Battle Of The Neighborhoods

Coursera_capstone

Problem Statement

- “Suggest whether or not to move to a neighborhood based on its neighborhood profile. Suggest a
tourist which neighborhood to visit based on their interest. “
- Cities are, in part, entities varying from coffee shops to operas that not only provide to the needs of local citizens but also to tourists from around the world
- For bigger cities, the entities can be spread apart or concentrated based on its geographical proximity to a place of significance like a monument or a memorial, resulting in an ecosystem of neighborhoods that evolve and change over time.

Methodology: Data Sources

- NYU Spatial Data Repository
- Foursquare's 'Places API

rough	Neighborhood	Latitude	Longit
Bronx	Wakefield	40.894705	-73.847
Bronx	Co-op City	40.874294	-73.829
Bronx	Eastchester	40.887556	-73.827
Bronx	Fieldston	40.895437	-73.905
Bronx	Riverdale	40.890834	-73.912

ood Latitude	Neighborhood Longitude	
40.894705	-73.847201	T
40.894705	-73.847201	24
40.894705	-73.847201	
40.894705	-73.847201	
40.894705	-73.847201	

Data Analysis

- Most entries pulled from the API request included a 'state' parameter equal to either 'New York' or 'NY.' Some entries included a 'state' parameter equal to 'CA', 'MA', and 'NJ' and will need to be removed

What states are the venues in?

```
prelim_venue_data.groupby('Venue S
```

```
Venue State
MA              2
NJ             17
NY           8817
New York      617
Name: Venue State, dtype: int64
```

What venue categories are the entries in?

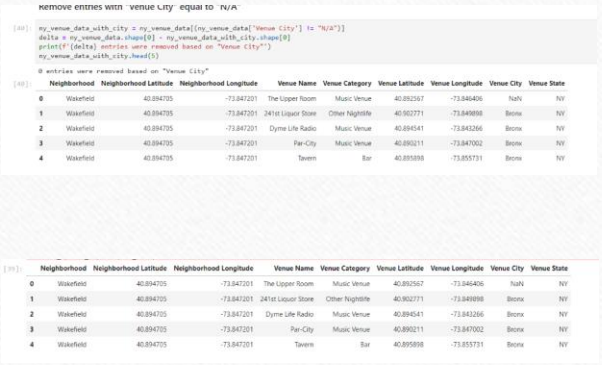
```
n_unique = len(prelim_venue_data['Venue Category'].uniqu
print(f'There are {n_unique} unique venue categories in
prelim_venue_data.groupby('Venue Category')['Venue Categ
```

```
There are 149 unique venue categories in this dataframe
Venue Category
Bar                2094
Lounge              929
Cocktail Bar        500
Nightclub           481
Other Nightlife     476
Music Venue         470
Pub                 341
Wine Bar            278
Sports Bar          264
```

Data Pre-Processing

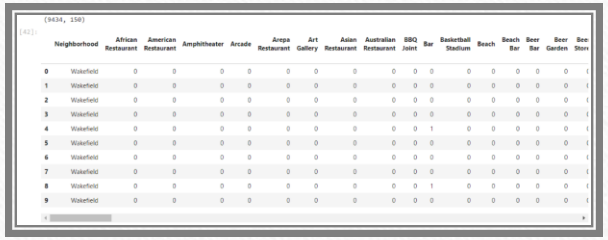
- The preliminary dataset was cleaned according to the Exploratory Data Analysis section above. First, venues located in states other than “New York” or “NY” were removed. Entries with “Venue State” equal to “New York” were changed to “NY.

- Entries returned by Foursquare with no ‘Venue City’ and given the ‘N/A’ treatment were also removed:



OneHotEncoding Venue Categories

- In order to use Foursquare's category values to find similar neighborhoods based on venues, a onehotencoding representation of each entry was created using Pandas' 'get_dummies' function. The result was a dataframe of New York City venues where entry venue category is represented by a value of 1 in the column of matching venue category, as shown below:



Data Visualization

- Venue counts were determined for each venue category using the one hot encoded dataframe:
- Using the dataframe of venue counts shown above, horizontal bar plots were created for select
  venue categories to help visualize the top 25 neighborhoods with the most of each particular
  venue.

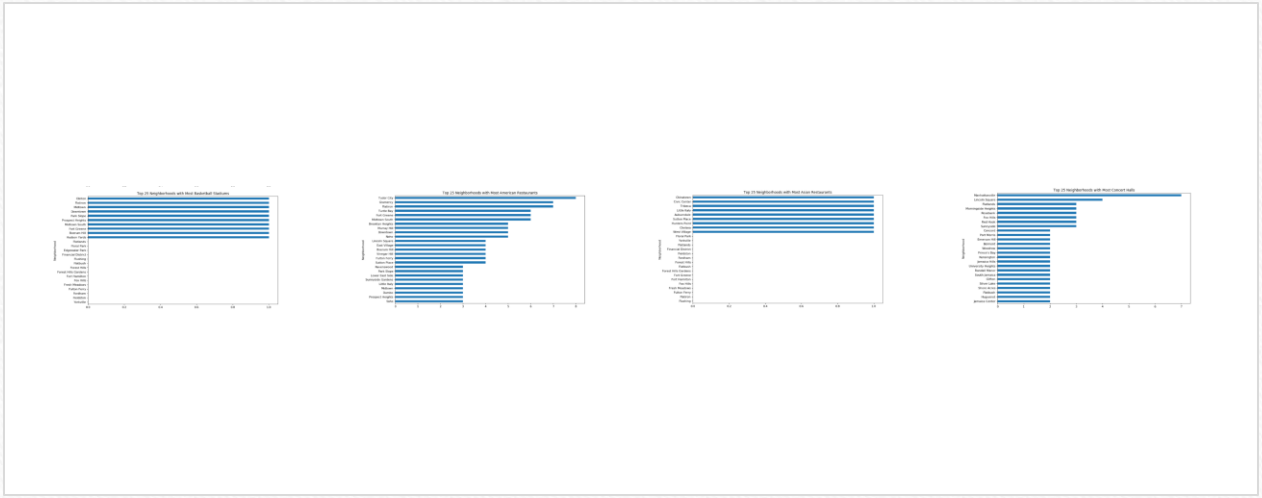
Determine the total amount of venues of each category in each neighborhood

```
venue_counts = my_venue_category_onehot.groupby('Neighborhood').sum()
venue_counts.head(10)
```

Neighborhood	African Restaurant	American Restaurant	Amphitheater	Arcade	Arepa Restaurant	Art Gallery	Asian Restaurant	Australian Restaurant	BBQ Joint	Bar	Basketball Stadium	Beach	Beach Bar	Beer Bar	Bee Garde
Allerton	0	0	0	0	0	0	0	0	0	7	0	0	0	0	
Annadale	0	0	0	0	0	0	0	0	0	2	0	0	0	0	
Arden Heights	0	0	0	0	0	0	0	0	0	3	0	0	0	0	
Arlington	0	0	0	0	0	0	0	0	0	3	0	0	0	0	
Arrochar	0	0	0	0	0	0	0	0	0	3	0	0	0	0	
Arverne	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Astoria	0	0	0	0	0	0	0	0	0	19	0	0	0	0	
Astoria Heights	0	0	0	0	0	0	0	0	0	5	0	0	1	0	
Auburndale	0	0	0	0	0	0	1	0	0	14	0	0	0	0	
Bath Beach	0	0	0	0	0	0	0	0	1	4	0	0	0	0	

```
plot_categories = ['Concert Hall', 'American Restaurant', 'Asian Restaurant', 'Basketball Stadium', 'Piano Bar']
n = 25
for category in plot_categories:
    plt.figure(num=None, figsize=(12, 7), dpi=80, facecolor='w', edgecolor='k')
    plt.title(f'Top {n} Neighborhoods with Most {category}s')
    top_category_neighborhoods = venue_counts[category].sort_values(ascending=False)[0:n]
    top_category_neighborhoods = top_category_neighborhoods.sort_values(ascending=True)
    top_category_neighborhoods.plot.barh(y=category, rot=0)
```

Data Visualization contd.



- We can see the top neighborhoods for selected venue categories.
- Some neighborhoods don't have the venue at all which will be a good information for a new venture to set up business.

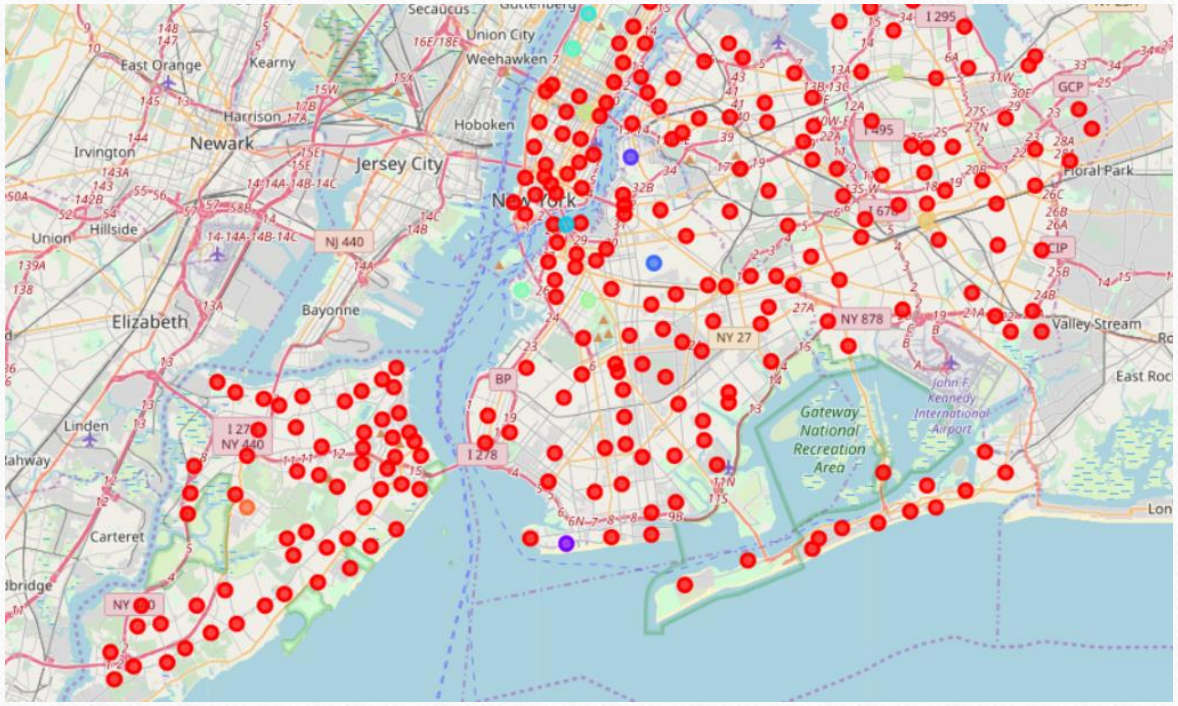
Cluster Modeling

- Scikitlearn's KMeans clustering was used to determine similar neighborhoods based on
  venue percentage. The image below shows the data being scaled and the K-Means model
  being created:

[52]:

	Neighborhood	Latitude	Longitude	Cluster Labels	1st Top Venue Category	2nd Top Venue Category	3rd Top Venue Category	4th Top Venue Category	5th Top Venue Category
0	Wakefield	40.894705	-73.847201	0.0	Smoke Shop	Music Venue	Nightlife Spot	Speakeasy	Other Nightlife
1	Co-op City	40.874294	-73.829939	0.0	Jazz Club	Recording Studio	Sports Bar	Nightclub	Pub
2	Eastchester	40.887556	-73.827806	0.0	Recording Studio	Jazz Club	Sports Bar	Nightclub	Cocktail Bar
3	Fieldston	40.895437	-73.905643	0.0	Piano Bar	Nightlife Spot	Sports Bar	Record Shop	Pub
4	Riverdale	40.890834	-73.912585	0.0	Nightlife Spot	Speakeasy	Music Venue	Hotel Bar	Sports Bar

Cluster Visualization



Conclusion

- Machine learning and clustering algorithms can be applied to multidimensional datasets to find similarities and patterns in the data. Clusters of neighborhoods of similar profile can be generated using high quality venue location data. There is a preface on high quality because analysis models are only as good as the input into them . Foursquare offers a robust 'Places API' service that can be leveragd s in similar studies and modelmaking. This project could be expanded on in a number of different ways. Foursquare's API could be further interrogated to retrieve and consider more related venues in New York City. New datasets of venues can be acquired and potentially merged with what was retrieved from Foursquare. The DBSCAN clustering algorithm, better at maintaining dense clusters and ignoring outliers, could be implemented and compared to KMeans. The clustering model could become the basis for a recommendation system aimed to provide neighborhoods of similar profile to users.

Thank You