Different City Similarities

1 Introduction

In the previous project, we have used the K-Mean model to cluster Toronto's neighborhoods based on the classification of venues near the neighborhood. In such projects, our classification goals are limited to the same city. Now let’s take a longer-term view and continue to explore the similarity between different cities. For example, a person who dislikes change very much must move out of his city for some irresistible reasons. Instead of living in another city, how would he choose the area of his new home?

This article will start with the two simplest cities and try to find similar neighborhoods between cities, not just the same city. And continue our attempts to discuss the relationship between some well-known cities around the world based on similarity, and explore the types of different cities based on country classification. In this way, people who have the courage to try new things can give cities similarities to find different life experiences. At the same time, people who are forced to move can also live in neighborhoods that make them feel comfortable.

2 Data

Our research is still based on geographic location and we send requests to Foursquare API to get more detailed reports.

We chose New York and Los Angeles as the starting point for the discussion. The location information of urban neighborhoods is obtained from Neighborhood Data for Social Change. By reading the website information of https://usc.data.socrata.com/, the location coordinates of the neighborhoods of New York and Los Angeles can be obtained. However, due to some unavoidable reasons, the data we obtained still needs to be cleaned up in advance, such as deleting redundant redundancy we don’t need, correcting obvious errors, and making up for the limited data. It is then integrated into a Dataframe that is easy to read and understand. The final data processing step also encodes the already integrated data, so that the text data is converted into digital data that is easy for the computer to process. And finally feed the model to get the results we want.

In the next stage of discussion, we want to make an inventory of large cities around the world. To this end, we selected the research report of the Globalization and World Cities Research Network as a third-party organization that we can trust and obtain global city data of interest from there. This continues to obtain venue information in different cities through the Foursquare API. To be able to analyze the similarity between them.

3 Methodology

This research report starts with the analysis of the two simplest cities, and gradually expands while thinking about the feasibility and improvement of the research done. Finally, the result of global city similarity is generated.

3.1 New York and Los Angeles Neighborhood Analysis

In this part, we will study two cities, New York and Los Angeles, integrate their neighborhood information, and combine different venue categories to integrate the neighborhoods of the two cities and find similar neighborhoods.

First, we send a request to The Neighborhood Data for Social Change (NDSC) to obtain the required location information for New York and Los Angeles neighborhoods. At the same time, get the theme coordinates of the two cities from geolocator.

```
print(df_LA.shape)
df_LA.head()
```

```
(272, 3)
```

	Neighborhood	Latitude	Longitude
0	LA_Acton	34.497355	-118.169810
1	LA_Adams-Normandie	34.031461	-118.300208
2	LA_Agoura Hills	34.146736	-118.759885
3	LA_Agua Dulce	34.504927	-118.317104
4	LA_Alhambra	34.085539	-118.136512

```
print(df_NY.shape)
df_NY.head()
```

```
(306, 3)
```

	Neighborhood	Latitude	Longitude
0	NY_Wakefield	40.894705	-73.847201
1	NY_Co-op City	40.874294	-73.829939
2	NY_Eastchester	40.887556	-73.827806
3	NY_Fieldston	40.895437	-73.905643
4	NY_Riverdale	40.890834	-73.912585

LA and NY city coordinates

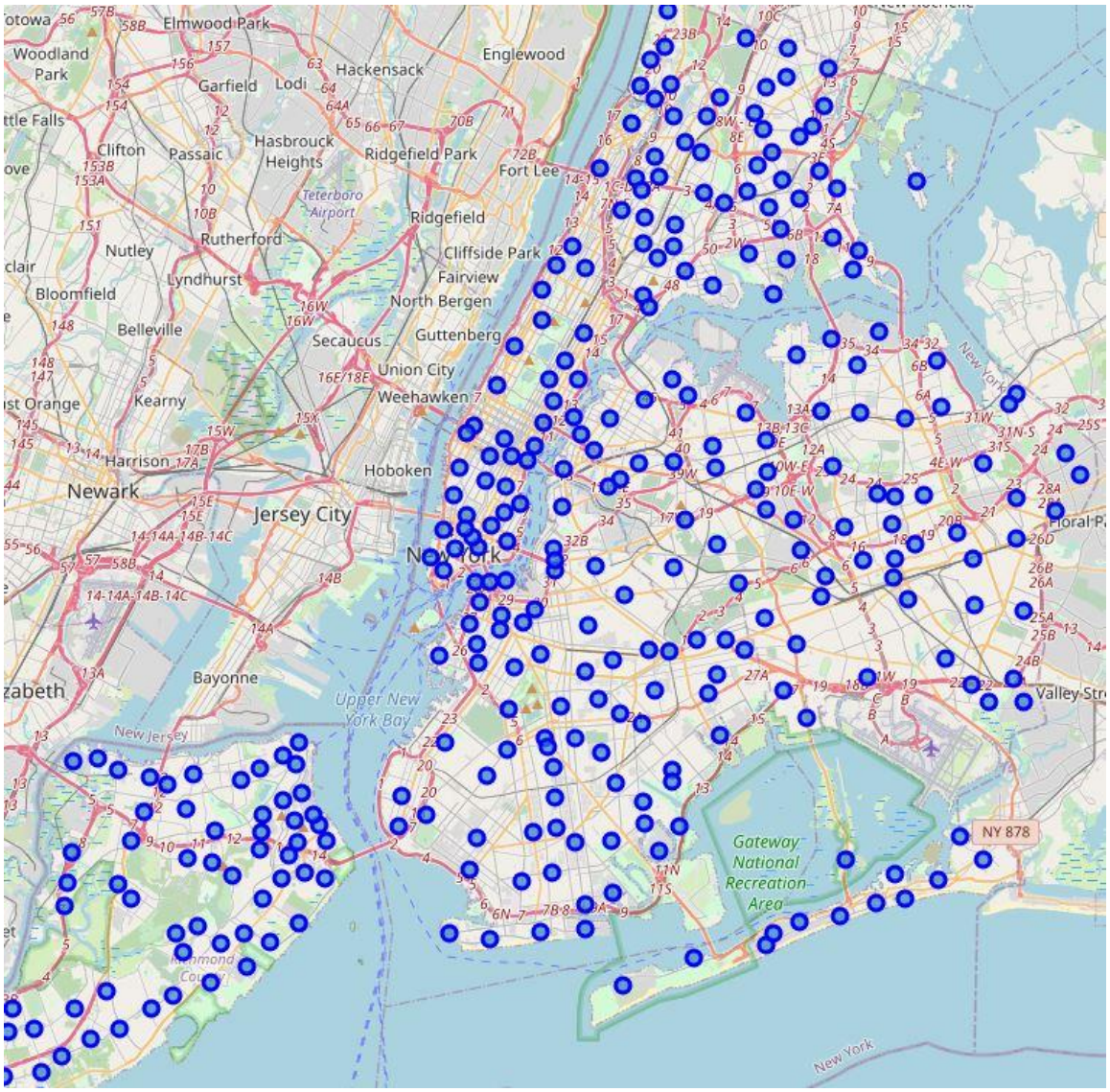
```
address = 'New York, NY, US'
latitude_NY,longitude_NY = get_city_coordinate(address)
print('The geograpical coordinate of NY are {}, {}.'.format(latitude_NY, longitude_NY))
```

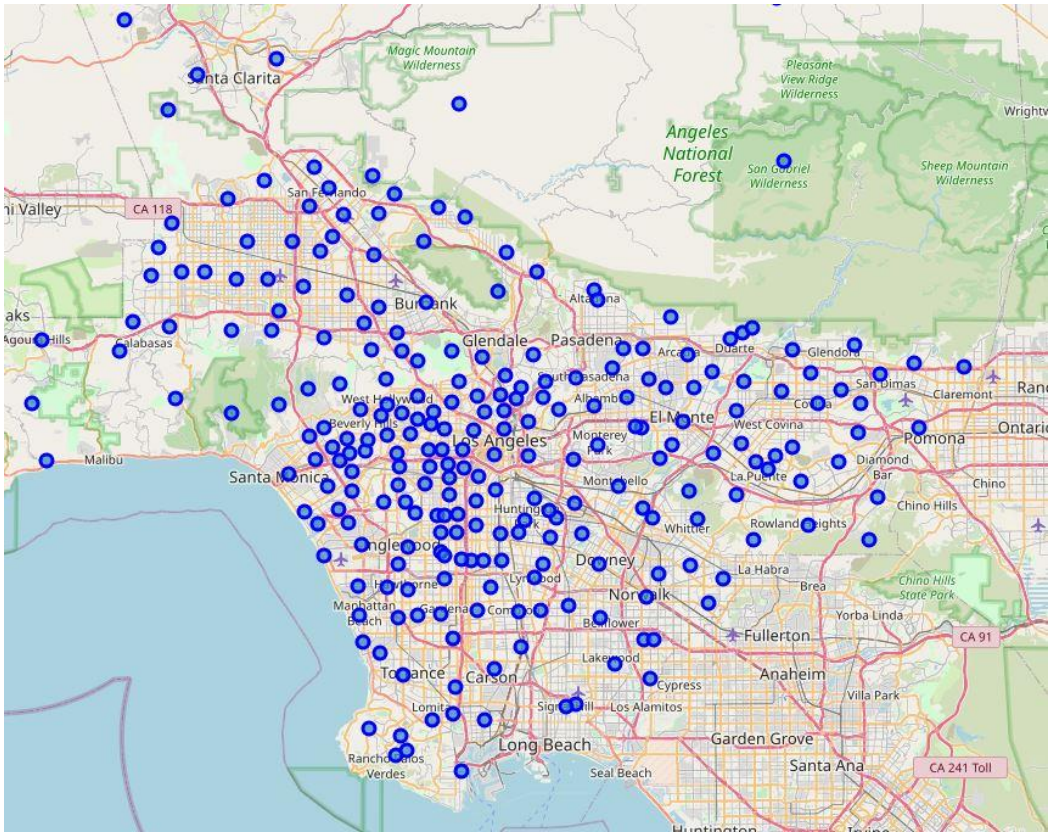
```
The geograpical coordinate of NY are 40.7127281, -74.0060152.
```

```
address = 'Los Angeles, CA, US'
latitude_LA, longitude_LA = get_city_coordinate(address)
print('The geograpical coordinate of LA are {}, {}.'.format(latitude_LA, longitude_LA))
```

```
The geograpical coordinate of LA are 34.0536909, -118.2427666.
```

Then based on the neighborhood coordinates of the two cities, we can first visually see their neighborhood on the map.


Elmwood Park
Woodland Park
Englewood
Hackensack
Garfield
Lodi
Clifton
Passaic
Hasbrouck Heights
Ridgefield Park
Fort Lee
Teterboro Airport
Ridgefield
Rutherford
Nutley
Lyndhurst
Cliffside Park
Fairview
Bloomfield
North Bergen
Belleville
Secaucus
Guttenberg
Union City
Weehawken
Kearny
Harrison
Hoboken
Newark
Jersey City
New York
Bayonne
Upper New York Bay
New Jersey
Gateway National Recreation Area
NY 878
New York



Get information about venues near the neighborhood by sending a request to Foursquare. Following the venues correspond neighborhoods for New York and Los Angeles.

```
print(neighbor_venues_NY.shape)
neighbor_venues_NY.drop_duplicates(subset=['Venue Id'], inplace=True)
print(neighbor_venues_NY.shape)
neighbor_venues_NY.head()
```

```
(10090, 8)
(9553, 8)
```

	Neighborhood	Latitude	Longitude	Venue Id	Venue Name	Venue Latitude	Venue Longitude	Venue Category
0	NY_Wakefield	40.894705	-73.847201	4c537892fd2ea593cb077a28	Lollipops Gelato	40.894123	-73.845892	Dessert Shop
1	NY_Wakefield	40.894705	-73.847201	5d5f5044d0ae1c0008f043c3	Walgreens	40.896528	-73.844700	Pharmacy
2	NY_Wakefield	40.894705	-73.847201	4c783cef3badb1f7e4244b54	Carvel Ice Cream	40.890487	-73.848568	Ice Cream Shop
3	NY_Wakefield	40.894705	-73.847201	4d6af9426107f04dedeb297a	Rite Aid	40.896649	-73.844846	Pharmacy
4	NY_Wakefield	40.894705	-73.847201	4c25c212f1272d7f836385c5	Dunkin'	40.890459	-73.849089	Donut Shop

```
print(neighbor_venues_LA.shape)
neighbor_venues_LA.drop_duplicates(subset=['Venue Id'], inplace=True)
print(neighbor_venues_LA.shape)
neighbor_venues_LA.head()
```

```
(2957, 8)
(2953, 8)
```

	Neighborhood	Latitude	Longitude	Venue Id	Venue Name	Venue Latitude	Venue Longitude	Venue Category
0	LA_Acton	34.497355	-118.169810	5a326a84345cbe5681ec1aaf	Epik Engineering	34.498718	-118.168046	Construction & Landscaping
1	LA_Acton	34.497355	-118.169810	549fa7db498e5892fb9a8275	Alma Gardening Co.	34.494762	-118.172550	Construction & Landscaping
2	LA_Adams-Normandie	34.031461	-118.300208	5498d200498e8153c17c751a	Orange Door Sushi	34.032485	-118.299368	Sushi Restaurant
3	LA_Adams-Normandie	34.031461	-118.300208	5cdb759f15173e002c3af5a2	Shell	34.033095	-118.300025	Gas Station
4	LA_Adams-Normandie	34.031461	-118.300208	4d0825e043b36ea8189f2bef	Little Xian	34.032292	-118.299465	Sushi Restaurant

We see that the total number of venues has decreased in both cities after deduplication, because the search radius of different neighborhoods will overlap. But here we do not deal with overlap, because this is itself a characteristic of the neighborhood. However, it is necessary to bear this in mind, and deduplication will be an important step in the subsequent city-based research.

Here we can also compare all the venues in the two cities, and what are the frequent venues will have a great influence on the classification of the city. These are Top 15 categories of each city.

```
Venue_Category_sum_LA[:15]
```

Venue Category	Number
Mexican Restaurant	135
Pizza Place	92
Coffee Shop	87
Fast Food Restaurant	77
Grocery Store	70
Bakery	67
Convenience Store	66
Park	62
Chinese Restaurant	58
Sandwich Place	58
Café	47
Pharmacy	46
Burger Joint	44
American Restaurant	42
Clothing Store	39

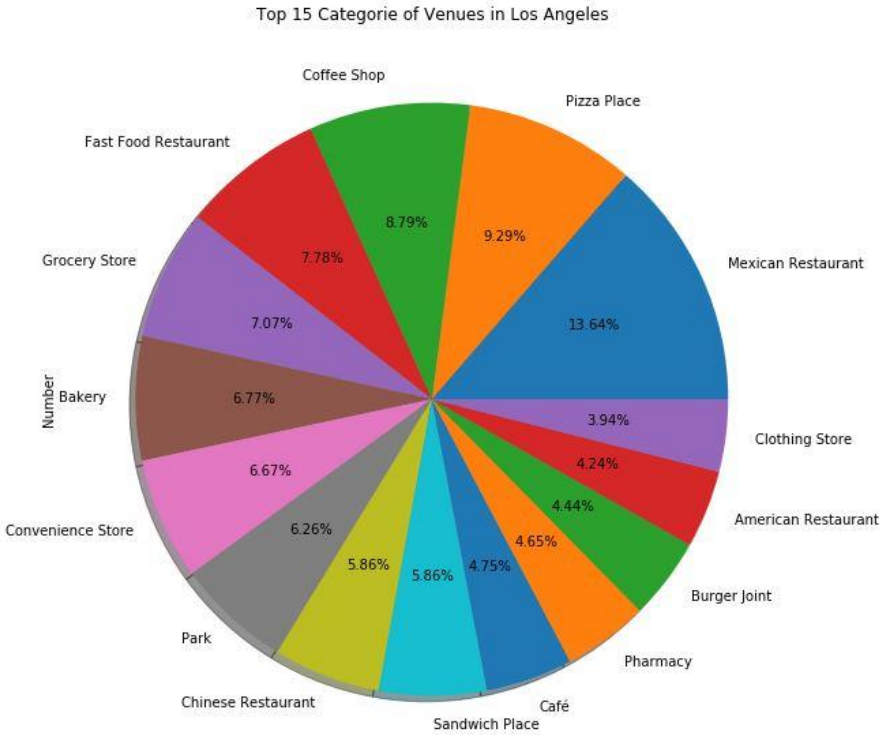
```
Venue_Category_sum_NY[:15]
```

Venue Category	Number
Pizza Place	421
Italian Restaurant	298
Coffee Shop	285
Deli / Bodega	260
Bakery	216
Bar	215
Chinese Restaurant	202
Sandwich Place	179
Grocery Store	176
Mexican Restaurant	173
Pharmacy	171
Park	170
Donut Shop	163
Café	162
American Restaurant	147

And we can also see that in figure.

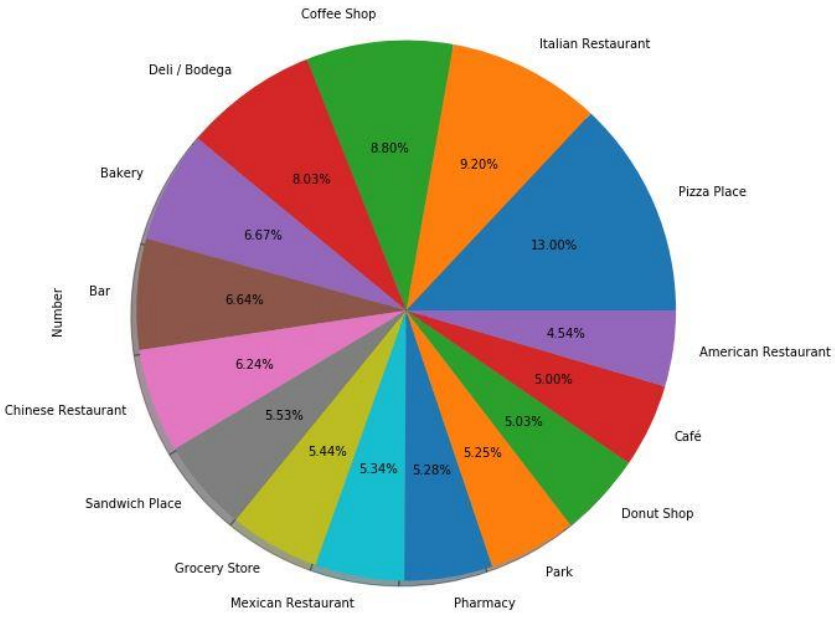
```
Venue_Category_sum_LA[:15].plot(kind='pie',subplots=True,figsize=(25,10),autopct='%1.2f%%',legend=None,
                                shadow=True)
plt.title('Top 15 Categorie of Venues in Los Angeles')
```

```
Text(0.5, 1.0, 'Top 15 Categorie of Venues in Los Angeles')
```




```
Venue_Category_sum_NY[:15].plot(kind='pie',subplots=True,figsize=(25,10),autopct='%1.2f%%',legend=None,
                                shadow=True)
plt.title('Top 15 Categorie of Venues in New York')
```

```
Text(0.5, 1.0, 'Top 15 Categorie of Venues in New York')
```



Next, we apply One-Hot encoding and group the neighborhoods, then get the neighborhood feature that is program readable to feed to our K-Mean clustering model.

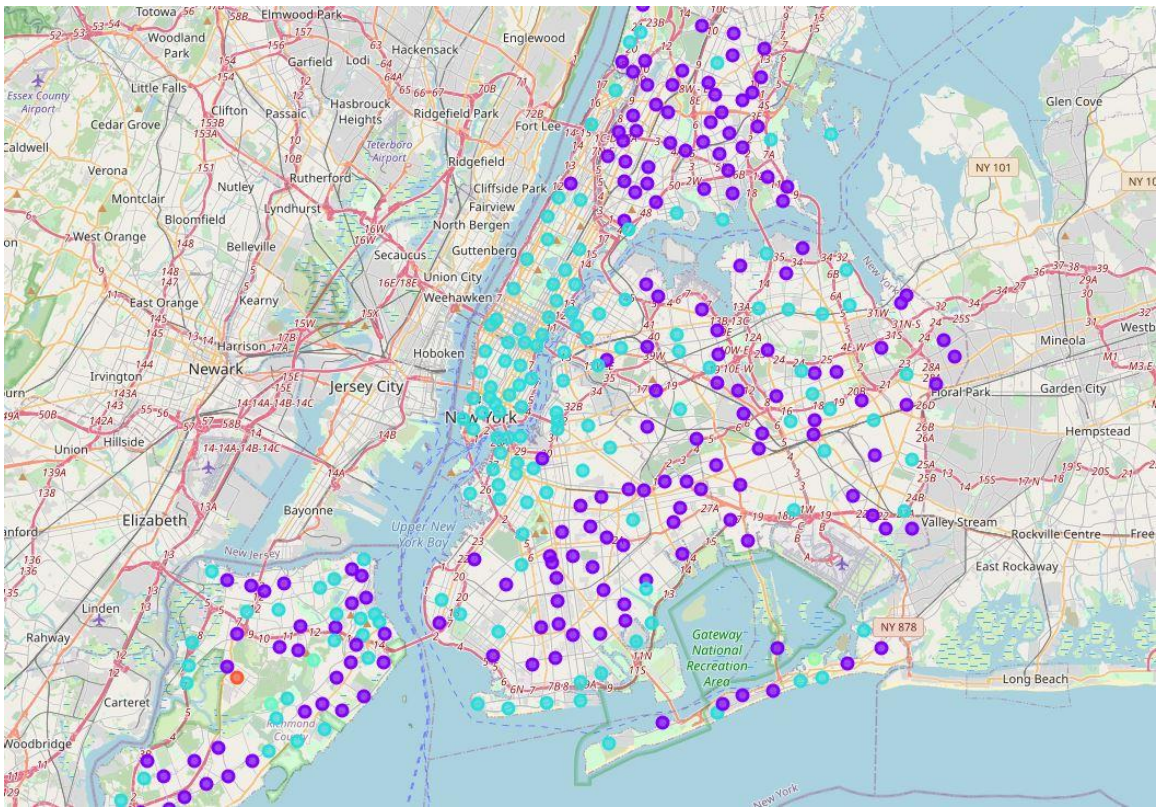
Group neighborhood and get neighborhood features

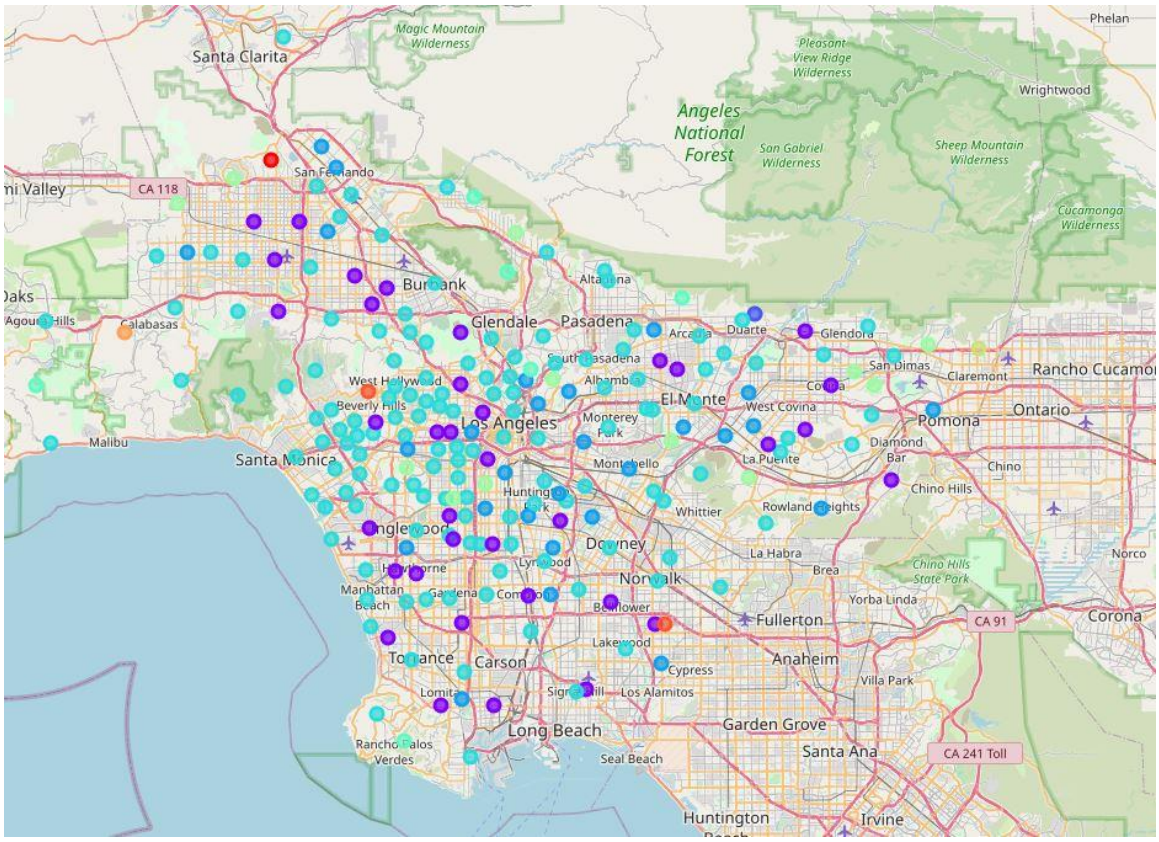
```
cities_grouped = onehot_neighbor_venues_comb.groupby('City_Neighborhood').mean().reset_index()
print(cities_grouped.shape)
cities_grouped.head()
```

```
(537, 466)
```

	City_Neighborhood	ATM	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	Airport	Airport Terminal	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	E
0	LA_Acton	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	LA_Adams-Normandie	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	LA_Agoura Hills	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.038462	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	LA_Agua Dulce	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	LA_Alhambra	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Then K-Mean result will classify neighborhood from the two city in to our pre-defined 10-clusers. As we find from the pie-chart, most of the top categories from the two city are same, and we expect the clusters are distributed the same. Let's see them below.





The same color of points means they are in the same group. As we read from the plot, most neighborhoods are classified in 2 groups, the purple group and the blue group, even we defined 10 clusters.

Now we have our intuitive idea that New York and Los Angeles are relatively same. We want to verify our suppose next.

3.2 New York and Los Angeles City Analysis.

From the city coordinates we obtained from geolocater, we can send request to Foursquare and get the venues for each city.

```
print(venues_la.shape)
venues_la.head()
```

```
(100, 7)
```

	City	Ciry Latitude	City Longitude	Venue Name	Venue Latitude	Venue Longitude	Venue Category
0	LA	34.053691	-118.242767	Walt Disney Concert Hall	34.055511	-118.249284	Concert Hall
1	LA	34.053691	-118.242767	The Last Bookstore	34.047620	-118.249852	Bookstore
2	LA	34.053691	-118.242767	Grand Central Market	34.050675	-118.248741	Market
3	LA	34.053691	-118.242767	Mr. Speedy Plumbing & Rooter Inc.	34.042538	-118.233864	Home Service
4	LA	34.053691	-118.242767	Vista Hermosa Park	34.061601	-118.256857	Park

```
print(venues_ny.shape)
venues_ny.head()
```

```
(100, 7)
```

	City	Ciry Latitude	City Longitude	Venue Name	Venue Latitude	Venue Longitude	Venue Category
0	NY	40.712728	-74.006015	Aire Ancient Baths	40.718141	-74.004941	Spa
1	NY	40.712728	-74.006015	9/11 Memorial North Pool	40.712077	-74.013187	Memorial Site
2	NY	40.712728	-74.006015	Crown Shy	40.706187	-74.007490	Restaurant
3	NY	40.712728	-74.006015	One World Trade Center	40.713069	-74.013133	Building
4	NY	40.712728	-74.006015	The Rooftop @ Pier 17	40.705463	-74.001598	Music Venue

Then follow the previous steps, we apply one-hot encoding then group by neighborhoods to feed them to our model. Then we get the city similarities.

```
feature_ny, feature_la = cityFeatures(onehot_ny, onehot_la)
```

```
feature_la
```

	City	Yoga Studio	Wine Shop	Bakery	Sandwich Place	Salon / Barbershop	Park	French Restaurant	Movie Theater	Pizza Place	Deli / Bodega	Trail	Coffee Shop	Theater
0	LA	0.02	0.02	0.02	0.04	0.01	0.04	0.01	0.01	0.04	0.03	0.06	0.04	0.02

```
feature_ny
```

	City	Yoga Studio	Wine Shop	Bakery	Sandwich Place	Salon / Barbershop	Park	French Restaurant	Movie Theater	Pizza Place	Deli / Bodega	Trail	Coffee Shop	Theater
0	NY	0.02	0.02	0.02	0.02	0.01	0.18	0.01	0.02	0.03	0.02	0.01	0.01	0.03

```
feature_similarity(feature_ny, feature_la)
```

```
0.3071258285846134
```

We can also write them in one function for easy call.

```
city_similarity(['NY',latitude_NY,longitude_NY],['LA',latitude_LA,longitude_LA])
```

```
The similarity of city NY and city LA is 0.3071258285846134.

0.3071258285846134
```

Finally, we compare several US cities to compare the similarities.

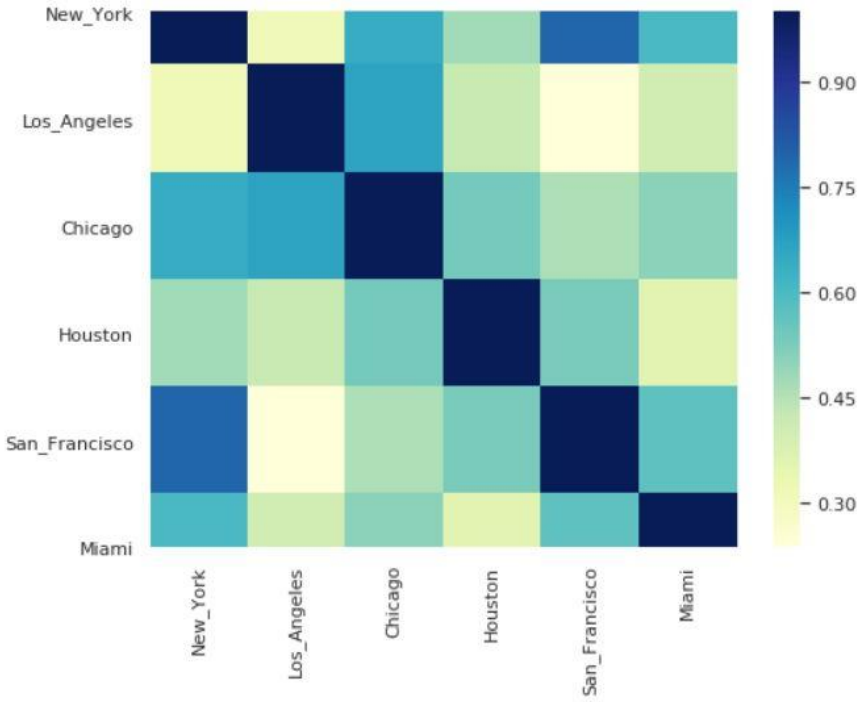
```
New_York = ['NY',40.71274, -74.005974]
Los_Angeles = ['LA',34.05, -118.25]
Chicago = ['Chicago', 41.881944, -87.627778]
San_Francisco = ['San Francisco',37.7775, -122.416389]
Miami = ['Miami',25.775278, -80.208889]
Houston = ['Houston',29.762778, -95.383056]

Cities = [New_York, Los_Angeles, Chicago, Houston, San_Francisco, Miami]
```

For easy visualization, we use heat plot.

```
label_city = ['New_York', 'Los_Angeles', 'Chicago', 'Houston', 'San_Francisco', 'Miami']
Matrix_us_city = pd.DataFrame(Matrix, columns=label_city, index=label_city)
```

```
import seaborn as sns
sns.set(rc={'figure.figsize':(8,6)})
ax = sns.heatmap(Matrix_us_city, cmap="YlGnBu")
```



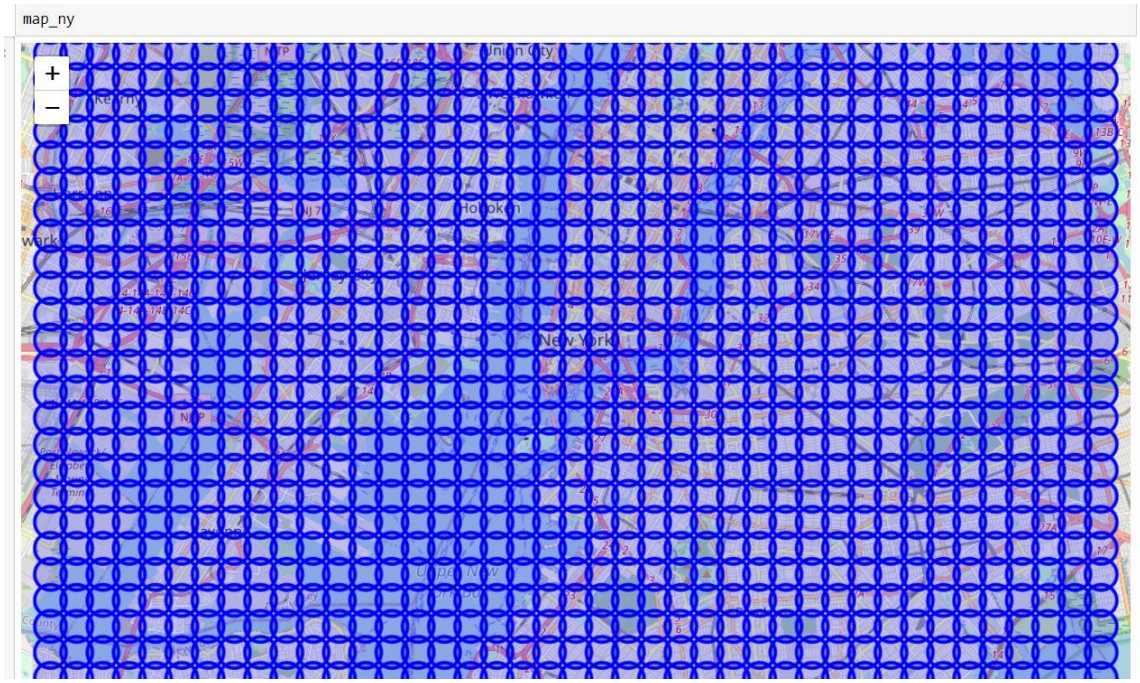
Here a question raise: we see from 3.1 that New York and Los Angeles should perform much more similar then we obtained from the heat plot.

The answer might be this: The Foursquare API is personal free, the limit of the request is 500 radius and 100 venues. Thus, our searched venues are only small part of the whole city, we need to find a method to coverage the whole city and obtain as many as venues from the city. So that the category is enough to encoding the feature.

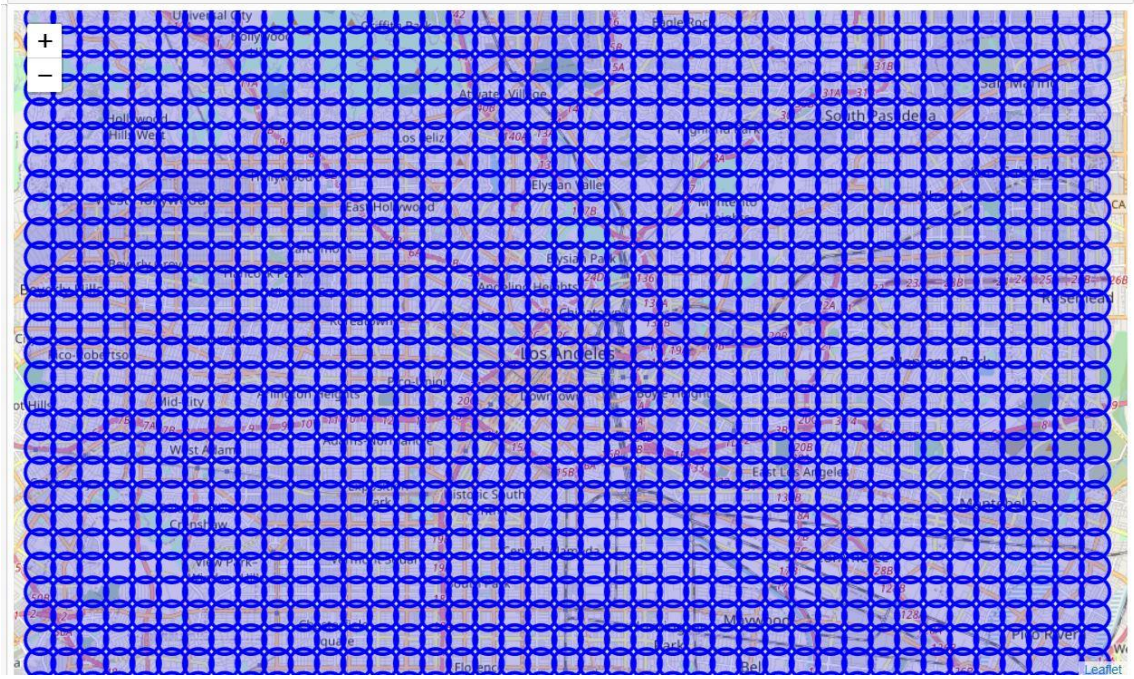
3.3 Padding Analysis

As mentioned before, there are only a small radius and venues for each request. We cannot get the venues of the whole city in one request. Thus, we need pad the area and send request for each pad and then combine them together.

First, we pad the cities by draw lots of that small circles that a request Foursquare accept. Remember that we should overlap somehow to best reduce the uncovered points.



```
map_la
```



Then, we send request to Foursquare and get the city venues. As we overlapped, there are much venues are duplicated in our dataset. We remove the duplicated by their unique ID Foursquare assigned.

```
venues_whole_la = getNearbyVenues_padCity('LA', lats_la, lngs_la)
```

```
print(venues_whole_la.shape)
venues_whole_la.drop_duplicates(subset=['Venue Id'], inplace=True)
print(venues_whole_la.shape)
venues_whole_la.head()
```

```
(22837, 8)
(15851, 8)
```

	City	Circle Latitude	Circle Longitude	Venue Id	Venue Name	Venue Latitude	Venue Longitude	Venue Category
0	LA	33.933691	-118.402767	4b5b63f8f964a5202cfa28e3	Flight Path Learning Center Museum	33.932137	-118.405278	Museum
1	LA	33.933691	-118.402767	4b73c6adf964a52087bb2de3	Embassy Suites by Hilton	33.930516	-118.400708	Hotel
2	LA	33.933691	-118.402767	516c6b9f498e44df7fd96038	Runway 7R - 25L	33.934993	-118.406074	Airport Service
3	LA	33.933691	-118.402767	4bafeeb5f964a520842c3ce3	1440 Bistro & Bar	33.930591	-118.400527	New American Restaurant
4	LA	33.933691	-118.402767	4c2273b29085d13afc5a86cc	Atlantic Aviation (LAX)	33.932380	-118.398805	Airport Terminal

```
venues_whole_ny = getNearbyVenues_padCity('NY', lats_ny, lngs_ny)
```

```
print(venues_whole_ny.shape)
venues_whole_ny.drop_duplicates(subset=['Venue Id'], inplace=True)
print(venues_whole_ny.shape)
venues_whole_ny.head()
```

```
(35604, 8)
(23709, 8)
```

	City	Circle Latitude	Circle Longitude	Venue Id	Venue Name	Venue Latitude	Venue Longitude	Venue Category
0	NY	40.592728	-74.166015	4e3807436284fcf739a14c12	Trader Joe's	40.589997	-74.165715	Grocery Store
1	NY	40.592728	-74.166015	4b316f69f964a520ef0625e3	City Wine Cellar	40.591564	-74.164036	Liquor Store
2	NY	40.592728	-74.166015	4bd423cf462cb713b7c5df07	Heartland Bagels - Richmond Ave	40.591464	-74.164186	Bagel Shop
3	NY	40.592728	-74.166015	4fdf3bfb0039f72fcfa41cb8	European Wax Center	40.591103	-74.164488	Health & Beauty Service
4	NY	40.592728	-74.166015	4b85d75ff964a520cb7531e3	Holy Schnitzel	40.589778	-74.164390	Restaurant

We find more then 15000 venues for Los Angeles and more than 23000 in New York. Then we can follow the same process to calculate the similarities.

```
paddedCity_similarity('New York, NY, US', 'Los Angeles, CA, US')
```

```
The similarity of city New York, NY, US and city Los Angeles, CA, US is 0.6629298646638225.
```

City	Los Angeles, CA, US	New York, NY, US
City		
Los Angeles, CA, US	1.00000	0.66293
New York, NY, US	0.66293	1.00000

This time we got similarities much higher than before as we expected.

4 Results: Global Cities

The model we used seems good on New York and Los Angeles. Let's extend to global cities. Firstly, we need the city list which can be accessed by Globalization and World Cities Research Network. And repeat the model feeding process to get our results.

We also want to group them by countries to check cities in same culture. We need outer join the two DataFrames by the 55 cities. If we look details about these cities, we can find that most famous global cities are in developed countries from Europe or North American. Few of them from Eastern Asian and South African, which have different cultures compared to Western world.

```
print(global_cities.shape)
global_cities.head()
```

```
(55, 1)
```

	CITIES
0	Amsterdam
1	Atlanta
2	Bangkok
3	Barcelona
4	Beijing

```
print(df_country.shape)
df_country.head()
```

```
(433, 2)
```

	CITY	Country
0	Aabenraa	Denmark
1	Aarhus	Denmark
2	Aberdeen	United Kingdom
3	Abu Dhabi	United Arab Emirates
4	Adelaide	Australia

We still send requests to Foursquare and receive the venues form global cities.

```
print('DataFrame contains {len(venues_global.City.unique()))
venues_global.head()
```

```
55
```

	City	Circle Latitude	Circle Longitude	Venue Id	Venue Name	Venue Latitude	Venue Longitude	Venue Category
0	Amsterdam	52.36854	4.889976	4a856313f964a52035fe1fe3	The American Book Center	52.368911	4.889425	Bookstore
1	Amsterdam	52.36854	4.889976	52af231b11d2d07f6e9bff9f	Van Stapele Koekmakerij	52.368828	4.888481	Dessert Shop
2	Amsterdam	52.36854	4.889976	4a2bbd19f964a520db961fe3	Gartine	52.369157	4.891615	Breakfast Spot
3	Amsterdam	52.36854	4.889976	5b0fa7e212f0a9002cb223cb	Bhatti Pasal	52.368055	4.890838	Restaurant
4	Amsterdam	52.36854	4.889976	4b586efef964a520855728e3	Frens Haringhandel	52.367336	4.891121	Food Truck

Still we encoding and grouping the venues list by city and feed them to our model.

```
print(grouped_global.shape)
grouped_global.head()
```

```
(55, 590)
```

City	ATM	Abruzzo Restaurant	Acai House	Accessories Store	Adult Boutique	Advertising Agency	Afghan Restaurant	African Restaurant	Airport Lounge	Airport Service	Alsatian Restaurant	American Restaurant	Amphitheater
Amsterdam	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.001957	0.0	0.0	0.0	0.001957	0.0
Atlanta	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.039801	0.0
Bangkok	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0
Barcelona	0.0	0.0	0.0	0.004552	0.0	0.0	0.0	0.001517	0.0	0.0	0.0	0.001517	0.0
Beijing	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0

Currently the city is sorted by name and we resort by country.

Sort Cities by their country

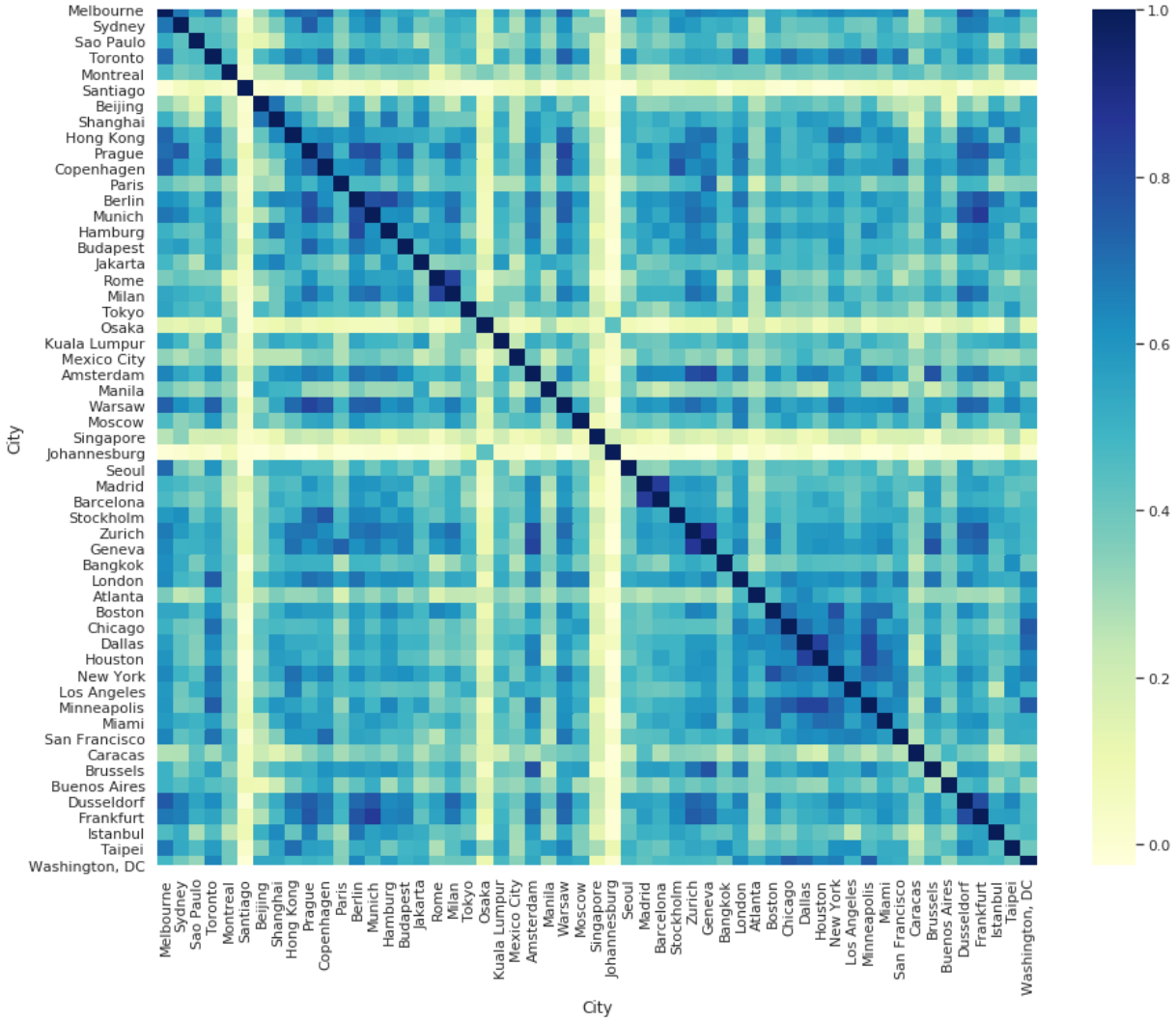
```
grouped_global = grouped_global.join(df_country.set_index('CITY'), on='City')
grouped_global.sort_values(by='Country ', inplace=True)
grouped_global.drop(['Country '], axis=1, inplace=True)
grouped_global.head()
```

City	ATM	Abruzzo Restaurant	Acai House	Accessories Store	Adult Boutique	Advertising Agency	Afghan Restaurant	African Restaurant
Melbourne	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.001912
Sydney	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.000000
Sao Paulo	0.0	0.0	0.002114	0.002114	0.0	0.0	0.0	0.000000
Toronto	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.000000
Montreal	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.000000

Then we feed the encoded dataset to our model and out put the heat plot for global city similarities.

Plot a Heat-map to see their relations

```
import seaborn as sns
sns.set(rc={'figure.figsize':(15,12)})
ax = sns.heatmap(Matrix_global, cmap="YlGnBu")
```



5 Discussion and Conclusions

From the final heat plot we can find several interesting facts:

1. There are some cities with very light color which means the city is much different than any other cities. If we look these cities in detail we can find that they are in different cultures, like East Asian including Japan and China, compared with the rest cities from western world like Europe and North American.
2. Most deep color occurred besides the diagnose line. Especially we can see some large square covering the diagnose line. Because we grouped cities from same country together, they must have similar culture and suffering from similar venue categories.
3. Some international cities crowed people from all over the world and can be recognized multi-culture city like New York and Paris.