1. **Introduction/Business Problem**

How many times have you been in this situation? It's a Friday night. Your friends and you want to take on the city, maybe try a new spot. But there's risk associated with that, what if the venue isn't "fire." It's lame. Your friends want to dip, but where to now? You're in a new part of the city. No one's particularly familiar with it. Do you want to subject yourself to that kind of stress when you could otherwise be having a great night on the town?

The solution to this lies in leveraging location data from Foursquare's location API. By analyzing venue information from locations throughout a city, one can group venues into "walkable" areas and use information from venues within those areas to group similar areas together.

This description of the data is useful to people who want to find an area with a density of venues that they can check out throughout a night. This would-be application could be a one-stop spot to determine your next day/night out.

2. **Data**

The project will use the Foursquare API's "explore" endpoint to query for venues in a particular geographic area. Then, I will collect more information on each venue using the "venues" endpoint. (Note the initial plan was to use the "trending" endpoint to get additional information as that info is more useful from a Data Science perspective, but it seems that support for that endpoint is deprecated or not maintained properly)

The data that will be leveraged are:

- Name of venue
- Location (latitude, longitude, and address)
- Category/Type of Venue
- Stats (Rating and Number of Likes)
- Hours
- Website
- If the trending endpoint had been supported:
    - ~~Popular (hours of week when it is the most popular)~~
    - ~~Attributes (such as price tier)~~

The popular hours and price tier information would have been interesting to analyze, but unfortunately all the analysis will have to be done on using the categories information.

The locations of the venues will be markers on a Folium map. The groups of "walkable" areas will be marked using circles of radius .5 miles (0.8km walkable by most American standards). Folium popups will be used to display the above information. Therefore, a user can tap on a venue and see all the information they would want to know about it.

3. Methodology

Most of the interesting programming parts of this project can be found in the Python notebook also available in this GitHub. In this section, I will include screenshots for the maps that are generated that cannot be displayed on Github.

3a)

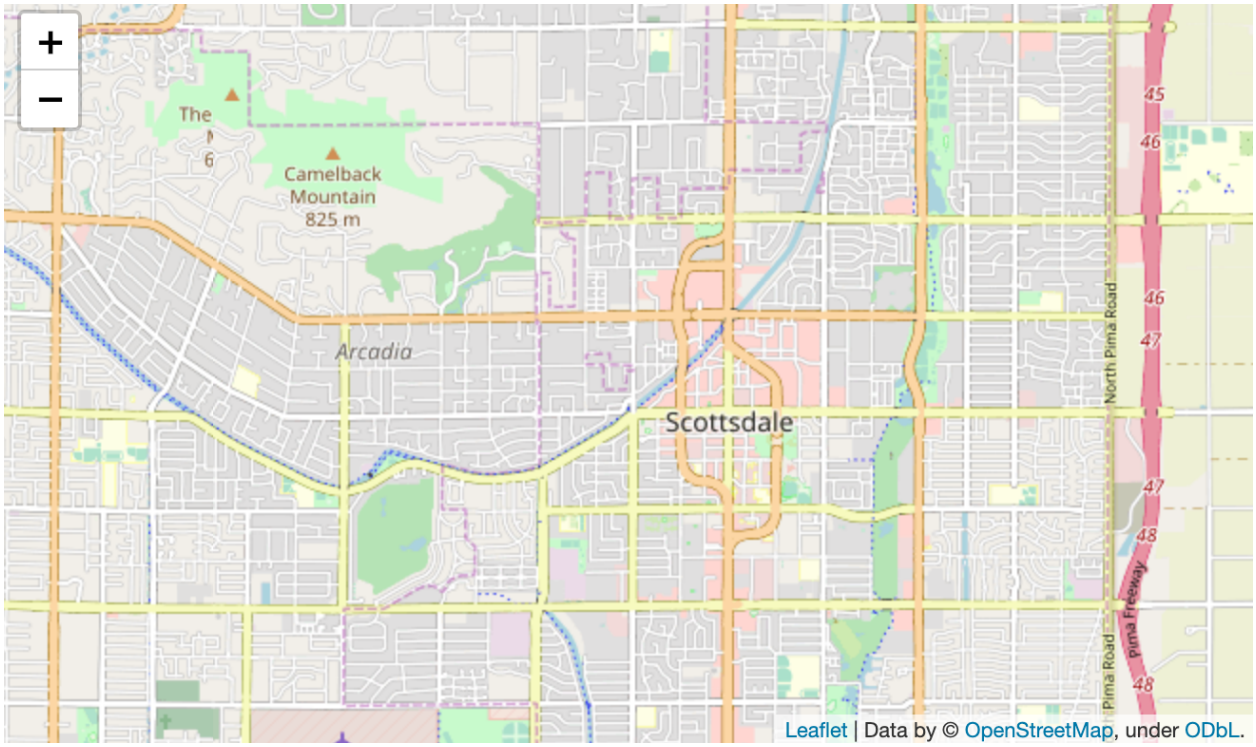


Fig 1: Empty map of example city

3b)

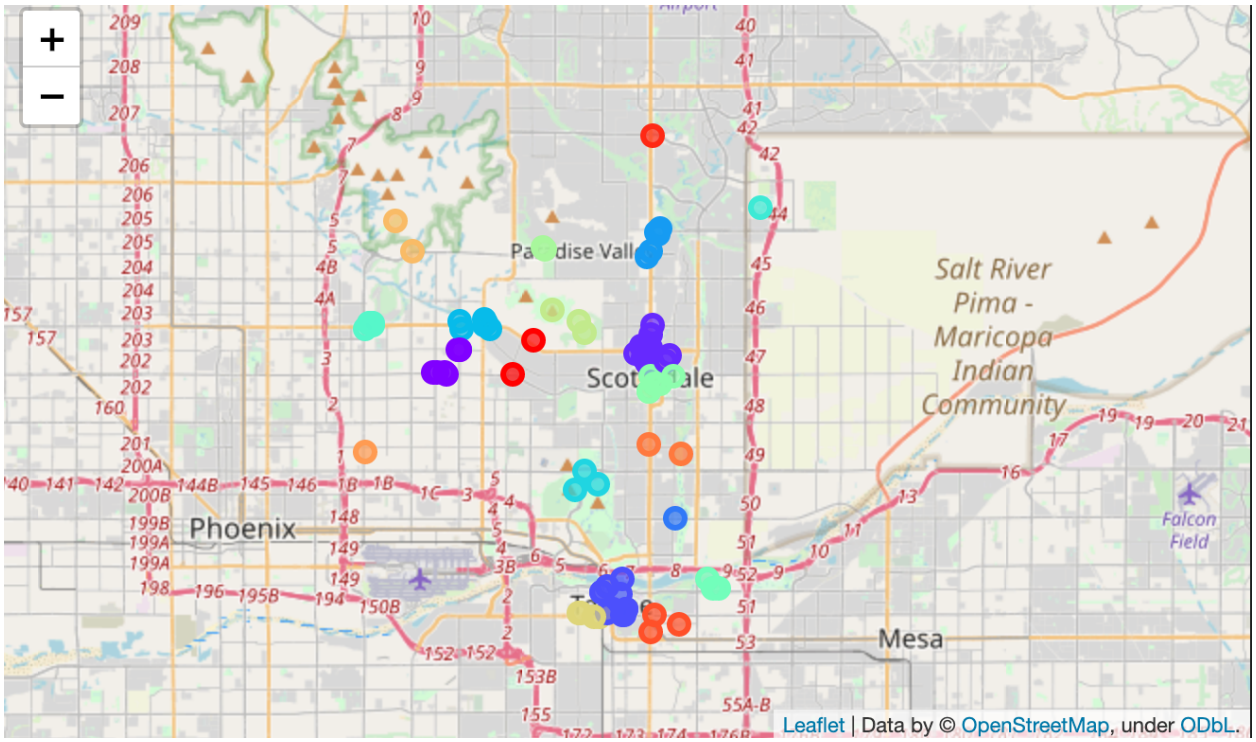


Fig 2: Map of all venues within ~10 miles (16km) of city center colored by the "walkable" area they area apart of

3c)

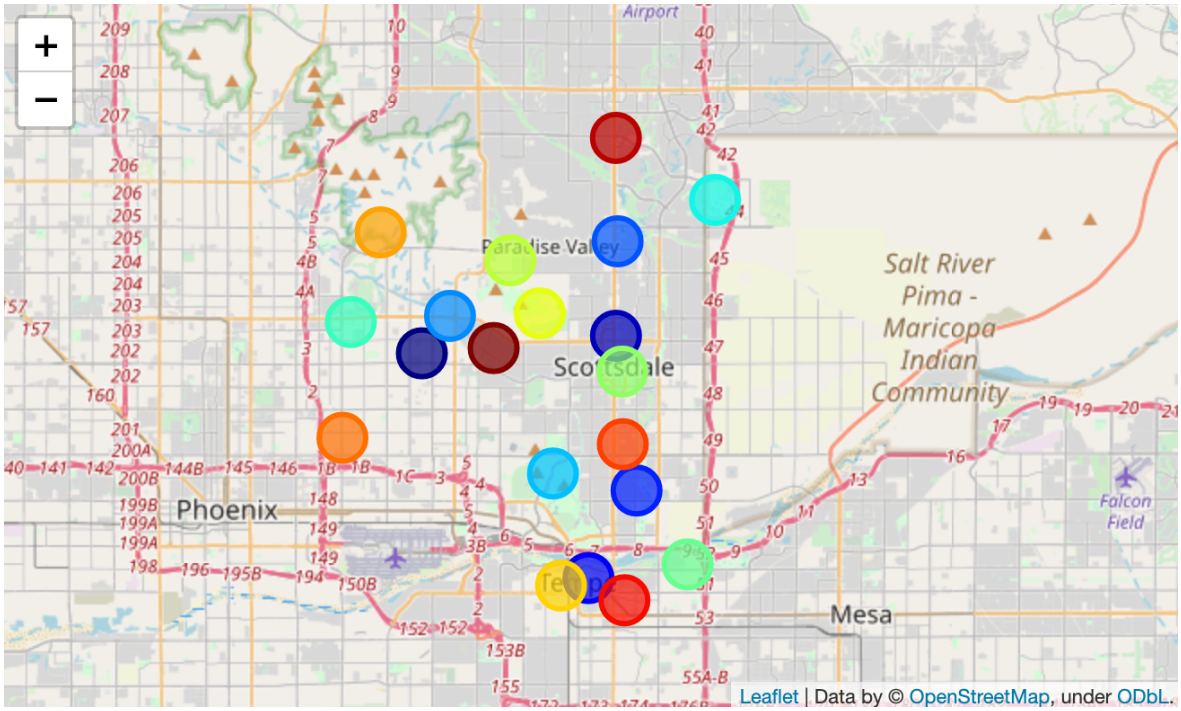


Fig 3: The "walkable" areas displayed on the map

3d)

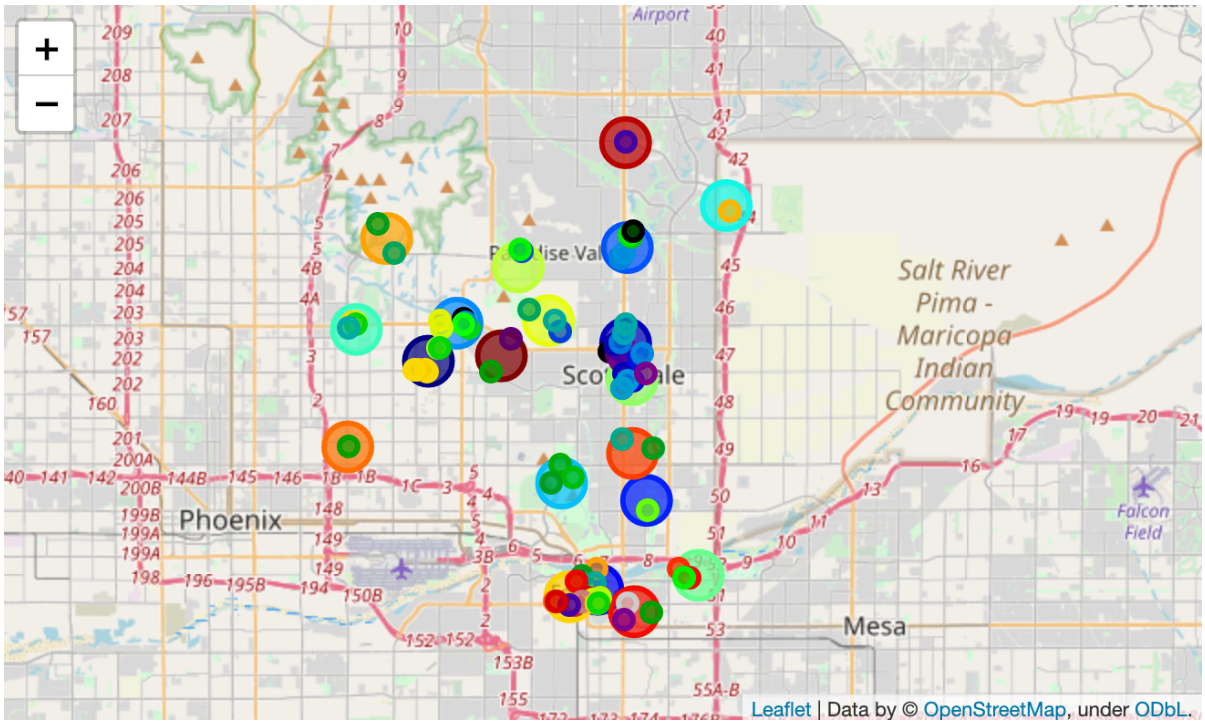


Fig 4: The venue locations inside their "walkable" areas. The color of the "walkable" areas is the result of the clustering algorithm grouping them by similarity of top venues

Machine Learning

The clusters for both the "walkable" areas and the coloring of those areas were done using a K-means clustering algorithm explained more thoroughly in notebook.

The first clustering is used to determine the geographic areas is set up such that the number of clusters divides the original city query area into 0.5 miles sections. The second clustering algorithm determines clusters based on the similarity of the most common venues within each cluster. As stated previously, this information is displayed through the color of the .5 mile area that cluster defines.

```
num_top_venues = 5

for cluster in venues_grouped['Cluster Number']:
    print("----" + str(cluster) + "----")
    temp = venues_grouped[venues_grouped['Cluster Number'] == cluster].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

```
----1----
                venue  freq
0         Burger Joint  0.29
1  American Restaurant  0.14
2            Wine Bar  0.14
3             Tiki Bar  0.14
4         Pizza Place  0.14


----2----
               venue  freq
0          Multiplex  0.08
1              Plaza  0.08
2         Restaurant  0.08
3              Hotel  0.08
4  Electronics Store  0.08


----3----
                    venue  freq
0      American Restaurant  0.08
1                 Mountain  0.08
2                      Bar  0.08
3                     Lake  0.08
4  New American Restaurant  0.08
```

Picture 1: Display a category and frequency of venue for each .5 mile group

	Cluster	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	1	Burger Joint	American Restaurant	Wine Bar	Pizza Place	Donut Shop	Tiki Bar	Fried Chicken Joint	Bar	Ice Cream Shop	Hotel
1	2	Coffee Shop	Restaurant	Mediterranean Restaurant	Electronics Store	Multiplex	Cupcake Shop	Pizza Place	Hotel	Plaza	Grocery Store
2	3	American Restaurant	Park	Bar	Lake	Breakfast Spot	Wine Bar	Burger Joint	Sandwich Place	Mountain	Movie Theater
3	4	Greek Restaurant	Wine Shop	Lake	Italian Restaurant	Ice Cream Shop	Hotel	Grocery Store	Gluten-free Restaurant	German Restaurant	Gastropub
4	5	Playground	Grocery Store	Mexican Restaurant	Steakhouse	Museum	Wine Shop	Hotel	Greek Restaurant	Gluten-free Restaurant	German Restaurant

Table 1: Cluster and associated top 10 venues

	Venue ID	Venue	Venue Latitude	Venue Longitude	Venue Category	Cluster Number	Popular Venue Cluster Label	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue
0	4b205619f964a520c63024e3	Mastro's City Hall Steakhouse	33.501572	-111.931429	Steakhouse	2	0	Coffee Shop	Restaurant	Mediterranean Restaurant	Electronics Store
2	4acfcc52f964a5201ed620e3	Scottsdale Waterfront	33.500062	-111.927895	Plaza	2	0	Coffee Shop	Restaurant	Mediterranean Restaurant	Electronics Store
3	47fe99e7f964a520e84e1fe3	Olive & Ivy Restaurant + Marketplace	33.500098	-111.928382	Mediterranean Restaurant	2	0	Coffee Shop	Restaurant	Mediterranean Restaurant	Electronics Store
4	4af46491f964a52006f221e3	Cartel Coffee Lab	33.498454	-111.927565	Coffee Shop	2	0	Coffee Shop	Restaurant	Mediterranean Restaurant	Electronics Store
5	58a2b17cb3cdc8794dcb937b	Apple Fashion Square	33.503550	-111.926432	Electronics Store	2	0	Coffee Shop	Restaurant	Mediterranean Restaurant	Electronics Store
...	...	...	...	...	...	...	...	...	...	...	...

Table 2: Merged dataframe with associated top venues near a particular venue

K-means is run on the top 10 venue data from the dataframe in Table 2 to determine clusters of similar venues.

4. Results

Much of the results are in the notebook. I will use this space to display examples of the final map output which can't be viewed on GitHub

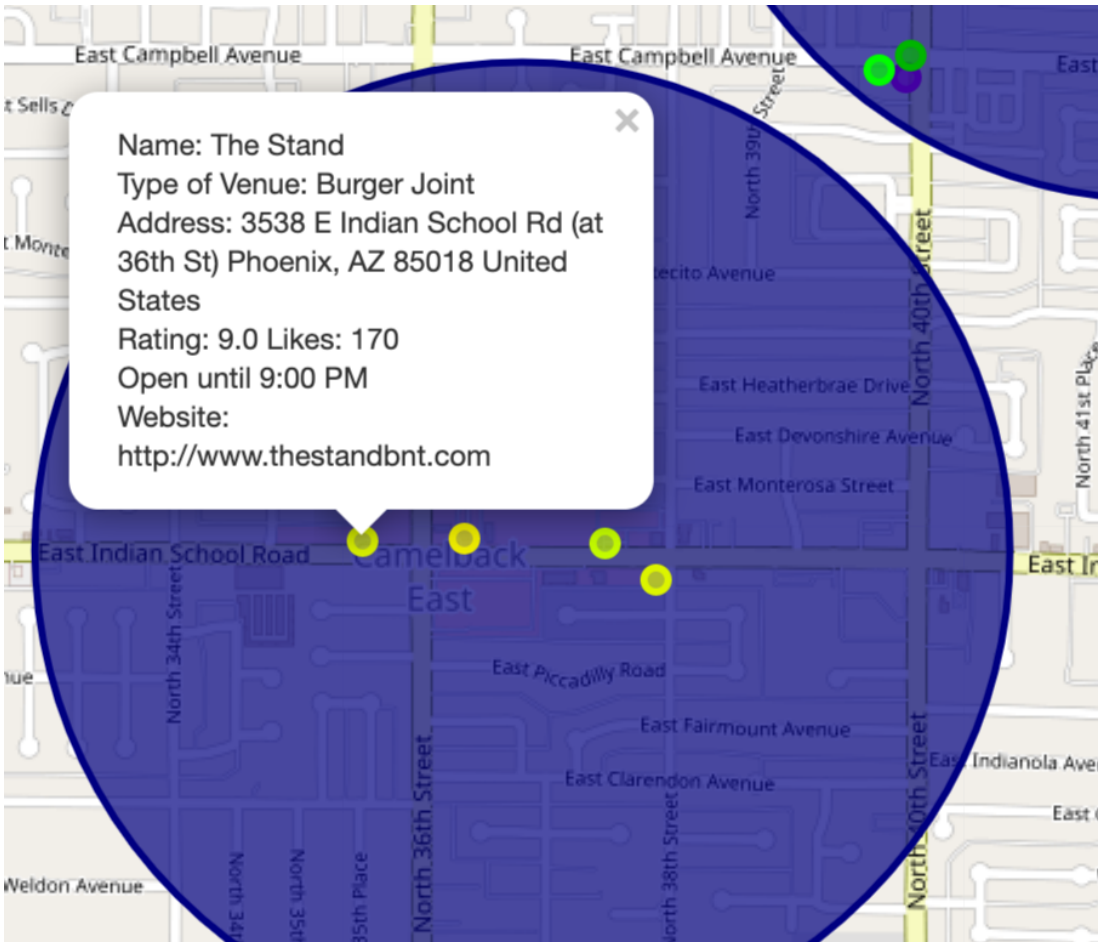


Fig 4: Example of popup of venue information

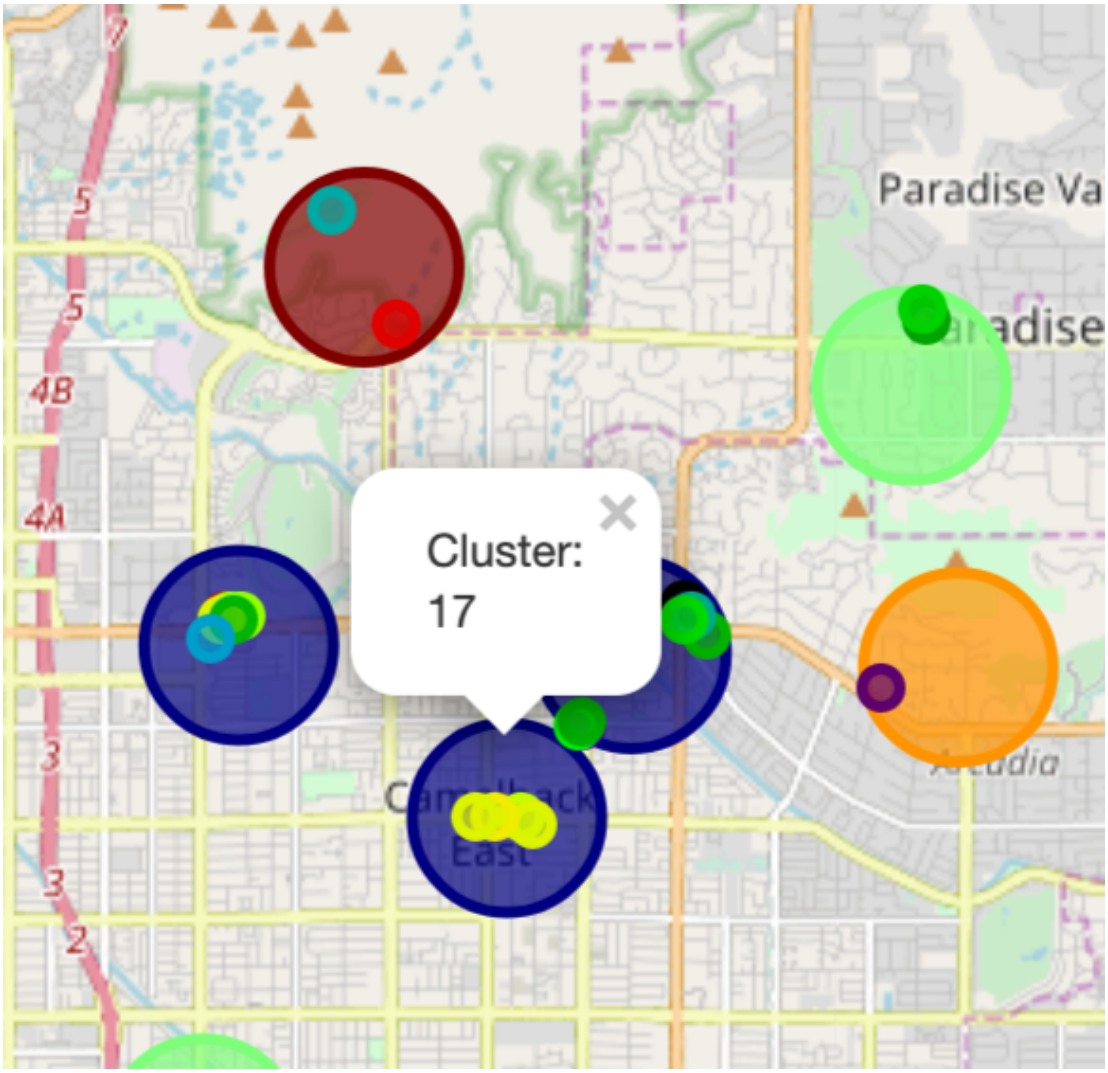


Fig 5: Example of venues that have similar top venues to each other (the three in navy blue have more similar top venues than those of the other colors surrounding them)

```
----1----
                     venue  freq
0      American Restaurant  0.22
1                Wine Bar  0.11
2       Italian Restaurant  0.11
3                Steakhouse  0.11
4  New American Restaurant  0.11


----2----
                     venue  freq
0      American Restaurant  0.09
1                 Mountain  0.09
2                      Bar  0.09
3  New American Restaurant  0.09
4                     Park  0.09


----3----
                  venue  freq
0                  Park   1.0
1  American Restaurant   0.0
2                  Bar   0.0
3             Mountain   0.0
4        Movie Theater   0.0
```

```
----4----
                venue  freq
0              Museum   0.2
1          Steakhouse   0.2
2       Grocery Store   0.2
3          Playground   0.2
4  Mexican Restaurant   0.2


----5----
                  venue  freq
0     Botanical Garden  0.33
1                 Park  0.33
2       Scenic Lookout  0.33
3  American Restaurant  0.00
4           Restaurant  0.00


----6----
                     venue  freq
0  New American Restaurant  0.33
1      American Restaurant  0.17
2           Ice Cream Shop  0.17
3             Burger Joint  0.17
4                    Hotel  0.17
```

```
----7----
                venue  freq
0           Wine Shop  0.33
1  Japanese Restaurant  0.33
2                 Spa  0.33
3              Resort  0.00
4        Movie Theater  0.00


----8----
                venue  freq
0              Resort   0.5
1                 Spa   0.5
2  American Restaurant   0.0
3             Mountain   0.0
4        Movie Theater   0.0


----9----
                     venue  freq
0  Mediterranean Restaurant   1.0
1                       Bar   0.0
2                  Mountain   0.0
3             Movie Theater   0.0
4                 Multiplex   0.0
```

```
----10----
                venue  freq
0     Greek Restaurant   1.0
1  American Restaurant   0.0
2   Mexican Restaurant   0.0
3        Movie Theater   0.0
4            Multiplex   0.0


----11----
                     venue  freq
0  Mediterranean Restaurant  0.09
1                 Multiplex  0.09
2                Steakhouse  0.09
3               Pizza Place  0.09
4               Coffee Shop  0.09


----12----
                venue  freq
0                Park   0.5
1        Deli / Bodega   0.5
2  American Restaurant   0.0
3          Restaurant   0.0
4        Movie Theater   0.0
```

```
----13----
                venue  freq
0             Brewery   1.0
1  American Restaurant   0.0
2          Restaurant   0.0
3       Movie Theater   0.0
4           Multiplex   0.0


----14----
                venue  freq
0     Baseball Stadium   1.0
1  American Restaurant   0.0
2          Restaurant   0.0
3       Movie Theater   0.0
4           Multiplex   0.0


----15----
                venue  freq
0               Trail   0.5
1                Park   0.5
2  American Restaurant   0.0
3              Resort   0.0
4            Mountain   0.0
```

```
----16----
                venue  freq
0         Burger Joint  0.50
1             Tiki Bar  0.25
2           Donut Shop  0.25
3  American Restaurant  0.00
4           Restaurant  0.00


----17----
                venue  freq
0         Coffee Shop  0.29
1  Mexican Restaurant  0.14
2   German Restaurant  0.14
3             Theater  0.14
4                Café  0.14


----18----
                venue  freq
0         Pizza Place   1.0
1  American Restaurant   0.0
2                 Bar   0.0
3            Mountain   0.0
4       Movie Theater   0.0
```

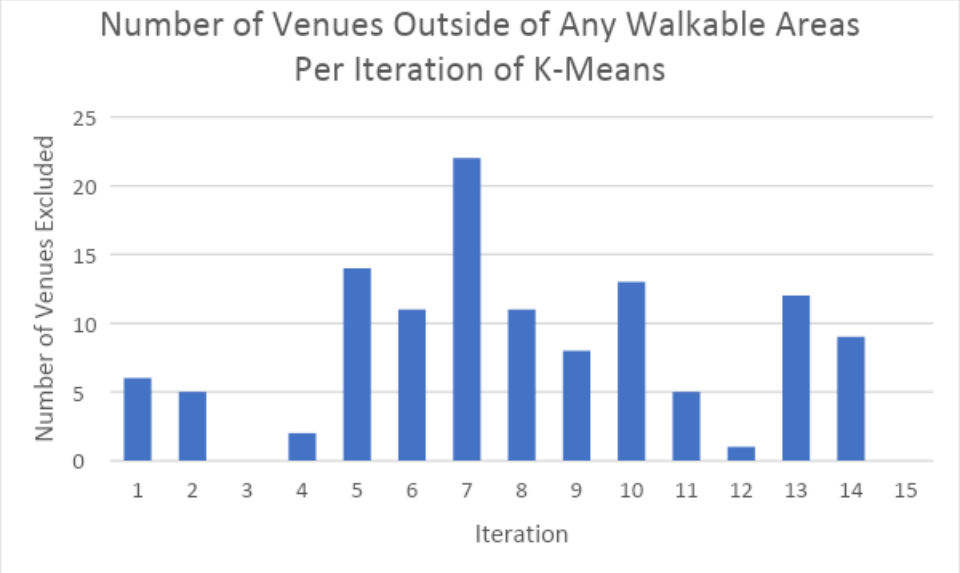
```
----19----
            venue  freq
0             Bar  0.25
1    Liquor Store  0.25
2     Pizza Place  0.25
3  Sandwich Place  0.25
4          Resort  0.00


----20----
                venue  freq
0             Brewery  0.33
1         Coffee Shop  0.33
2  Fried Chicken Joint  0.33
3  American Restaurant  0.00
4       Sandwich Place  0.00
```

Fig 6: Each cluster's top 5 venues and their frequency

5. Discussion

In an effort to make this system generalizable some issues arise. For one, K-means can produce different results with each iteration. In fact, through development I tracked how many venues got excluded from a walkable area.



Graph 1: Number of venues outside of any walkable Area per iteration of K-Means

On average the number of venues excluded is 7.9 with a standard deviation of 5.9, implying this is not the most consistent solution. The minimum excluded is 0 implying

sometimes it is perfect. The max was at iteration 7 with 22 excluded, implying sometimes the result can be quite bad.

The benefit of this system is that it can work with any city that has Foursquare data. The downside is that results may vary. Future work to lower the chances of this happening is to predefine (initialize) the centroids and distribute them around the venues perhaps after the first time of running the algorithm to avoid randomness.

Another downside to using K-means is that it doesn't always produce mutually exclusive areas. There might be overlap just due to the distribution of venues. K-means was chosen because of its autonomy, which would remove the need for geospatial data for every city. However, it still yields some unexpected results. Perhaps using DBSCAN in future iterations would produce better results as DBSCAN works better with densities of data.

The current implementation only allows for querying up to 100 venues as Foursquare places limits on how many times one can query for this price tier and as a college student it is hard for me to justify the cost to attain more queries.

It would be interesting to see the true distribution of venues across a city be getting information on all the venues of that city. This might also help address the issue shown in the notebook where we begin with 100 venues but reduce down to 78 after grouping based on "walkable distance." More venues might better distribute the area centers putting more venues within "walkable distance."

Finally, it would be useful to a user to see not just the cluster label but also an average or description of the venues within that cluster such as average rating of an area, latest hours of a venue in that area, etc. All these can be expanded on in further iterations of this project.

6. Conclusion

It's one thing to know what places there are to visit within your city. It's another thing to be able to pick an optimal location based on the venues within that area and discover new parts of a city that you may like. This implementation allows for that and gives a user visual cues on the information relevant to them planning their next visit to the city. Using K-means as a method for getting this information is an interesting choice as it is generalizable to different cities in a robust fashion, albeit with some limitations.