Exploring Yogyakarta, Indonesia

Thomas Theo

thomas.t@outlook.co.id

1. Introduction

1.1. Background

Yogyakarta is arguably the cultural capital city of Indonesia, particularly Javanese culture. The city and province, are still ruled by a monarchy, an anomaly in the 21th century. Yogyakarta is home for plethora of incredible buildings, landmarks, and monuments. Borobudur temple, one of the Seven Wonders are located in the nearby city of Magelang; other famous places also includes, but not limited to: Prambanan temple, and Kraton Yogyakarta.

The city are also famous for its culinary heritage that's distinct from the other region, with food such as 'Gudeg', 'Bakpia', et cetera. One of the most famous food in from the city is called 'Bakpia'. Bakpia is a pastry filled with numerous filings, the classic and the most famous filling are green peas (mung beans), but other flavor and brand has innovate and produce numerous other fillings as well, e.g. chocolate, cheese, even durian. Bakpia is usually bought as a souvenirs from the city.

1.2. Objective

The research objectives are:

1) to find bakpia location in Yogyakarta;

2) explore trending venues in Yogyakarta so it can help tourist prioritize their trips.

2. Data sources and acquisition

The research use two main data sources:

1) Yogyakarta coordinates that's obtained from geopy.geocoders and Nominatim, where the result is latitude: -7.8011945, longitude: 110.364917;

2) Foursquare API using client id and client secret to explore trending venues in Yogyakarta and to find bakpia store in Yogyakarta.

3. Exploratory data analysis

1) Import all the necessary packages (including but not limited to):

a. requests;
b. pandas
c. numpy
d. geopy.geocoders
e. folium
f. et cetera

2) Find Yogyakarta latitudes and longitudes using geopy.geocoders and Nominati and display the map using Folium

Yogyakarta Location and Maps

In [2]:
```
# Yogyakarta Indonesia Longitude and Latitude

address = 'Yogyakarta'
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
lat = location.latitude
long = location.longitude
print(address + ' coordinates: {},{}.'.format(lat,long))
```

```
Yogyakarta coordinates: -7.8011945,110.364917.
```

In [3]:
```
map = folium.Map(location=[lat, long], zoom_start=15)
map
```

Out[3]:

3) Enter the Foursquare API using the format:

Foursquare API

In [4]:
```
CLIENT_ID = 'C1GZPKDDUHFF4ONSKUWUZJJVHJMW5JCYMXN2HUEK2HKJODLM' # your Foursquare ID
CLIENT_SECRET = 'PCSYFGSY1CQ4CVYWMMXJSDRYEFTORMTDCVT2IRV4DJM3HXKA' # your Foursquare
Secret
VERSION = '20180605'
LIMIT = 100
radius = 500
```

In [5]:
```
url = 'https://api.foursquare.com/v2/venues/search?&client_id={}&client_secret={}&v=
{}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    lat,
    long,
    radius,
    LIMIT)
url
```

Out[5]:
```
'https://api.foursquare.com/v2/venues/search?&client_id=C1GZPKDDUHFF4ONSKUWUZJJVHJMW
5JCYMXN2HUEK2HKJODLM&client_secret=PCSYFGSY1CQ4CVYWMMXJSDRYEFTORMTDCVT2IRV4DJM3HXKA&
v=20180605&ll=-7.8011945,110.364917&radius=500&limit=100'
```

4) Search Bakpia store location using search_query on Foursquare API

Check for Bakpia location in Yogyakarta

In [6]:
```
search_query = 'bakpia'
radius = 800
print(search_query + ' .... OK!')
```

```
bakpia .... OK!
```

In [7]:
```
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, lat, long, VERSION, search_query, radius, LIMIT)
url
```

Out[7]:
```
'https://api.foursquare.com/v2/venues/search?client_id=C1GZPKDDUHFF4ONSKUWUZJJVHJMW5JCYMXN2HUEK2HKJODLM&client_secret=PCSYFGSY1CQ4CVYWMMXJSDRYEFTORMTDCVT2IRV4DJM3HXKA&ll=-7.8011945,110.364917&v=20180605&query=bakpia&radius=800&limit=100'
```

In [8]:
```
results = requests.get(url).json()
results
```

Out[8]:
```
{'meta': {'code': 200, 'requestId': '5de79e7347b43d4c24255d45'},
 'response': {'venues': [{'id': '4d8c5649ac798cfad84629e4',
    'name': 'Bakpia Pathok 25',
    'location': {'address': 'Jl. AIP II KS Tubun No. 65',
     'crossStreet': 'Ngampilan',
     'lat': -7.79753955456098,
     'lng': 110.36126322017809,
     'labeledLatLngs': [{'label': 'display',
```

5) Analyze relevant the data gathered from Foursquare API and transform it into data frame to ease data analysis processes.

In [11]:
```
# assign relevant part of JSON to venues
venues=results['response']['venues']

# tranform venues into a dataframe
dataframe = json_normalize(venues)
dataframe.head()
```

Out[11]:

	id	name	categories	referralId	hasPerk	loca
0	4d8c5649ac798cfad84629e4	Bakpia Pathok 25	[{'id': '4bf58dd8d48988d1c7941735', 'name': 'S...	v-1575460458	False	Jl. A Tubı
1	4c31569c3896e21e3af7e690	Bakpia Patuk 75	[{'id': '4bf58dd8d48988d16a941735', 'name': 'B...	v-1575460458	False	Jl. A K.S.
2	4df6109f18a88611c6c537a9	Bakpia Pia Djogdja	[{'id': '4bf58dd8d48988d1c7941735', 'name': 'S...	v-1575460458	False	Jl. D

6) Clean the data and create a map out of it using Folium

In [14]:
```
venues_map = folium.Map(location=[lat, long], zoom_start=20) # generate map centred
around the Conrad Hotel

# add a red circle marker to represent the Conrad Hotel
folium.features.CircleMarker(
    [lat, long],
    radius=10,
    color='red',
    popup='Conrad Hotel',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(venues_map)

# add the Italian restaurants as blue circle markers
for lat, lng, label in zip(dataframe_filtered.lat, dataframe_filtered.lng, dataframe
_filtered.categories):
    folium.features.CircleMarker(
        [lat, lng],
        radius=5,
        color='blue',
        popup=label,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(venues_map)

# display map
venues_map
```

Out[14]:

7) Moving to the second objectives: to check for trending venues in Yogyakarta, still using Foursquare API

In [16]:
```
if len(results['response']['venues']) == 0:
    trending_venues_df = 'No trending venues are available at the moment!'

else:
    trending_venues = results['response']['venues']
    trending_venues_df = json_normalize(trending_venues)

    # filter columns
    columns_filtered = ['name', 'categories'] + ['location.distance', 'location.cit
y', 'location.postalCode', 'location.state', 'location.country', 'location.lat', 'lo
cation.lng']
    trending_venues_df = trending_venues_df.loc[:, columns_filtered]

    # filter the category for each row
    trending_venues_df['categories'] = trending_venues_df.apply(get_category_type, a
xis=1)
trending_venues_df
```

Out[16]: 'No trending venues are available at the moment!'

as you can see, the result is that there's no trending venues are available at the moment, thus when we tried to visualize a map using Folium:

In [17]:
```
if len(results['response']['venues']) == 0:
    venues_map = 'Cannot generate visual as no trending venues are available at the
moment!'

else:
    venues_map = folium.Map(location=[latitude, longitude], zoom_start=15) # generat
e map centred around Ecco


    # add Ecco as a red circle mark
    folium.features.CircleMarker(
        [latitude, longitude],
        radius=10,
        popup='Ecco',
        fill=True,
        color='red',
        fill_color='red',
        fill_opacity=0.6
    ).add_to(venues_map)


    # add the trending venues as blue circle markers
    for lat, lng, label in zip(trending_venues_df['location.lat'], trending_venues_d
f['location.lng'], trending_venues_df['name']):
        folium.features.CircleMarker(
            [lat, lng],
            radius=5,
            poup=label,
            fill=True,
            color='blue',
            fill_color='blue',
            fill_opacity=0.6
        ).add_to(venues_map)

venues_map
```

Out[17]: 'Cannot generate visual as no trending venues are available at the moment!'

Additional:

Check for review, tips, and comment at one of the Bakpia store:

Check for Review

(Bakpia Pathuk 25 as an example)

In [18]:
```
venue_id = '4d8c5649ac798cfad84629e4' # Bakpia Pathuk 25
url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(venue_id, CLIENT_ID, CLIENT_SECRET, VERSION)
url
```

Out[18]:
```
'https://api.foursquare.com/v2/venues/4d8c5649ac798cfad84629e4?client_id=C1GZPKDDUHFF4ONSKUWUZJJVHJMW5JCYMXN2HUEK2HKJODLM&client_secret=PCSYFGSY1CQ4CVYWMMXJSDRYEFTORMTDCVT2IRV4DJM3HXKA&v=20180605'
```

In [19]:
```
result = requests.get(url).json()
print(result['response']['venue'].keys())
result['response']['venue']
```

```
dict_keys(['id', 'name', 'contact', 'location', 'canonicalUrl', 'categories', 'verified', 'stats', 'url', 'price', 'likes', 'dislike', 'ok', 'rating', 'ratingColor', 'ratingSignals', 'allowMenuUrlEdit', 'beenHere', 'specials', 'photos', 'reasons', 'hereNow', 'createdAt', 'tips', 'shortUrl', 'timeZone', 'listed', 'popular', 'pageUpdates', 'inbox', 'attributes', 'bestPhoto', 'colors'])
```

Tips and review results extracted from Foursquare API

Out[24]:

	text	agreeCount	disagreeCount	id	user.firstName	user.lastName	user.gender	user.
0	Bakpia pathok 25 salah satu bakpia yg terkenal di yogyakarta, dgn harga RP 25 rb anda bisa dapet 1 kotak bakpia yg masih hangat.. lokasinya strategis atau anda bisa minta aner tukang becak ke sana...	4	0	50ddaaa3e4b0b8da1c30b572	aprizal ramadhan	NaN	male	1154

Translation: *Bakpia Pathok 25* (One of the Bakpia Brand), is very famous in Yogyakarta, with the price of IDR25k, you can get one freshly baked Bakpia. The location is strategic, you can reach it by rickshaw.

P.S. Full version of the code can be viewed at Python Notebook