Capstone Project - The Battle of Neighborhoods

Jingran Li

11/11/2018

Table of content

Introduction - Business Problem

- With the help of four square, people can learn which venue is the best option for them to eat dinner or have fun. Although venues on four square have been categorized into different filtering tags and scored by previous customers, people are still likely to review venues' tips. In other words, reviewing tips is a common method by which people can learn the venue thoroughly and from various perspectives.
- However, reading tips is very consuming because the tips are unstructured. It is not that easy to extract useful information from a bunch of tips quickly and correctly. In this project, a data visualization method – word cloud is applied so that the key information in tips can be presented in an effective and apparent way.

Introduction – target people

- People who will be interested in this project are the ones want to save time reading customers' tips. With word cloud figures, people can get an intuitive idea what advantages or disadvantages of the restaurant.

data

- **Data Source:** The data are collected from four square developer platform.
- **Data Description:** The data I will use in this project is the restaurants' tips in Manhattan.

DATA DESCRIPTION

- First, I will get the top 10 rated restaurants in Manhattan.
- I used to plan collecting the 10 most popular and 10 most recent tips for each venue. However, due to the limit of the tip request limit, I can only get two tips from each restaurant.
- Since there are photos in some tips, I will extract the text only and use the text for the following language processing.
- For each restaurant, i will generate the word cloud figure individually. From the word cloud figure, people can learn the advantages or disadvantages of the venue intuitively.

DATA COLLECTION & CLEANING

Get the latitude and longitude of Manhattan

```
address = 'Manhattan, NY'

geolocator = Nominatim()
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

Search 10 highly rated restaurants in Manhattan

```
search_query = 'Restaurant'
limit = 10
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&limit={}'.\
format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, search_query,limit)
results = requests.get(url).json()
# assign relevant part of JSON to venues
venues = results['response']['venues']

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

```
(10, 25)
```

DATA COLLECTION & CLEANING

Filter the name, category, and anything associated with location

```python
# keep only columns that include venue name, and anything that is associated with location
filtered_columns = ['name', 'categories'] + [col for col in dataframe.columns if col.startswith('location.')] + ['id']
dataframe_filtered = dataframe.loc[:, filtered_columns]

# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']

# filter the category for each row
dataframe_filtered['categories'] = dataframe_filtered.apply(get_category_type, axis=1)

# clean column names by keeping only last term
dataframe_filtered.columns = [column.split('.')[-1] for column in dataframe_filtered.columns]

dataframe_filtered
```

DATA COLLECTION & CLEANING

	name	categories	address	cc	city	country	crossStreet	distance	formattedAddress	labeledLatLngs	lat	lng	neighborhood	postalCode	state	id
0	Gabriela's Restaurant & Tequila Bar	Mexican Restaurant	688 Columbus Ave	US	New York	United States	at 93rd St.	761	[688 Columbus Ave (at 93rd St.), New York, NY 10025, United States]	[{'label': 'display', 'lat': 40.79123991711048, 'lng': -73.96873529559616}]	40.791240	-73.968735	NaN	10025	NY	49f3ab02f964a520d16a1fe3
1	Fred's Restaurant	American Restaurant	476 Amsterdam Ave	US	New York	United States	at W 83rd St	1494	[476 Amsterdam Ave (at W 83rd St), New York, NY 10024, United States]	[{'label': 'display', 'lat': 40.7855722, 'lng': -73.976527}]	40.785572	-73.976527	NaN	10024	NY	44281118f964a520ba311fe3
2	3 Guys Restaurant	Diner	49 E 96th St	US	New York	United States	Madison Ave	570	[49 E 96th St (Madison Ave), New York, NY 10128, United States]	[{'label': 'display', 'lat': 40.787442622504265, 'lng': -73.95403610873488}]	40.787443	-73.954036	NaN	10128	NY	4a897cb1f964a5201f0820e3
3	Junior's Restaurant & Bakery	American Restaurant	1515 Broadway	US	New York	United States	at W 45th St	4168	[1515 Broadway (at W 45th St), New York, NY 10036, United States]	[{'label': 'display', 'lat': 40.758539, 'lng': -73.986477}]	40.758539	-73.986477	Theater District	10036	NY	462a6065f964a520d9451fe3
4	Carmine's Italian Restaurant	Italian Restaurant	2450 Broadway	US	New York	United States	btwn W 90th & W 91st	1198	[2450 Broadway (btwn W 90th & W 91st), New York, NY 10024, United States]	[{'label': 'display', 'lat': 40.7910963, 'lng': -73.9739914}]	40.791096	-73.973991	NaN	10024	NY	4a7778a1f964a5209be41fe3
5	Demarchelier Restaurant	Bistro	50 E 86th St	US	New York	United States	at Madison Ave	1042	[50 E 86th St (at Madison Ave), New York, NY 10028, United States]	[{'label': 'display', 'lat': 40.780769874738176, 'lng': -73.9586131676082 3}]	40.780770	-73.958613	NaN	10028	NY	4a9037cef964a5209a1620e3
6	Tom's Restaurant	Diner	2880 Broadway	US	New York	United States	at W 112th St	1785	[2880 Broadway (at W 112th St), New York, NY 10025, United States]	[{'label': 'display', 'lat': 40.80549395837133, 'lng': -73.96571226552344}]	40.805494	-73.965712	NaN	10025	NY	415c9e00f964a520501d1fe3
7	Malecon Restaurant II	Latin American Restaurant	764 Amsterdam Ave	US	New York	United States	btw 97th St & 98th St	987	[764 Amsterdam Ave (btw 97th St & 98th St), New York, NY 10025, United States]	[{'label': 'display', 'lat': 40.79493159833159, 'lng': -73.96964755745924}]	40.794932	-73.969648	NaN	10025	NY	4a2eb2b0f964a52036981fe3
8	Carmine's Italian Restaurant	Italian Restaurant	200 W 44th St	US	New York	United States	btwn Broadway & 8th Ave	4280	[200 W 44th St (btwn Broadway & 8th Ave), New York, NY 10036, United States]	[{'label': 'display', 'lat': 40.7574973, 'lng': -73.9867788}]	40.757497	-73.986779	NaN	10036	NY	3fd66200f964a5209ee81ee3
9	The New Amity Restaurant	Diner	1134 Madison Ave	US	New York	United States	84th St.	1144	[1134 Madison Ave (84th St.), New York, NY 10028, United States]	[{'label': 'display', 'lat': 40.77980470462351, 'lng': -73.95966389381256}]	40.779805	-73.959664	NaN	10028	NY	4b282b9af964a520309024e3

DATA VISULIZATION

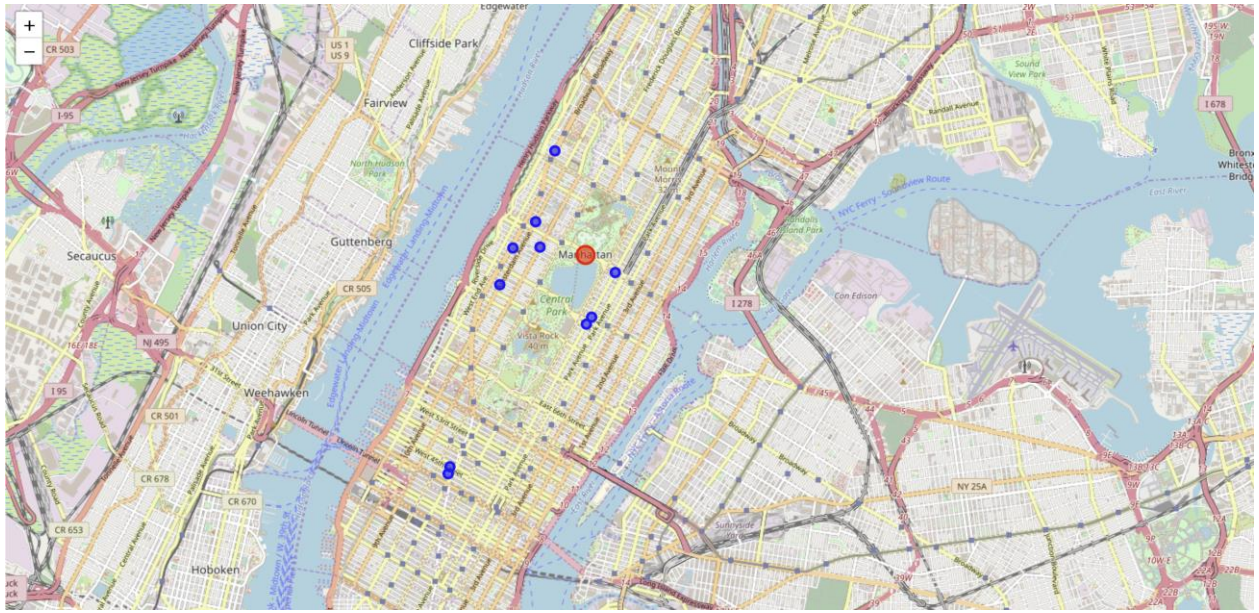
Generate map of Manhattan with markers of the 10 restaurants.

```
venues_map = folium.Map(location=[latitude, longitude], zoom_start=13)

# add a red circle marker to represent the Conrad Hotel
folium.features.CircleMarker(
    [latitude, longitude],
    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
```



WORD CLOUD

Build a function to extract the tip text

```
# function that extracts the tip text
def get_tip_text(venue_id):
    limit = 15 # set limit to be greater than or equal to the total number of tips
    url = 'https://api.foursquare.com/v2/venues/{}/tips?client_id={}&client_secret={}&v={}&limit={}'\
    .format(venue_id, CLIENT_ID, CLIENT_SECRET, VERSION, limit)
    results = requests.get(url).json()
    try:
        tips = results['response']['tips']['items']
        tips_df = json_normalize(tips) # json normalize tips
        # columns to keep
        filtered_columns = ['text']
        tips_filtered = tips_df.loc[:, filtered_columns]
        val_tot = ' '
        for val in tips_filtered['text']:
            val_tot = val_tot + str(val)
        tokens = val_tot.split()
        for i in range(len(tokens)):
            tokens[i] = tokens[i].lower()
        comment_words = ' '
        for words in tokens:
            comment_words = comment_words + words + ' '
    except:
        comment_words = ' '

    if len(comment_words) == 0:
        return None
    else:
        return comment_words
```

Plot the word cloud for each restaurant.

```
stopwords = set(STOPWORDS)
for name, venue_id in zip(dataframe_filtered['name'], dataframe_filtered['id']):
    comment_words = get_tip_text(venue_id)
    generate_word(comment_words)
    wordcloud = WordCloud(width = 800, height = 800,
                    background_color ='white',
                    stopwords = stopwords,
                    min_font_size = 10).generate(comment_words)

    # plot the WordCloud image
    plt.figure(figsize = (8, 8), facecolor = None)
    plt.imshow(wordcloud)
    plt.axis("off")
    plt.tight_layout(pad = 0)

    plt.savefig('name_tip_WordCloud'+name+'.png')
    plt.show()
```

DISCUSSION

- 3 Guys Restaurant:
- From the following figure, it can be observed that some keywords like “busy”, “pricey”, “primavera”, and “yum”. Thus, we can get a broad idea about this restaurant: food are yummy but a little pricy and the restaurant is always busy.