decision and find appropriate neighborhoods? This is the problem I would like to address in this capstone project taking Jakarta as an example. In this project, I am going to use Foursquare location data and clustering methods to group the districts to different group by their Business venues information.

Import the required libraries

```
import numpy as np # library to handle data in a vectorized manner
```

```
import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files
import requests # library to handle requests
import pandas as pd # library for data analsysis
import numpy as np # library to handle data in a vectorized manner
import random # library for random number generation


!pip install geopy
from geopy.geocoders import Nominatim # module to convert an address into latit
ude and longitude values

# libraries for displaying images
from IPython.display import Image
from IPython.core.display import HTML

# tranforming json file into a pandas dataframe library
from pandas.io.json import json_normalize


! pip install folium==0.5.0
import folium # plotting library

!pip install opencage
from opencage.geocoder import OpenCageGeocode

from bs4 import BeautifulSoup

from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score

import matplotlib.cm as cm
import matplotlib.colors as colors
import matplotlib.pyplot as plt
%matplotlib inline

print('Folium installed')
print('Libraries imported.')
```

2. Data

For this project, the data we use are :

- Data Covid-19 cases per distict in Jakarta

Data Source
: https://drive.google.com/file/d/1w5ovPYjXREfd7lz9o3GwAUQuqcDYhCC5/view.

But we will only use **nama_kota (CITY)** and **nama_kelurahan (DISTICT)** columns

- Restaurant in each neighborhood of Jakarta

Data Source : Foursquare API

- Longitude & Latittude of Jakarta City and the Districts

Data Source : OpenCage Geocoder API

```
In [82]: # Read in the data Covid-19 cases per district (28 May,2020)
df = pd.read_excel('https://raw.githubusercontent.com/IbrahimBalweel/capstone-project/main/Standar%20Kelurahan%20Data%20Corona%20(28%20Mei%202020%20Pukul%2009.00).xlsx')
df.head()
```

Out[82]:

	ID_KEL	ID_KEL.1	Nama_provinsi	nama_kota	nama_kecamatan	nama_kelurahan	ODP	Proses Pemantauan	Selesai Pemantauan	PDP	Masih Dirawat	Pul Se
0	NaN	NaN	NaN	NaN	NaN	TOTAL	30704	746	29958	9577	1005	857
1	BELUM DIKETAHUI	BELUM DIKETAHUI	BELUM DIKETAHUI	BELUM DIKETAHUI	BELUM DIKETAHUI	BELUM DIKETAHUI	3723	107	3616	2584	272	231
2	LUAR DKI JAKARTA	LUAR DKI JAKARTA	LUAR DKI JAKARTA	LUAR DKI JAKARTA	LUAR DKI JAKARTA	LUAR DKI JAKARTA	5706	342	5364	1638	228	140
3	3173061005	3173061005	DKI JAKARTA	JAKARTA BARAT	KALI DERES	PEGADUNGAN	129	0	129	61	0	61
4	3174071006	3174071006	DKI JAKARTA	JAKARTA SELATAN	KEBAYORAN BARU	SENAYAN	7	0	7	13	0	13

3. Methodology¶

After little manipulation, the data-frame is obtained as below:

	CITY	DISTRICT
3	JAKARTA BARAT	PEGADUNGAN
4	JAKARTA SELATAN	SENAYAN
5	JAKARTA BARAT	KEBON JERUK
6	JAKARTA UTARA	KELAPA GADING TIMUR
7	JAKARTA BARAT	TOMANG

Get Latitude & Longitude of Jakarta city and the districts

We will use the API service from OpenCage Geocoder to obtain the latitude and

```
# Get latitude and longitude of all districts
key = '3e6c52ff0034daa0a0b1ba2ca760cfa9'
geocoder = OpenCageGeocode(key)

list_lat = []   # create empty lists for latitude
list_long = []  # create empty lists for longitude
for index, row in df.iterrows(): # iterate over rows in dataframe
    District = row['DISTRICT']
    query = str(District)+', Jakarta'
    results = geocoder.geocode(query)
    lat = results[0]['geometry']['lat']
    long = results[0]['geometry']['lng']
    list_lat.append(lat)
    list_long.append(long)
# create new columns from lists
df['Latitude'] = list_lat
df['Longitude'] = list_long
```

Finally, let's make use of Foursquare API and get the top 100 venues that are in Jakarta within a radius of 500 meters.

```
print('There are {} uniques catogories.'.format(len(jkt_venues['Venue Category'].unique())))
```

```
There are 252 uniques catogories.
```

```
jkt_venues.loc[:,'Venue Category'].value_counts()
```

```
Indonesian Restaurant          225
Coffee Shop                    183
Noodle House                   158
Asian Restaurant               157
Fast Food Restaurant           148
Convenience Store              147
Chinese Restaurant             139
Food Truck                     111
Hotel                          101
Café                            87
Pizza Place                     73
Japanese Restaurant             57
Restaurant                      53
Seafood Restaurant              49
Food Court                      46
Bakery                          35
Soup Place                      34
Indonesian Meatball Place       33
Shopping Mall                   32
```

Let's analyze each neighborhood to know about the top 5 venues of each one. Lets create a data-frame with pandas one hot encoding for the venue categories.

```
jkt_onehot = pd.get_dummies(jkt_venues_top['Venue Category'], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
jkt_onehot['Neighborhood'] = jkt_venues_top['Neighborhood']

# move neighborhood column to the first column
fixed_columns = jkt_onehot.columns.tolist()
fixed_columns.insert(0, fixed_columns.pop(fixed_columns.index('Neighborhood')))
jkt_onehot = jkt_onehot.reindex(columns = fixed_columns)

print(jkt_onehot.shape)
jkt_onehot.head()
```

```
(2513, 215)
```

	Neighborhood	Accessories Store	Acehnese Restaurant	Airport	American Restaurant	Arcade	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Australian Restaurant	Auto Dealership	Auto Workshop	Automotive Shop	BBQ Joint	Bakery	Balinese Restaurant	Bar	Basketball Court
7	SENAYAN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	SENAYAN	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	SENAYAN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	SENAYAN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	SENAYAN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Use pandas groupby on neighborhood column and calculate the mean of the

Output each neighborhood along with the top 5 most common venues:

```
#Output each neighborhood along with the top 5 most common venues:
num_top_venues = 5

for hood in jkt_grouped['Neighborhood']:
    print('------'+hood+'------')
    temp = jkt_grouped[jkt_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})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

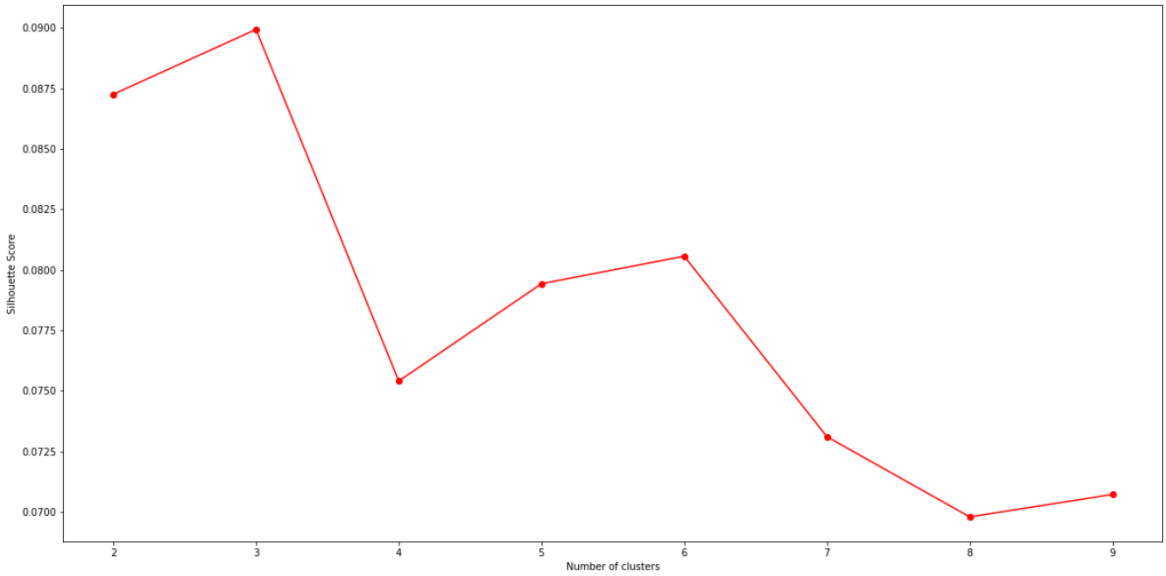
```
------BALI MESTER------
                  venue  freq
0      Convenience Store  0.20
1   Fast Food Restaurant  0.13
2     Chinese Restaurant  0.07
3  Indonesian Restaurant  0.07
4    Japanese Restaurant  0.07


------BANGKA------
          venue  freq
0        Lounge  0.08
1   Coffee Shop  0.08
2  Camera Store  0.08
3          Café  0.08
4        Bakery  0.04


------BENDUNGAN HILIR------
```

Cluster Neighborhoods

Here k-Nearest Neighborhoods clustering technique is used. Lets use the silhouette_score to obtain the best value for the number of clusters.



As seen from the above line plot, the best number of clusters having the highest silhouette score is 4. So, lets consider the number of clusters as 4. Finally, we try to cluster these districts based on the venue categories and use K-Means clustering. So, our expectation would be based on the similarities of venue categories, these districts will be clustered. I have used the code below :

```
# select best number of clusters
kclusters = 4

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

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

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

Add the cluster labels to the neighborhoods_venues_sorted dataframe. And lets create a new datframe jkt_merged which has the neighborhood details, cluster labels and the 10 most common venues in that neighborhood.

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

jkt_merged = jkt_venues_top[jkt_venues_top.columns[0:3]].drop_duplicates()
jkt_merged.reset_index(drop = True, inplace = True)

# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
jkt_merged = jkt_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

jkt_merged.head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood 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 Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	SENAYAN	-6.226911	106.809920	0	Korean Restaurant	Café	Department Store	American Restaurant	Chinese Restaurant	Restaurant	Grocery Store	Sushi Restaurant	Bar	Coffee Shop
1	KEBON JERUK	-6.192572	106.769726	0	Noodle House	Auto Dealership	Concert Hall	Asian Restaurant	Indonesian Restaurant	Food Truck	Music School	Café	Coffee Shop	Convenience Store
2	KELAPA GADING TIMUR	-6.168293	106.904214	1	Noodle House	Food Truck	Chinese Restaurant	Indonesian Restaurant	Asian Restaurant	Steakhouse	Ice Cream Shop	Massage Studio	Food & Drink Shop	Farmers Market
3	TOMANG	-6.172725	106.797301	2	Convenience Store	Indonesian Restaurant	Café	Soup Place	Coffee Shop	Music Venue	Gym	Noodle House	Food Truck	Food Stand
4	PONDOK PINANG	-6.275479	106.780403	3	Food Court	Noodle House	Coffee Shop	Pizza Place	Indonesian Restaurant	Boutique	Park	Asian Restaurant	Convenience Store	Food Truck

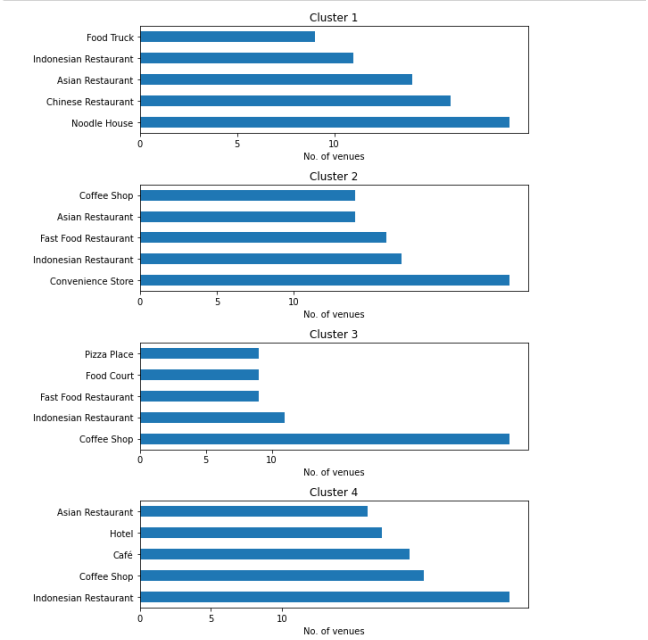
4. Result

The clustering is completely based on the most common venues obtained from Foursquare data.

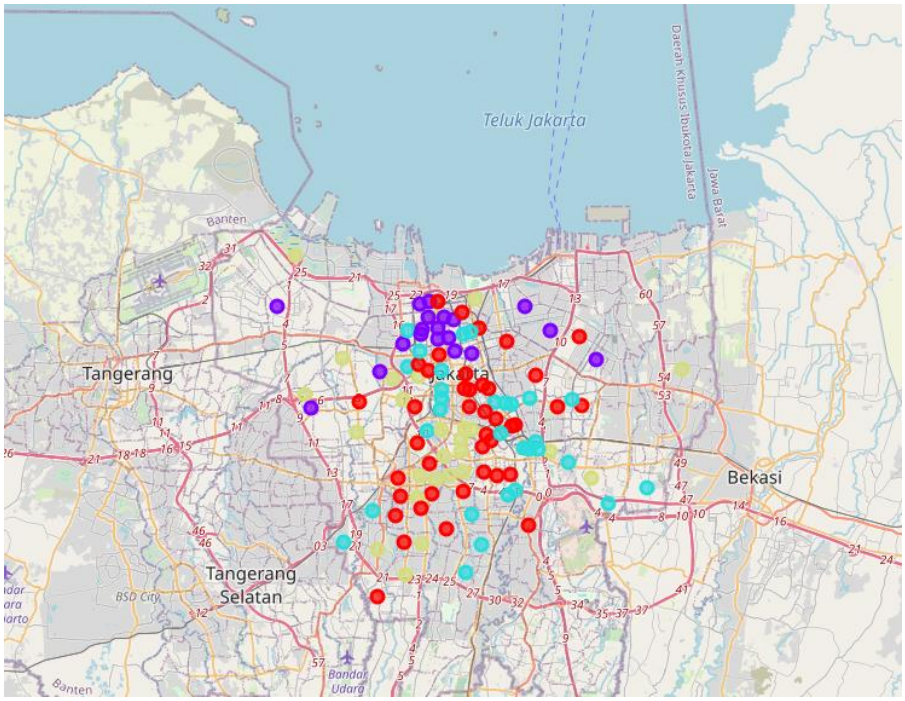
What we see in the table are the city districts and their most common venues, and they now have been assigned five different cluster labels from 0 to 4.

We got a glimpse of the Restaurants in Jakarta and were able to find out some interesting insights which might be useful to investors who plan to open a Business in Jakarta.

Lets visulaize the top 5 most common venue categories in each of the cluster.



We can represent these 4 clusters in a leaflet map using Folium library as below:



5. Conclusion

In the current digital age, there are many real-life problems/cases. We can find corresponding solutions by searching for data-analyzing data. As seen in the example above, the data content is based on the distribution of the most common dining places (restaurants) in the Jakarta neighborhoods. The results of the analysis can help investors determine the most suitable areas for investment.

I used some commonly used python libraries to extract web data, used the Foursquare API to explore the main areas of Jakarta, and used the Folium leaflet map to see the results of the region segmentation.

Similarly, data can also be used to solve other problems that most people face in large cities. The potential for such analysis in real life is discussed in detail. In addition, some shortcomings and improvement opportunities are mentioned to represent more realistic pictures.