Exploring and defining the best path

Nicolas SIMON

August 7th 2019

1. Introduction

1.1 Background :

Travelling and visiting often requires a significant amount of time for organization. Planning the right path is a long and manual process, where you spend most of your times googling in order to pick the best venues, spots, bars, etc.…

Take this example for instance : consider you're a manager and you have to organize a music tour for a music band. Your objectives are multiple :

- Which concert venues will provide the most visibility to the band and are most likely to be crowded during shows? The aim here is to of course, increase the band's popularity but also make higher profit by selling more show tickets.
- In what order should the shows be played? Minimizing travel distances is important to reduce costs of travelling, but also plays a role on the band's mood : the more time you spend in your travel van driving, the more your mood/moral is affected.

This example could also be employed for, let's say, simple tourists visiting a city or even a country: what are the trending venues I should absolutely see and in which order? It's complicated to organize a trip when you nothing about the place you'll be visiting.

1.2 Interest :

Obviously, travel agencies or even individuals would be interested in accurate path proposition, for competitive advantage and business values. These paths can be highly personalized depending on the person's interest and envies.

1.3 Case of Study :

In this report, we're going to focus on a specific case study which I already described above : the Music Manager case. The band he managed has gained the opportunity to play on a USA Tour, and are allowed one show in each state of the United States. Unfortunately, the manager's contact book is rather empty and he has no idea which venue to book for the shows, and in what order the shows should be played. Let's give him a little help.

2. Data Acquisition and Cleaning

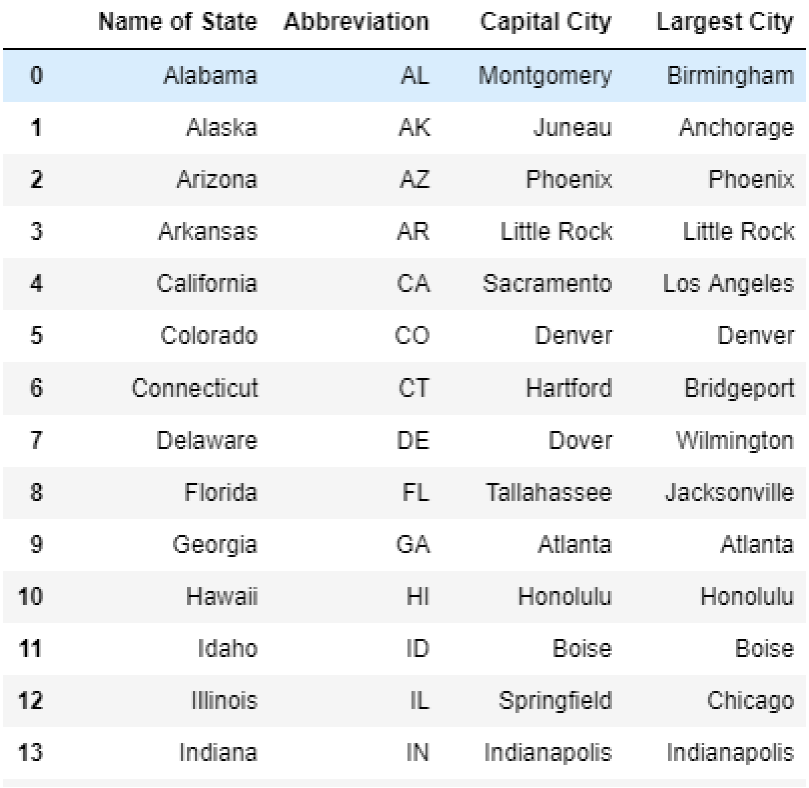
2.1 Data Sources

Most of the data will be acquired via Foursquare queries, which will help us explore specific venues and gather their location, address and overall ratings.

In our study case, we'll need the name and location of every state of the USA, the tabloid of state names is gathered via a Wikipedia page and will help focus on specific cities.

2.2 Data Cleaning

The first "data set" we acquire is a list of the states in the US, and requires no cleaning at all.



	Name of State	Abbreviation	Capital City	Largest City
0	Alabama	AL	Montgomery	Birmingham
1	Alaska	AK	Juneau	Anchorage
2	Arizona	AZ	Phoenix	Phoenix
3	Arkansas	AR	Little Rock	Little Rock
4	California	CA	Sacramento	Los Angeles
5	Colorado	CO	Denver	Denver
6	Connecticut	CT	Hartford	Bridgeport
7	Delaware	DE	Dover	Wilmington
8	Florida	FL	Tallahassee	Jacksonville
9	Georgia	GA	Atlanta	Atlanta
10	Hawaii	HI	Honolulu	Honolulu
11	Idaho	ID	Boise	Boise
12	Illinois	IL	Springfield	Chicago
13	Indiana	IN	Indianapolis	Indianapolis

Figure 1 : List of states in the US

The next data set we acquire will be through Foursquare queries and will be stocked in data frames. Because the data is imported via a JSON file, the data frame requires more cleaning in order to make it more readable and relevant. The information we gather about venues have to contain the following features:

Name, Category, Address, City, State, Longitude, Latitude, ID, Rating, Likes

The cleaning process will be described in the methodology section.

3. Methodology

3.1 Global Methodology

What we want in the end is a list of venue addresses, and then order this list by order of passage.

The process to get there is divided in 6 steps:

Step 1 : Get the names of the two main cities in each state of America : the capital city and the largest city. We'll have to choose afterwards which one we'll be playing at.

Step 2 : For each city, explore for music venues and gather them in a data frame after cleaning.

Step 3 : Get the ratings and like of each venue and order them by these features.

Step 4 : To simplify our model, we assume that the popularity of the venue is showed by the number of likes it has and its rating. So we pick the venue with the highest rating/number of likes and add its information to our final tour list.

Step 5 : Start again from step 2 with the next city and implement the result each time on our final tour list.

Step 6 : Order the final tour list by order of passage.

3.2 Model Simplifications:

Unfortunately, I had to over simplify my model, the reasons are mainly because the Foursquare didn't allow a huge number of queries, especially to explore given venues:

- Instead of exploring venues in the capital city and the largest city in order to compare them afterwards. We say that we choose automatically the capital city, so we minimize the search queries.

- We're allowed to do 500 foursquare queries, and we have approximately 50 cities to explore. We set a limit to 10 venues per city.

- We assume that the popularity of the venue is given by its rating and number of likes.

- Finding the best path through each city is actually a really known mathematical problem called the **Travelling Salesman Problem**. The Algorithm for finding the best path is an O(n!) problem and, for now, not enough optimized to run through our 50 nodes (cities). 50 nodes to explore means that there are actually 50! possible paths to take. In order to make the algorithm runnable, we're going to restrict our USA Tour to only 15 cities, just to give a concrete example.

- The distance between two cities is calculated with a mathematical equation and corresponds to the length of a strict line between two coordinates. We could build a more precise model by having access to Google Map's API which would give us the precise distance between two locations using real roads and highways.

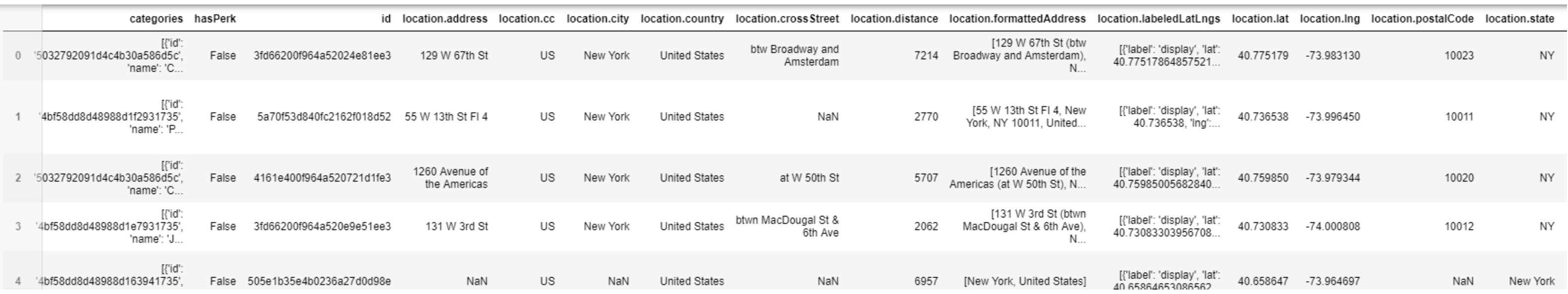
4. Results:

4.1 With one city : New York City

Before trying to run our algorithm through each 20 cities, let’s first try if it works a single city.

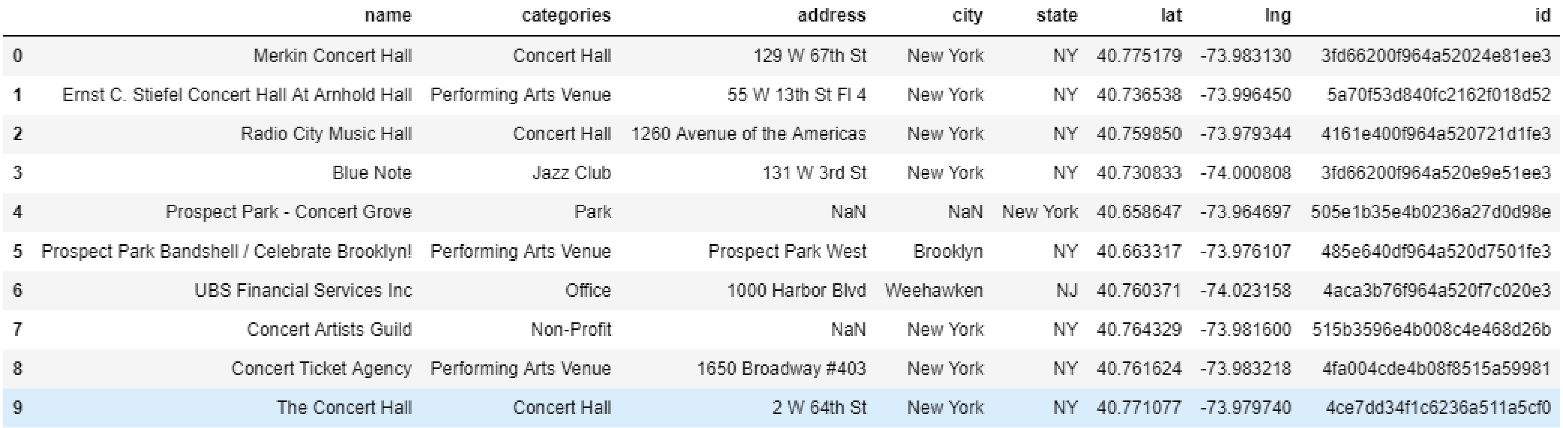
The global function which picks the best venue is divided in 2 main function which are:

- The cleaning algorithm : which get X venues (X is the limit you set) of a specific locations and puts them in a dataframe. The dataframe is then cleaned to make it more readable.



	categories	hasPerk	id	location.address	location.cc	location.city	location.country	location.crossStreet	location.distance	location.formattedAddress	location.labeledLatLngs	location.lat	location.lng	location.postalCode	location.state
0	[{'id': '5032792091d4c4b30a586d5c', 'name': 'C...	False	3fd66200f964a52024e81ee3	129 W 67th St	US	New York	United States	btw Broadway and Amsterdam	7214	[129 W 67th St (btw Broadway and Amsterdam), N...	[{'label': 'display', 'lat': 40.77517864857521...	40.775179	-73.983130	10023	NY
1	[{'id': '4bf58dd8d48988d1f2931735', 'name': 'P...	False	5a70f53d840fc2162f018d52	55 W 13th St Fl 4	US	New York	United States	NaN	2770	[55 W 13th St Fl 4, New York, NY 10011, United...	[{'label': 'display', 'lat': 40.736538, 'lng':...	40.736538	-73.996450	10011	NY
2	[{'id': '5032792091d4c4b30a586d5c', 'name': 'C...	False	4161e400f964a520721d1fe3	1260 Avenue of the Americas	US	New York	United States	at W 50th St	5707	[1260 Avenue of the Americas (at W 50th St), N...	[{'label': 'display', 'lat': 40.75985005682840...	40.759850	-73.979344	10020	NY
3	[{'id': '4bf58dd8d48988d1e7931735', 'name': 'J...	False	3fd66200f964a520e9e51ee3	131 W 3rd St	US	New York	United States	btwn MacDougal St & 6th Ave	2062	[131 W 3rd St (btwn MacDougal St & 6th Ave), N...	[{'label': 'display', 'lat': 40.73083303956708...	40.730833	-74.000808	10012	NY
4	[{'id': '4bf58dd8d48988d163941735',	False	505e1b35e4b0236a27d0d98e	NaN	US	NaN	United States	NaN	6957	[New York, United States]	[{'label': 'display', 'lat': 40.65864653086562	40.658647	-73.964697	NaN	New York

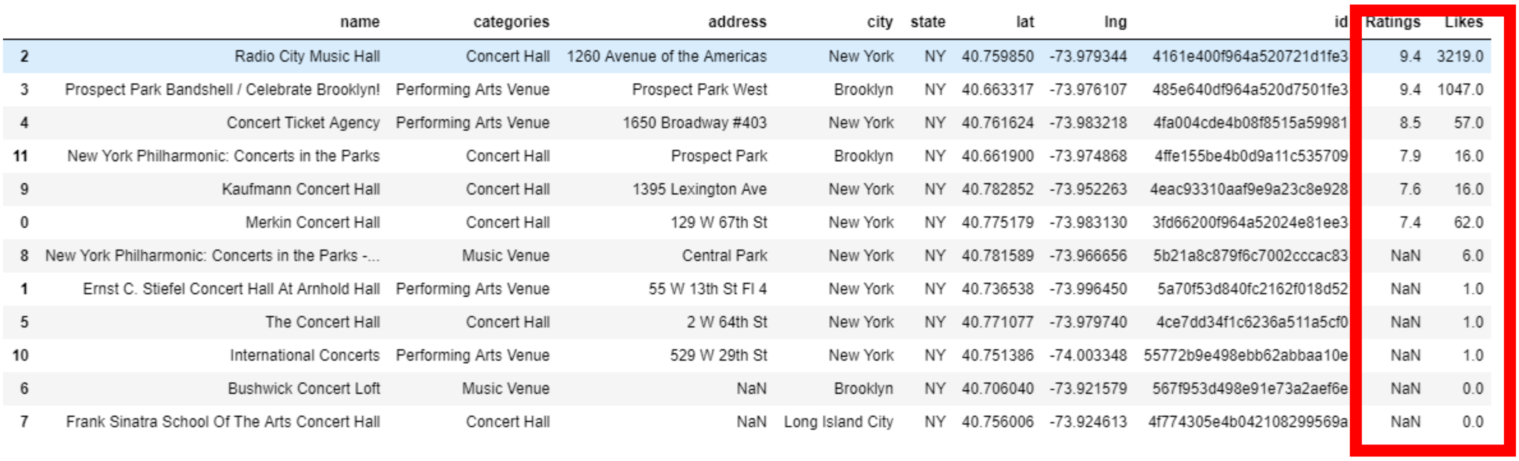
Figure 1 : Before cleaning



	name	categories	address	city	state	lat	lng	id
0	Merkin Concert Hall	Concert Hall	129 W 67th St	New York	NY	40.775179	-73.983130	3fd66200f964a52024e81ee3
1	Ernst C. Stiefel Concert Hall At Arnhold Hall	Performing Arts Venue	55 W 13th St Fl 4	New York	NY	40.736538	-73.996450	5a70f53d840fc2162f018d52
2	Radio City Music Hall	Concert Hall	1260 Avenue of the Americas	New York	NY	40.759850	-73.979344	4161e400f964a520721d1fe3
3	Blue Note	Jazz Club	131 W 3rd St	New York	NY	40.730833	-74.000808	3fd66200f964a520e9e51ee3
4	Prospect Park - Concert Grove	Park	NaN	NaN	New York	40.658647	-73.964697	505e1b35e4b0236a27d0d98e
5	Prospect Park Bandshell / Celebrate Brooklyn!	Performing Arts Venue	Prospect Park West	Brooklyn	NY	40.663317	-73.976107	485e640df964a520d7501fe3
6	UBS Financial Services Inc	Office	1000 Harbor Blvd	Weehawken	NJ	40.760371	-74.023158	4aca3b76f964a520f7c020e3
7	Concert Artists Guild	Non-Profit	NaN	New York	NY	40.764329	-73.981600	515b3596e4b008c4e468d26b
8	Concert Ticket Agency	Performing Arts Venue	1650 Broadway #403	New York	NY	40.761624	-73.983218	4fa004cde4b08f8515a59981
9	The Concert Hall	Concert Hall	2 W 64th St	New York	NY	40.771077	-73.979740	4ce7dd34f1c6236a511a5cf0

Figure 2 : After Cleaning

- The rating algorithm : which gets the ratings/likes of all the venues of the dataframe and then orders the dataframe by it’s popularity.

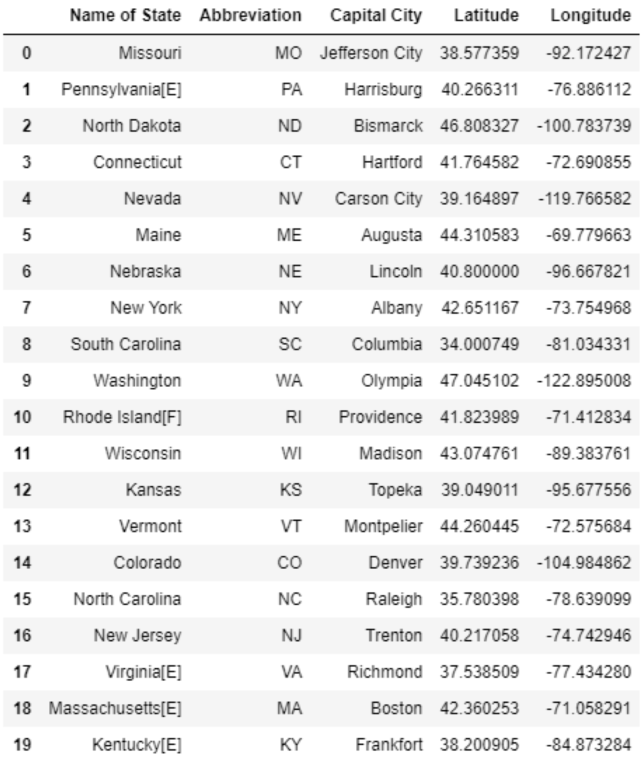


	name	categories	address	city	state	lat	lng	id	Ratings	Likes
2	Radio City Music Hall	Concert Hall	1260 Avenue of the Americas	New York	NY	40.759850	-73.979344	4161e400f964a520721d1fe3	9.4	3219.0
3	Prospect Park Bandshell / Celebrate Brooklyn!	Performing Arts Venue	Prospect Park West	Brooklyn	NY	40.663317	-73.976107	485e640df964a520d7501fe3	9.4	1047.0
4	Concert Ticket Agency	Performing Arts Venue	1650 Broadway #403	New York	NY	40.761624	-73.983218	4fa004cde4b08f8515a59981	8.5	57.0
11	New York Philharmonic: Concerts in the Parks	Concert Hall	Prospect Park	Brooklyn	NY	40.661900	-73.974868	4ffe155be4b0d9a11c535709	7.9	16.0
9	Kaufmann Concert Hall	Concert Hall	1395 Lexington Ave	New York	NY	40.782852	-73.952263	4eac93310aaf9e9a23c8e928	7.6	16.0
0	Merkin Concert Hall	Concert Hall	129 W 67th St	New York	NY	40.775179	-73.983130	3fd66200f964a52024e81ee3	7.4	62.0
8	New York Philharmonic: Concerts in the Parks -...	Music Venue	Central Park	New York	NY	40.781589	-73.966656	5b21a8c879f6c7002cccac83	NaN	6.0
1	Ernst C. Stiefel Concert Hall At Arnhold Hall	Performing Arts Venue	55 W 13th St Fl 4	New York	NY	40.736538	-73.996450	5a70f53d840fc2162f018d52	NaN	1.0
5	The Concert Hall	Concert Hall	2 W 64th St	New York	NY	40.771077	-73.979740	4ce7dd34f1c6236a511a5cf0	NaN	1.0
10	International Concerts	Performing Arts Venue	529 W 29th St	New York	NY	40.751386	-74.003348	55772b9e498ebb62abbaa10e	NaN	1.0
6	Bushwick Concert Loft	Music Venue	NaN	Brooklyn	NY	40.706040	-73.921579	567f953d498e91e73a2aef6e	NaN	0.0
7	Frank Sinatra School Of The Arts Concert Hall	Concert Hall	NaN	Long Island City	NY	40.756006	-73.924613	4f774305e4b042108299569a	NaN	0.0

- After, the best venue is picked. In this case the ‘Radio City Music Hall’

4.2 With all cities : USA tour

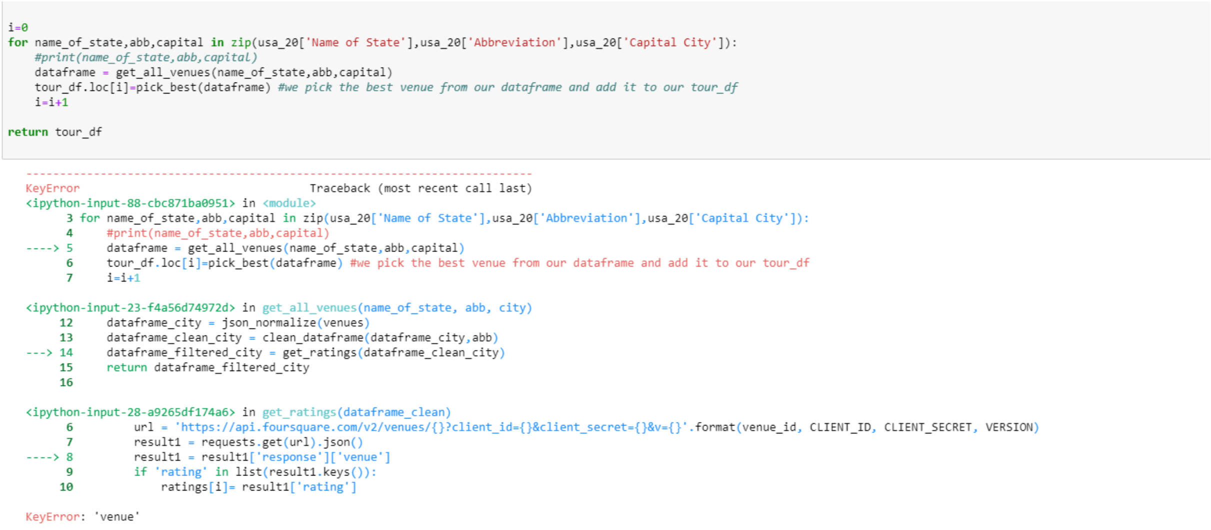
Now that we know that it works with a single city, we’ll have to iterate the process through a choosen number of cities. I personally picked 20 random capital cities from different states in the USA.



	Name of State	Abbreviation	Capital City	Latitude	Longitude
0	Missouri	MO	Jefferson City	38.577359	-92.172427
1	Pennsylvania[E]	PA	Harrisburg	40.266311	-76.886112
2	North Dakota	ND	Bismarck	46.808327	-100.783739
3	Connecticut	CT	Hartford	41.764582	-72.690855
4	Nevada	NV	Carson City	39.164897	-119.766582
5	Maine	ME	Augusta	44.310583	-69.779663
6	Nebraska	NE	Lincoln	40.800000	-96.667821
7	New York	NY	Albany	42.651167	-73.754968
8	South Carolina	SC	Columbia	34.000749	-81.034331
9	Washington	WA	Olympia	47.045102	-122.895008
10	Rhode Island[F]	RI	Providence	41.823989	-71.412834
11	Wisconsin	WI	Madison	43.074761	-89.383761
12	Kansas	KS	Topeka	39.049011	-95.677556
13	Vermont	VT	Montpelier	44.260445	-72.575684
14	Colorado	CO	Denver	39.739236	-104.984862
15	North Carolina	NC	Raleigh	35.780398	-78.639099
16	New Jersey	NJ	Trenton	40.217058	-74.742946
17	Virginia[E]	VA	Richmond	37.538509	-77.434280
18	Massachusetts[E]	MA	Boston	42.360253	-71.058291
19	Kentucky[E]	KY	Frankfort	38.200905	-84.873284

Figure 3 : 20 random capital cities and their coordinates.

Unfortunately, I exceed the number of queries each time I want to run my algorithm. Which makes it impossible for me to reach the 20 best venues. I always get the error :

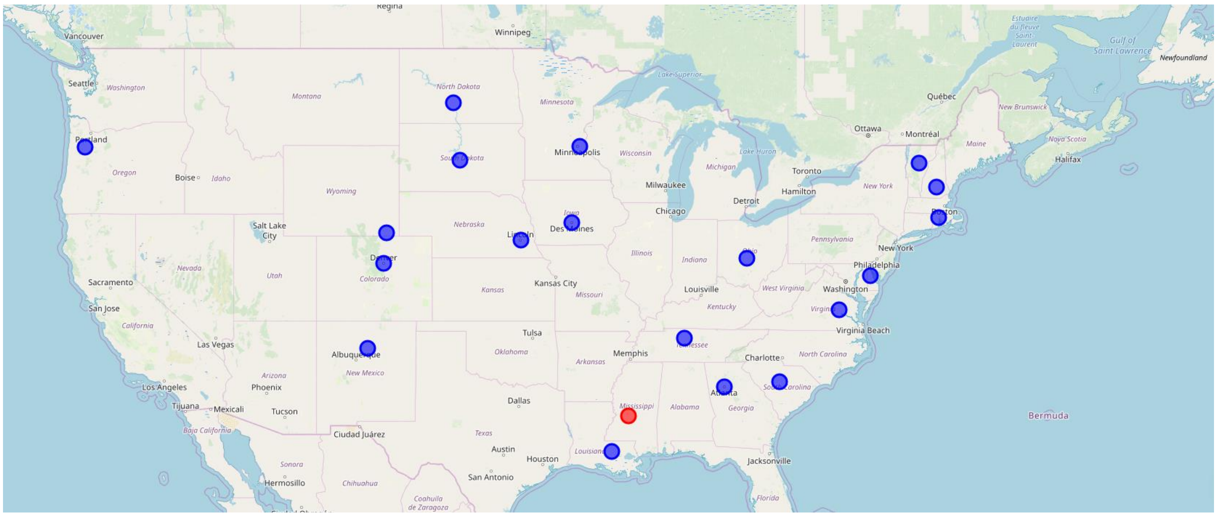


```
i=0
for name_of_state,abb,capital in zip(usa_20['Name of State'],usa_20['Abbreviation'],usa_20['Capital City']):
    #print(name_of_state,abb,capital)
    dataframe = get_all_venues(name_of_state,abb,capital)
    tour_df.loc[i]=pick_best(dataframe) #we pick the best venue from our dataframe and add it to our tour_df
    i=i+1

return tour_df
```

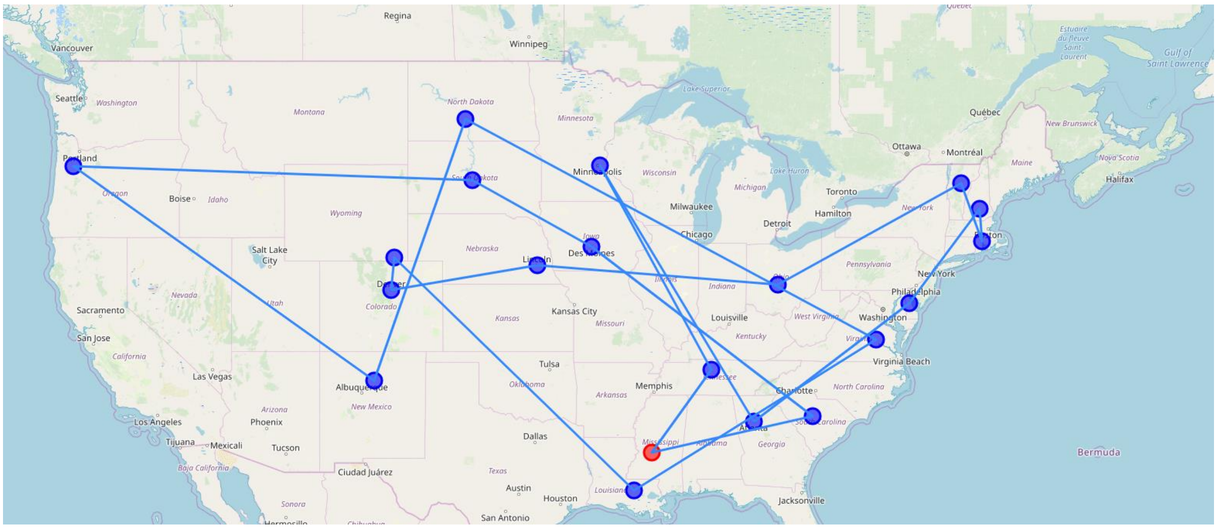
```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-88-cbc871ba0951> in <module>
      3 for name_of_state,abb,capital in zip(usa_20['Name of State'],usa_20['Abbreviation'],usa_20['Capital City']):
      4     #print(name_of_state,abb,capital)
----> 5     dataframe = get_all_venues(name_of_state,abb,capital)
      6     tour_df.loc[i]=pick_best(dataframe) #we pick the best venue from our dataframe and add it to our tour_df
      7     i=i+1

<ipython-input-23-f4a56d74972d> in get_all_venues(name_of_state, abb, city)
     12     dataframe_city = json_normalize(venues)
     13     dataframe_clean_city = clean_dataframe(dataframe_city,abb)
---> 14     dataframe_filtered_city = get_ratings(dataframe_clean_city)
     15     return dataframe_filtered_city
     16

<ipython-input-28-a9265df174a6> in get_ratings(dataframe_clean)
      6         url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(venue_id, CLIENT_ID, CLIENT_SECRET, VERSION)
      7         result1 = requests.get(url).json()
----> 8         result1 = result1['response']['venue']
      9         if 'rating' in list(result1.keys()):
     10             ratings[i]= result1['rating']

KeyError: 'venue'
```

Figure 4 : Error due to exceeded queries

Ultimately, I would have gotten a list of 20 venues to play at. Then I would have displayed them on a folium map. I tried again with less queries (by setting a lower Limit) and finally got all my venues on a map.



Now that we have our venues, we’d have to find the best path possible to minimize travel distances. This problem, as said before, is a really famous one : The travelling Salesman Problem. Using brute force to find the best path can be very computationally expensive (0(n!)), hence we choose to only get 20 venues and not 50 like we wanted. I used the library MLRose to find the best path, but it doesn’t let you choose the starting point. I displayed the path on the same folium map, the starting point is the one in red color :



5. Dicussion:

By looking at the displayed path, looks like the algorithm isn’t very optimal after all. MLRose tends to become less efficient the more nodes you have. If we’d try again but with 10 nodes this time, it’d find a better and optimal way.

To get a general feedback about this work, here’s my point of view :

First of all, many simplifications were made. Unfortunately, Foursquare API doesn’t let us do a sufficient number of queries to find and explore more venues. I had to set a limit of 10 venues per city, not to mention that sometimes these venues can sometimes be irrelevant and are dropped after cleaning. In addition, the fact that Foursquare API explore on a given “radius” isn’t very precise when it comes to getting venues of a specific city. The city isn’t defined by it’s radius so it makes it difficult to find the “real” limit of the given city.

Also the “distance” between each node is very approximative and could have been improved a lot by using google maps which calculates a correct travel distance by passing through highways and roads.

6. Conclusion:

In conclusion, I’d say that this work was only a displaying example of how we could use tools like Foursquare API to build highly personalized “road-trips” for individuals or even for travel agencies.

Scaling down to a city, we could apply the same work to build our own personalized “guided-tour” where we explore the “must-visit” venues and by finding the best path through the city. On top of that we could even combine it with the use of public transports, telling which bus/subway is best to take to minimize time and travel distances.

During this project, the only drawback is that I’ve been limited by the tools. Having a premium Foursquare/Google Maps API access would have been great to improve my model and go deeper into subtilities. I barley scratched the surface of the concept, and this project only showed a glimpse of what I could have done with more means and time.