

Capstone Project - The Battle of Neighborhoods Final Report

Table of contents

- [Introduction/Business Problem](#)
- [Data](#)
- [Methodology](#)
- [Analysis](#)
- [Results and Discussion](#)
- [Conclusion](#)

Introduction: Business Problem

Should I take this new Job at Merida, Yucatan, Mexico? Would it be safe? What if I lose access to my favorite venues categories? What neighborhood should I choose?... These are some of the most common questions when it comes to deciding whether to relocate by accepting a new job.



In the following work, we are going to analyze the city of Merida Yucatan with its neighborhoods, the process will start with a list of categories that will be queried to FourSquare, then we will analyze the results to determine which would be the ideal neighborhood to move to.

With this method, we'll understand how similar or dissimilar the neighborhoods of a given city are. The same technique allows users to identify similar neighborhoods among cities based on amenities or services being offered locally, and thus can help in understanding the local area activities, the hubs of different activities, how citizens are experiencing the city, and how they are utilizing its resources.

Which audience will benefit from this analysis?

1. A person that may wish to move to a neighborhood that provide the most suitable match for his living style
2. A person renting a home in a new city and may want to look for recommendations for locations in the city similar to other cities.
3. An investor is eager to know how developed a certain niche or market is within the city.

Data

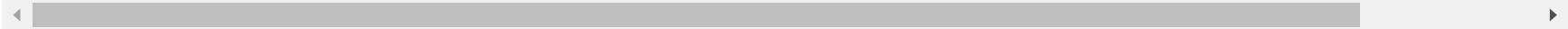
FourSquare location data

We are going to start from a list with the following categories:

- Rental Car Location
- Movie Theater
- Shopping Mall
- Supermarket
- Spa
- Boutique
- Hotel
- Travel agency
- Food
- Gym
- School

For each categorie in the list we will perform a venue recommendation, the following is a request URL with a partial JSON response as example:

[https://api.foursquare.com/v2/venues/explore?
client_id=#&client_secret=#&oauth_token=#&v=20180604&near=Merida,%20Yucatan&query=Rental%20Car%20Location&intent=browse&
\(https://api.foursquare.com/v2/venues/explore?
client_id=#&client_secret=#&oauth_token=#&v=20180604&near=Merida,%20Yucatan&query=Rental%20Car%20Location&intent=browse&](https://api.foursquare.com/v2/venues/explore?client_id=#&client_secret=#&oauth_token=#&v=20180604&near=Merida,%20Yucatan&query=Rental%20Car%20Location&intent=browse&https://api.foursquare.com/v2/venues/explore?client_id=#&client_secret=#&oauth_token=#&v=20180604&near=Merida,%20Yucatan&query=Rental%20Car%20Location&intent=browse&)



```
{"meta":{"code":200,"requestId":"608712c0e67ee12381b4011e"},"notifications":[{"type":"notificationTray","item":{"unreadCount":0}}],"response":{"suggestedFilters":{"header":"Tap to show:","filters":[{"name":"Open now","key":"openNow"}]}}, "geocode":
```

```
{"what": "", "where": "merida yucatan", "center": {"lat": 20.97537, "lng": -89.61696}, "displayString": "Mérida, YUC, Mexico", "cc": "MX", "geometry": {"bounds": {"ne": {"lat": 21.079218219110707, "lng": -89.54303352075961}, "sw": {"lat": 20.87298156415734, "lng": -89.72280315869435}}, "slug": "merida-mexico", "longId": "72057594041451285"}, "headerLocation": "Mérida", "headerFullLocation": "Mérida", "headerLocationGranularity": "city", "query": "car location", "totalResults": 20, "suggestedBounds": {"ne": {"lat": 21.0227280070603, "lng": -89.567665133385}, "sw": {"lat": 20.929645851733667, "lng": -89.67281047463536}}, "groups": [{"type": "Recommended Places", "name": "recommended", "items": [{"reasons": {"count": 0, "items": [{"summary": "This spot is popular", "type": "general", "reasonName": "globalInteractionReason"}]}], "venue": {"id": "4d95f28e942ba09389c3468c", "name": "Avis Renta de Autos", "location": {"address": "Calle 56-A Depto 1-A 451 Centro, AVENIDA COLON Y CUPULES", "crossStreet": "60", "lat": 20.986248, "lng": -89.619437, "labeledLatLngs": [{"label": "display", "lat": 20.986248, "lng": -89.619437}], "postalCode": "97000", "cc": "MX", "city": "Mérida", "state": "Yucatán", "country": "Mexico", "formattedAddress": "Calle 56-A Depto 1-A 451 Centro, AVENIDA COLON Y CUPULES (60)", "neighborhood": "97000 Mérida, Yucatán"}}]}
```

The reason why we are using "venues recommendation" instead of "venue search" is because this last one is limited to 50 results and it is important for this work to get as many venues as possible for a given category and not just the top 50.

Geopy

There are some results that do not have the zip code in the address field, but they still have the latitude and longitude in the Foursquare response, for those cases we will use Geopy passing the latitude and longitude in order to retrieve the zip code of the venue.

ETL process

```
In [1]: 1 city = 'Merida, Yucatan'  
2 categories = ["Rental Car Location", "Movie Theater", "Shopping Mall", "supermarket", "spa", "boutique", "hotel",  
3 "gas station", "gas bar", "gas pump", "gas station near me", "gas bar near me", "gas pump near me"]
```

Now we must set our FourSquare ID, secret and token so we can use the FourSquare API.

In [2]:

```

1 CLIENT_ID = '####' # your Foursquare ID
2 CLIENT_SECRET = '####' # your Foursquare Secret
3 ACCESS_TOKEN = '####' # your FourSquare Access Token
4 VERSION = '20180604'
5 print('Your credentails:')
6 print('CLIENT_ID: ' + CLIENT_ID)
7 print('CLIENT_SECRET:' + CLIENT_SECRET)

```

Your credentails:

```

CLIENT_ID: 2GH2JU3PYV3SJ1BWKW0IQQKV43GW1PZRH53CJS12C2SBATA
CLIENT_SECRET: TCFZCINTM2XDT233BDMOKGA0A301ZP2UKV3S4NUE04KIYL1E

```

In the FourSquare response, there are some venues that don't have the address information available, this is mandatory for our work as we must assign each venue with its corresponding neighborhood zip code. luckily for us, all the venues have their corresponding latitude and longitude values, that can be later transformed into an address using Geopy

In [3]:

```

1 from geopy.geocoders import Nominatim # module to convert an address into Latitude and Longitude values
2 import re # library to use regular expressions
3
4 #creating geolocator object
5 geolocator = Nominatim(user_agent="foursquare_agent")
6
7 #Defining the function that will transform any given Latitude and Longitude to an address with a zip code or
8 def coordinates_to_zipcode(lat,lng):
9     locator = Nominatim(user_agent="myGeocoder")
10    location = locator.geocode(lat+' '+lng)
11    #using re to findall regular expressions of five digits that means the zip code
12    try:
13        #print(location)
14        return(re.findall('97[0-9][0-9][0-9]',str(location))[0])
15    except:
16        return("NA")

```

Next we are going to create the function in charge of exploring each category for the given city.

```
In [4]: 1 def explore_city_venues_withoffset(CLIENT_ID, CLIENT_SECRET,ACCESS_TOKEN,VERSION,city,search_query):
2     url= 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&oauth_token={}&v={}&ne=
3     results = requests.get(url).json()
4     listofvenues = []
5     #getting the total number of results
6     if "totalResults" in results["response"].keys():
7         totalResults = results["response"]["totalResults"]
8         #getting the number of pages with 50 elements each
9         pagenumber = int(totalResults/50)
10        #print(totalResults, ", ",pagenumber+1)
11
12        for page in range(0,pagenumber+1):
13            url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&oauth_token={}&
14            #print(url)
15            results = requests.get(url).json()
16            for groups in results["response"]["groups"]:
17                for item in groups["items"]:
18                    vname = item["venue"]["name"]
19                    vid = item["venue"]["id"]
20                    catname = item["venue"]["categories"][0]["name"]
21                    lat=item["venue"]["location"]["lat"]
22                    lng=item["venue"]["location"]["lng"]
23                    zipcode = item["venue"]["location"]["postalCode"] if ("postalCode" in item["venue"]["lo
24                    try:
25                        zipcode = int(zipcode)
26                        if zipcode < 97000:
27                            zipcode = coordinates_to_zipcode(str(lat),str(lng))
28                    except:
29                        zipcode = coordinates_to_zipcode(str(lat),str(lng))
30
31
32                    listofvenues.append([vname,
33                                         vid,
34                                         catname,
35                                         zipcode,
36                                         lat,
37                                         lng])
38                    #print(item["venue"])
39                    #print(item["venue"]["name"], "/",item["venue"]["id"], "/",item["venue"]["categories"][0],
40
41                    return listofvenues
42
43 else:
44     print("No results")
```

Now it's time to call our functions for each category in order to create the dataframe

```
In [5]: 1 import requests # Library to handle requests  
2  
3 venueslist = []  
4 for cat in categories:  
5     venueslist = venueslist + explore_city_venues_withoffset(CLIENT_ID, CLIENT_SECRET, ACCESS_TOKEN, VERSION,
```

```
In [6]: 1 import pandas as pd # Library for data analysis  
2 dfvenues = pd.DataFrame(venueslist,columns=["name","id","cat","zip","lat","lng"])
```

We can see that our DataFrame was created, but we need to explore if there are any "NA" values, as a result of lack of information from Geopy

```
In [7]: 1 dfvenues.head()
```

Out[7]:

	name	id	cat	zip	lat	lng
0	Avis Renta de Autos	4d95f28e942ba09389c3468c	Rental Car Location	97000	20.986248	-89.619437
1	Mex Rent-A-Car	5a10386c5f68b97a20766a03	Rental Car Location	97255	20.934850	-89.664696
2	Avis Car Rental	4f94b0d9e4b06b0cf74cff10	Rental Car Location	97255	20.934127	-89.662862
3	Alamo	4f510e01e4b093e2bdb45e0f	Rental Car Location	97291	20.942641	-89.659737
4	Hertz	4dadcc39c43a18e54131d7b1a	Rental Car Location	97291	20.934291	-89.666352

```
In [8]: 1 dfvenues.shape
```

Out[8]: (906, 6)

In [9]: 1 dfvenues[dfvenues["zip"]=='NA']

Out[9]:

		name	id	cat	zip	lat	lng
99	NEXA Paseo Comercial	5650bc9b498ec68899a7f042		Shopping Mall	NA	20.981578	-89.556690
183	Chedraui	4ceabee5678aa093a61ff5ea		Supermarket	NA	20.999500	-89.573698
272	Promoda	52608e88498ea433916037c6		Boutique	NA	20.998675	-89.574040
444	hotel americana	53af66f1498ec970f3d136f3		Hotel	NA	20.755138	-88.743843
541	La Casa Del Kibi	4f67d5bbe4b0c15ca31d6f27	Fast Food Restaurant	NA	20.988277	-89.584451	
566	KIKE'S Panadería y Pastelería	4ea7553293adf18a2018bb60		Bakery	NA	20.990986	-89.577137
652	El rinconcito de xcalachén	4f1b0febe4b0a62718a6bc7a		Taco Place	NA	20.998889	-89.566854
677	Taqueria los primos	4feefae3e4b0c3c46da588c2	Mexican Restaurant	NA	20.996842	-89.571657	
709	California Fitness	59ed40432632ec468fc12848	Gym / Fitness Center	NA	20.997960	-89.574621	
756	NITRO GYM	518991c3498ebff9c2aac551		Gym	NA	20.991098	-89.579303
828	Unidad Deportiva del Sur	4e32d4c92271f49bf792121b		Track	NA	20.900158	-89.656070
829	La Strada	55fb615b498ed7002a3b2c6d		Cycle Studio	NA	21.000766	-89.573011

We note there are a few venues that were not able to resolve the zip code with Geopy, in this case, the information is not considered significant and will be ignored

In [10]: 1 dfvenues = dfvenues[dfvenues['zip']!='NA']
2 dfvenues.shape

Out[10]: (894, 6)

Methodology

In this project, we will direct our efforts on detecting as many venues in Merida City for the required categories. We will use the "explore" Venues Recommendation endpoint to overcome the limited 50 results that the "search" endpoint has.

With the results from FourSquare, we have noted that more than 500 venues didn't include the Zipcode, for that reason we pass the latitude and longitude of these venues to Geopy in order to recover this information.

We will plot the venues in the Meridas Map using Folium to start having an idea of the venues' distribution. Then we will group all venues by Zipcode get some hidden information to discover which neighborhood has the most venues categories, then we will work only with the top 20 neighborhoods.

At the end of our work with the data from <https://micodigopostal.org/yucatan/merida/> (<https://micodigopostal.org/yucatan/merida/>), and we will translate the zipcodes to neighborhood names so our clients can make a decision.

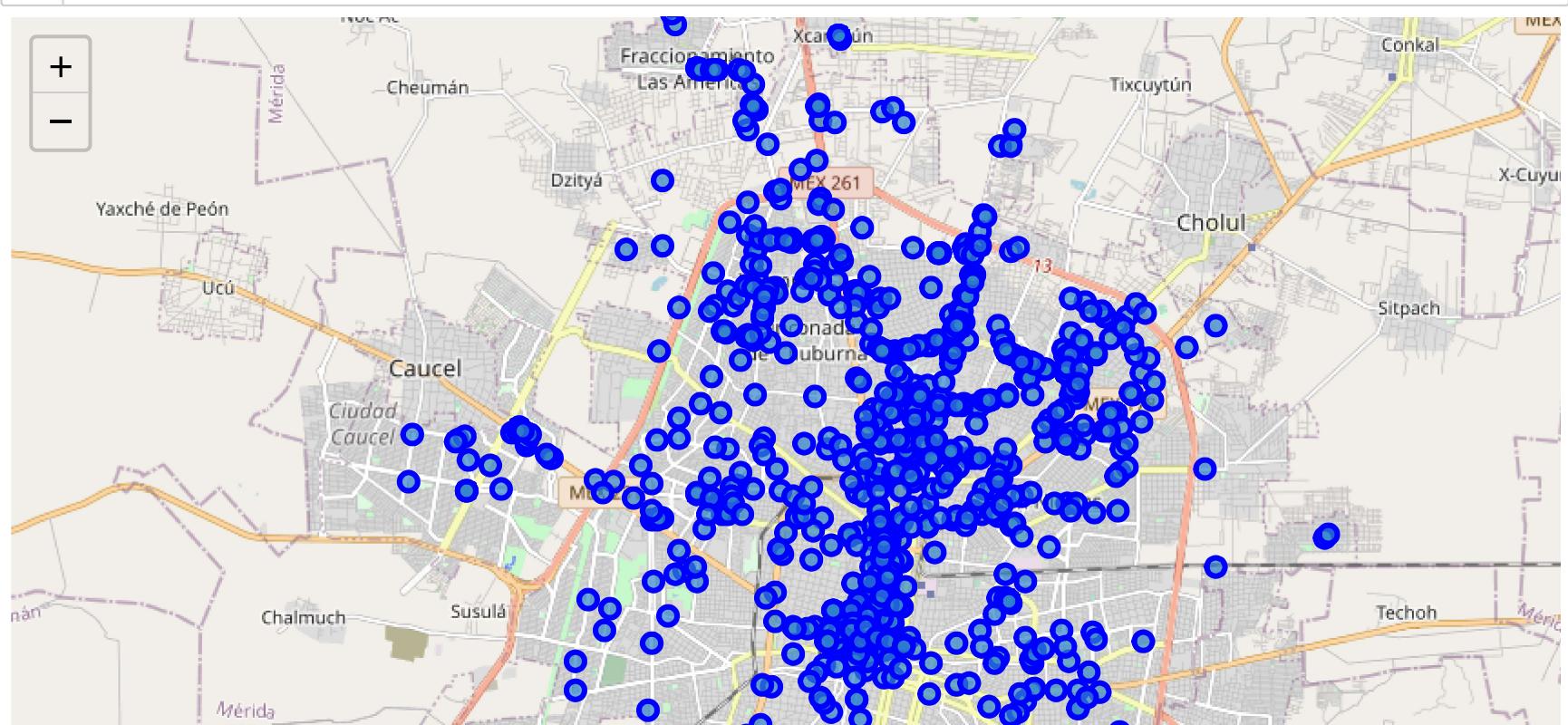
Analysis

```
In [11]: 1 dfvenues['zip'] = pd.to_numeric(dfvenues['zip'])
```

In [12]:

```
1 import folium # plotting library
2
3 map_merida = folium.Map(location=[20.9741254,-89.6360881], zoom_start=12)
4
5 # add markers to map
6 for lat, lng, name, cat in zip(dfvenues['lat'], dfvenues['lng'], dfvenues['name'], dfvenues['cat']):
7     label = '{} | {}'.format(name,cat)
8     label = folium.Popup(label, parse_html=True)
9     folium.CircleMarker(
10         [lat, lng],
11         radius=5,
12         popup=label,
13         color='blue',
14         fill=True,
15         fill_color='#3186cc',
16         fill_opacity=0.7,
17         parse_html=False).add_to(map_merida)
18
19 map_merida
```

Out[12]:





In the map above we observe that the south of the city is not as developed as the north, this gives us a clear reference of how different the quality of life can be in these two regions. This translates into lower rents in the south, but also a lack of services.

```
In [156]: 1 dfvenuesstop = dfvenues.groupby(['zip']).count().sort_values(by=['cat'], ascending=False).head(20)
2 topneighs = dfvenuesstop.index.tolist()
```

```
In [157]: 1 print(topneighs)
```

```
[97000, 97117, 97203, 97125, 97100, 97138, 97314, 97120, 97130, 97133, 97115, 97135, 97110, 97119, 97325, 97139, 97302, 97070, 97206, 97118]
```

```
In [126]: 1 zipmerida = pd.read_csv("zip-merida.csv", header=0, encoding='latin-1')
```

```
In [120]: 1 zipmerida.head()
```

Out[120]:

	name	type	zip	city
0	15 de Mayo	Fraccionamiento	97229	Mérida
1	5 Colonias	Colonia	97280	Mérida
2	Águilas Chuburna	Fraccionamiento	97215	Mérida
3	Álamos del Sur	Fraccionamiento	97285	Mérida
4	Alcalá Martín	Colonia	97050	Mérida

In [159]:

```
1 for zipcode in topneighs:
2     try:
3         #topneigh[topneigh["zip"]==zipcode]['name'] =
4         print(zipcode,zipmerida[zipmerida['zip']==zipcode]['name'].to_string(index=False),"\n")
5     except:
6         topneigh[topneigh["zip"]==zipcode]['name']='NA'
7     print("-----")
```

Puesta del Sol
Rincón Colonial
San José Chuburna
San Pablo
San Pedro Uxmal
Terranova
Tulias de Chuburna
Villas Del Prado
Villas Palma Real
Vista Alegre de Chuburna

97125 México
Privada Nuevo México

97100 Itzimna
Itzimna
Itzimna 2
Rinconada Ttzmina

Results and Discussion

Our analysis shows that there is a very marked difference between the south of the city with respect to the center and north of it. This may translate as an opportunity for any investor to develop business in that region as it has not been exploited, it may also indicate a limited amount of income from the population.

The results show that the downtown area is the area with the most commercial development, but it still implies that rents can be expensive, if the client would like to have the largest amount of commercial premises to which he is accustomed, he can choose any of the neighborhoods neighboring the center obtained from our analysis, such as:

- San Ramon Norte

- San Ramon Norte I
- San Ramon Sur
- Villareal
- Xaman-Kab
- Ampliación Francisco de Montejo
- Arboledas Chuburna
- Arcos del Sol
- Arekas
- Aurea Residencial
- Brisas de Chuburna
- Camara de La Construcción
- Cocoteros
- Cordeleros de Chuburna
- Del Bosque
- El Conquistador
- El Prado
- Francisco de Montejo
- Francisco de Montejo II
- Francisco de Montejo III
- Francisco de Montejo IV
- Francisco de Montejo V
- La Castellana
- Las Haciendas III
- Magnolias
- Platino
- Privada Chuburna de Hidalgo (II)
- Privada Chuburna Plus
- Privada Corozal
- Privada Palma Caribeña
- Puesta del Sol
- Rincón Colonial
- San José Chuburna
- San Pablo
- San Pedro Uxmal
- Terranova
- Tulias de Chuburna
- Villas Del Prado

- Villas Palma Real
- Vista Alegre de Chuburna

Conclusion

The purpose of this work was to determine if the city of Merida Yucatan had venues from a category appropriate to the needs of an employee seeking to relocate to work in this city.

The results obtained showed a good amount of developments and allowed us to obtain the recommended neighborhoods to live in from the list of the top 20 neighborhoods with the most venues available.

It is important to mention that the results are based on the information available from FourSquare, it is very likely that many businesses have not been registered on this platform but we got more than 900 venues distributed in the city, which is a good amount of data to provide a good picture of current status.

In []:

1