Apartment Rental Prices & Venues Data Analysis of Chicago

IBM/Coursera's Applied Data Science Capstone Project - The Battle of the Neighborhoods

Sagar Ippili

A. Introduction

The City of Chicago is the most populous city in the U.S. state of Illinois and the third-most-populous city in the United States. With an estimated population of 2,705,994 (2018), it is also the most populous city in the Midwestern United States. Chicago is the county seat of Cook County, the second-most-populous county in the US, with a small portion of the northwest side of the city extending into DuPage County near O'Hare Airport. Chicago is the principal city of the Chicago metropolitan area, often referred to as Chicagoland. At nearly 10 million people, the metropolitan area is third-most populous in the United States [1].

Being such a crowded city leads the owners of shops and social sharing places in the city where the population is dense. When we think of the people moving to Chicago city, they may want to choose the regions where rental values are lower. At the same time, they may want to choose the neighborhood according to the density of the social place. However, it is difficult to obtain information that will guide investors in this direction, nowadays.

In this project, we will help people who are moving to and looking for renting an apartment in Chicago or Chicago residents who have been already living here. We can answer questions like:

For people moving to Chicago:

- Which neighborhood has cheaper rent
- If a neighborhood is residential or commercial areas

For residents of Chicago city:

- If they are paying more than the average price for their apartment
- If there are similar neighborhoods to theirs with lower rents

When we consider all these problems, we can create a map and information chart where the rental price index is placed on Chicago and each neighborhood is clustered according to the venue density.

B. Data

To consider the problem we can list the data as below:

- I found the Chicago Neighborhood Boundaries' spatial file from the City of Chicago Data Portal. The '. geojson' file has coordinates of all city of Chicago. I used it to create a choropleth map of the Rental Price Index of Chicago [2].
- I used **Foursquare API** to get the most common venues of a given neighborhood of Chicago [3].
- The data on apartments (sqft, number of rooms, postal code, and price) is collected by scraping a local website with apartment listings (zillow.com) [3].
- I used Google Geocoder, Nominatim to get the coordinates for the City of Chicago and neighborhoods [4].

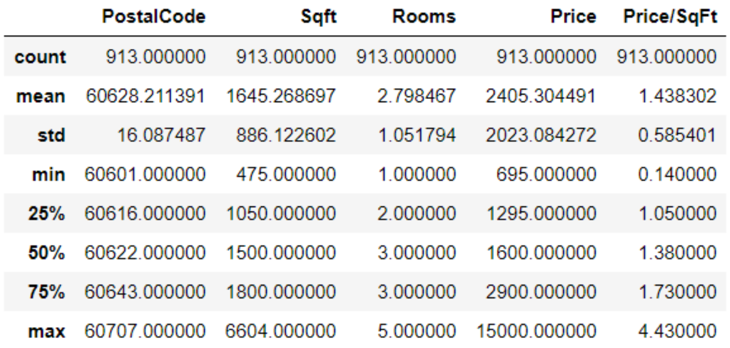
C. Data Wrangling

C.1. Data Scraping

I extracted the apartment rentals listed on Zillow to extract relevant data (Postal Code, Sqft, Rooms, Price) by looping through all pages until no more listings are found using Python's BeautifulSoup module.

C.2. Data Preprocessing

I formatted the extracted data from the Zillow website using the Pandas data frame and its features like formatting the values and dealing with the missing values and introduced a new column 'Price/SqFt' for exploratory data analysis.

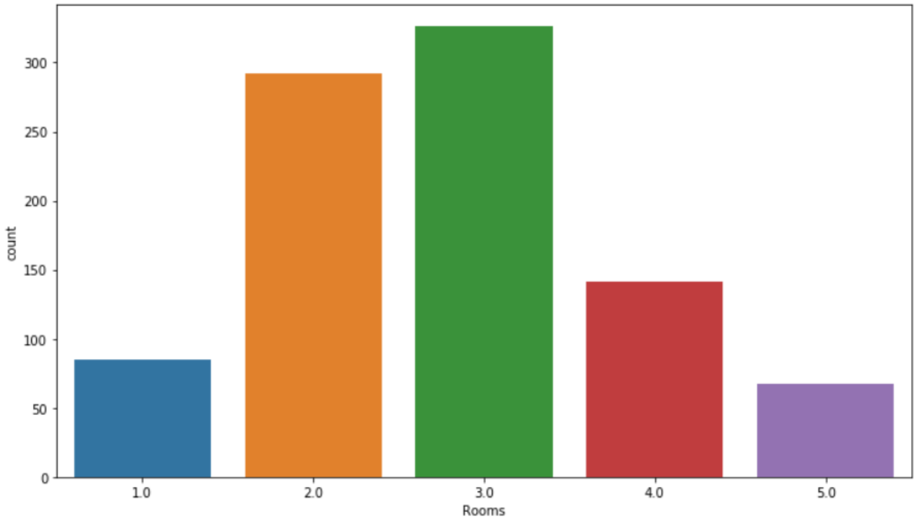


	PostalCode	Sqft	Rooms	Price	Price/SqFt
count	913.000000	913.000000	913.000000	913.000000	913.000000
mean	60628.211391	1645.268697	2.798467	2405.304491	1.438302
std	16.087487	886.122602	1.051794	2023.084272	0.585401
min	60601.000000	475.000000	1.000000	695.000000	0.140000
25%	60616.000000	1050.000000	2.000000	1295.000000	1.050000
50%	60622.000000	1500.000000	3.000000	1600.000000	1.380000
75%	60643.000000	1800.000000	3.000000	2900.000000	1.730000
max	60707.000000	6604.000000	5.000000	15000.000000	4.430000

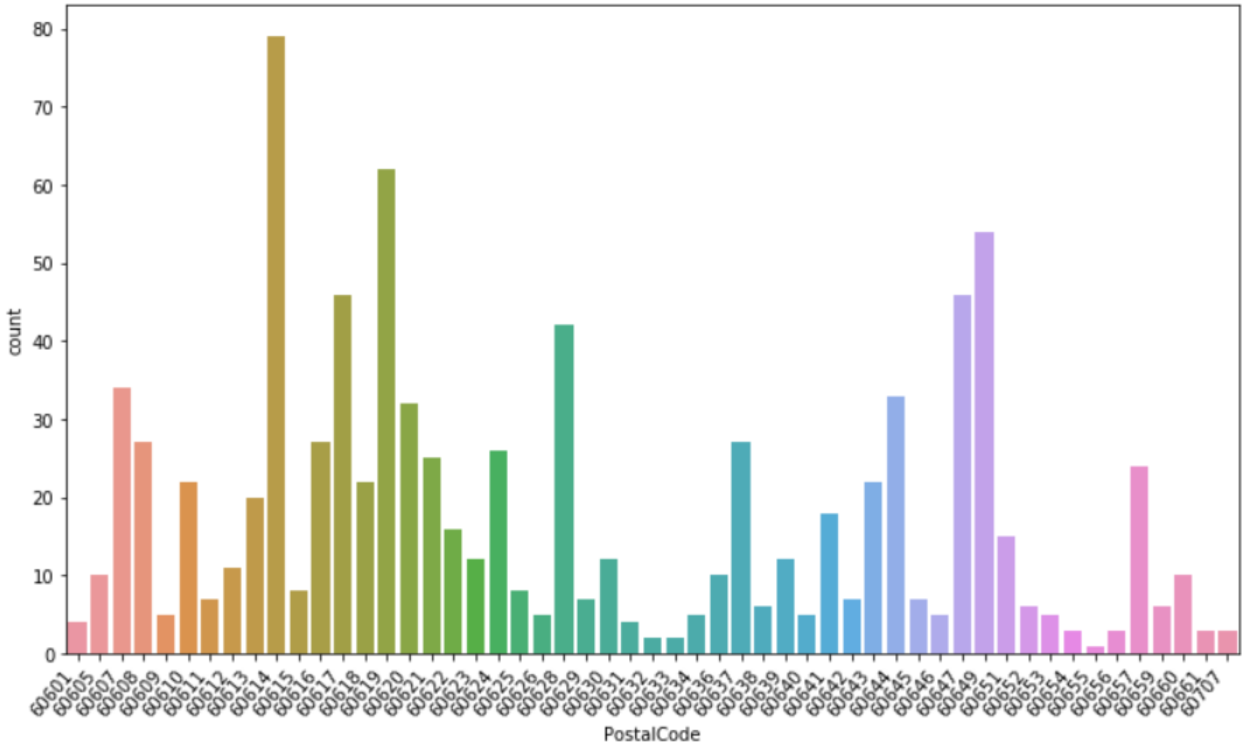
C.3. Data Visualization

I visualized the data using Python's Matplotlib, Seaborn modules with their artist layers to gain insights amongst the columns like Price, Rooms, Neighborhoods, and others as shown below.

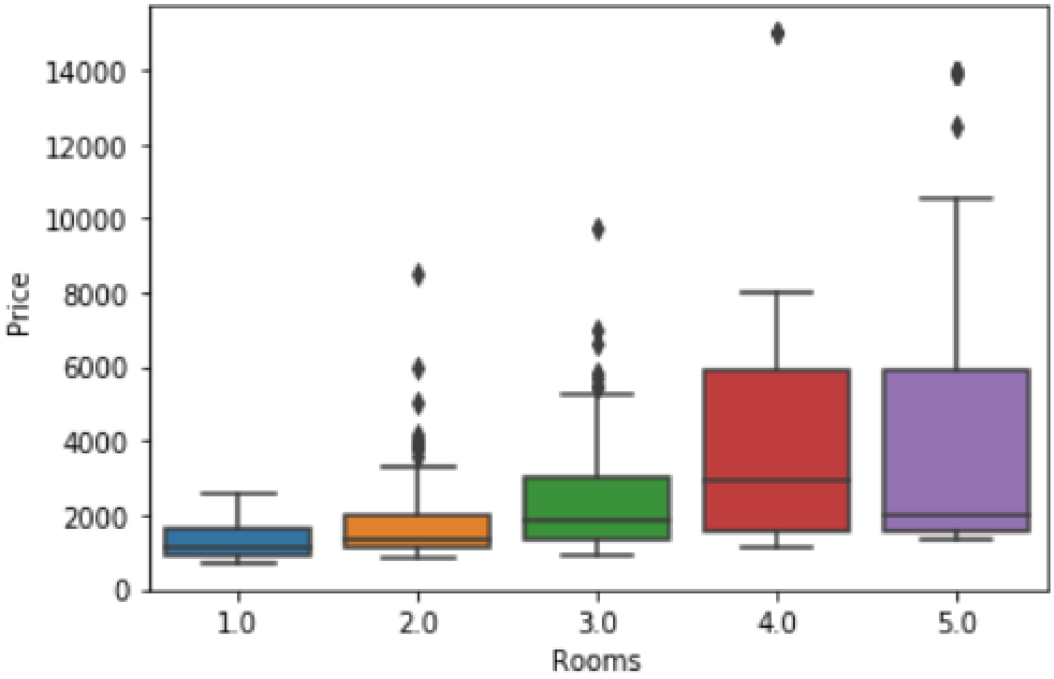
- Count of apartments with different number of bedrooms



- Number of Apartments in each neighborhood (Postal Code)

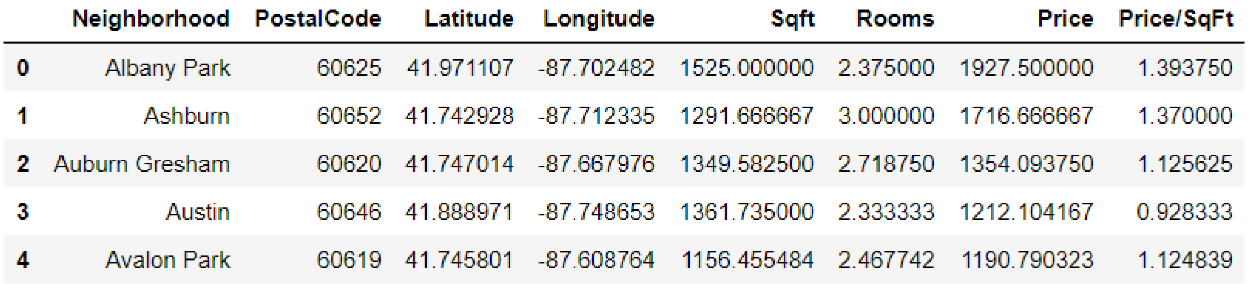


- Distribution of Price amongst the different number of bedrooms



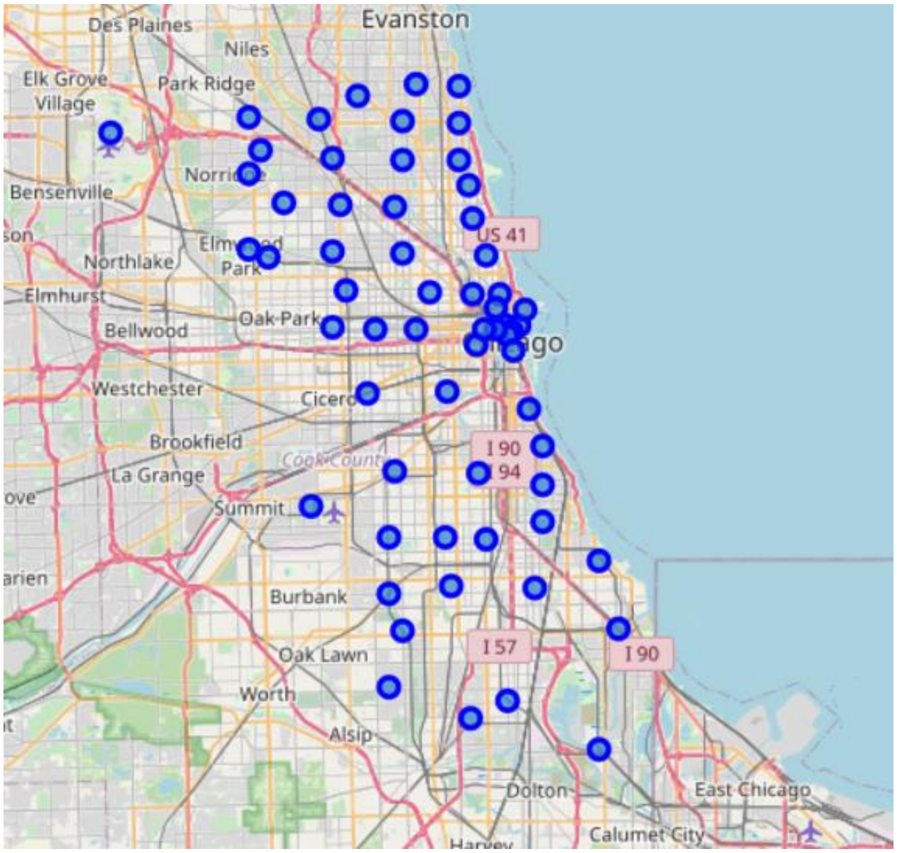
D. Methodology

In this project, I have directed all my efforts on detecting neighborhoods of Chicago that have affordable rental houses. My master data has the main components like *Neighborhood, Location Information, Size of the Apartment in Square Feet, Number of Bedrooms, Price, and Price per Square Feet.*

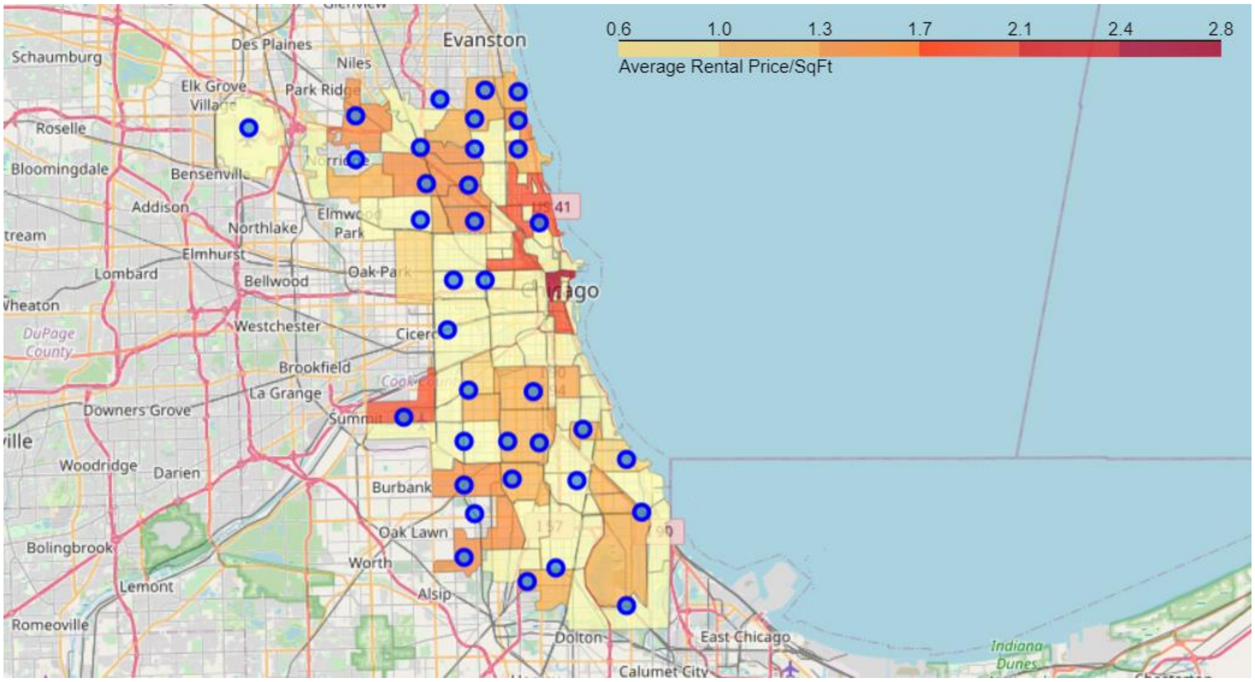


	Neighborhood	PostalCode	Latitude	Longitude	Sqft	Rooms	Price	Price/SqFt
0	Albany Park	60625	41.971107	-87.702482	1525.000000	2.375000	1927.500000	1.393750
1	Ashburn	60652	41.742928	-87.712335	1291.666667	3.000000	1716.666667	1.370000
2	Auburn Gresham	60620	41.747014	-87.667976	1349.582500	2.718750	1354.093750	1.125625
3	Austin	60646	41.888971	-87.748653	1361.735000	2.333333	1212.104167	0.928333
4	Avalon Park	60619	41.745801	-87.608764	1156.455484	2.467742	1190.790323	1.124839

I used the python folium library to visualize geographic details of City of Chicago and its neighborhoods and I created a map of Chicago with neighborhoods superimposed on top. I used latitude and longitude values to get the visual as below:

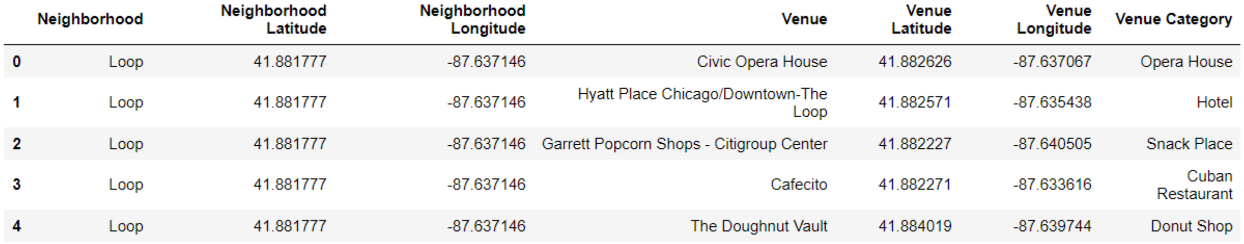


I have also visualized the neighborhoods with average rental prices in each neighborhood using a choropleth map with Chicago neighborhoods superimposed on top using the Folium library. For this purpose, I have downloaded the Chicago city's geospatial file (GeoJSON) from the City of Chicago Data Portal [2].



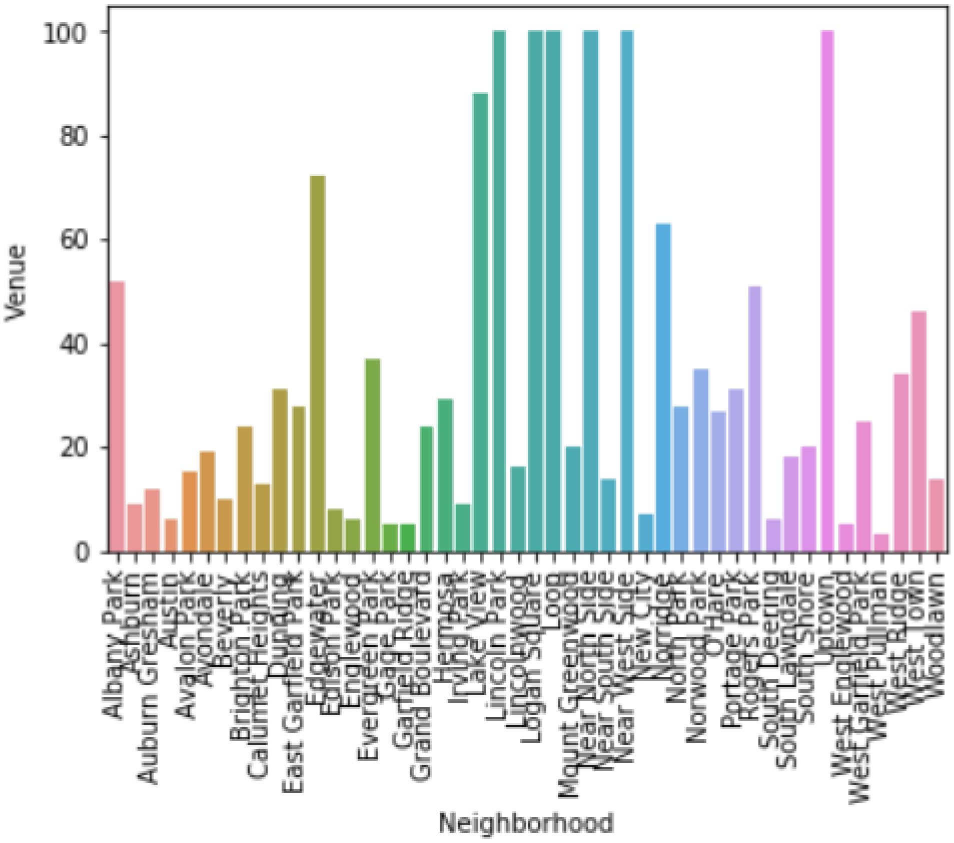
Note: The Color Legend/Scale shows the increase in the average rental price per square feet.

I utilized the Foursquare API to explore the neighborhoods and segment them. I designed the limit as 100 venues and the radius 700 meters for each borough from their given latitude and longitude information. Here is the head of the list for Venues name, category, latitude, and longitude information from Foursquare API.



	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Loop	41.881777	-87.637146	Civic Opera House	41.882626	-87.637067	Opera House
1	Loop	41.881777	-87.637146	Hyatt Place Chicago/Downtown-The Loop	41.882571	-87.635438	Hotel
2	Loop	41.881777	-87.637146	Garrett Popcorn Shops - Citigroup Center	41.882227	-87.640505	Snack Place
3	Loop	41.881777	-87.637146	Cafecito	41.882271	-87.633616	Cuban Restaurant
4	Loop	41.881777	-87.637146	The Doughnut Vault	41.884019	-87.639744	Donut Shop

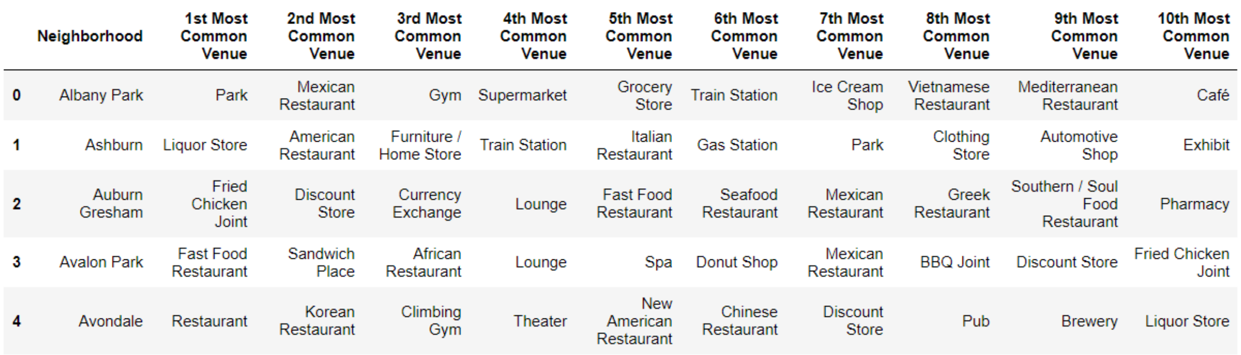
In summary of this data 250 unique venues were returned by Foursquare for all the neighborhoods altogether.



We can see that Edgewater, Lincoln Park, Logan Square, Uptown, and others have reached the limit of 100 venues. On the other hand; Ashburn, Auburn Gresham, Avalon Park, Beverly, Calumet Heights, and other neighborhoods have below 20 venues for the given Latitude(s) and Longitude(s) in the graph below.

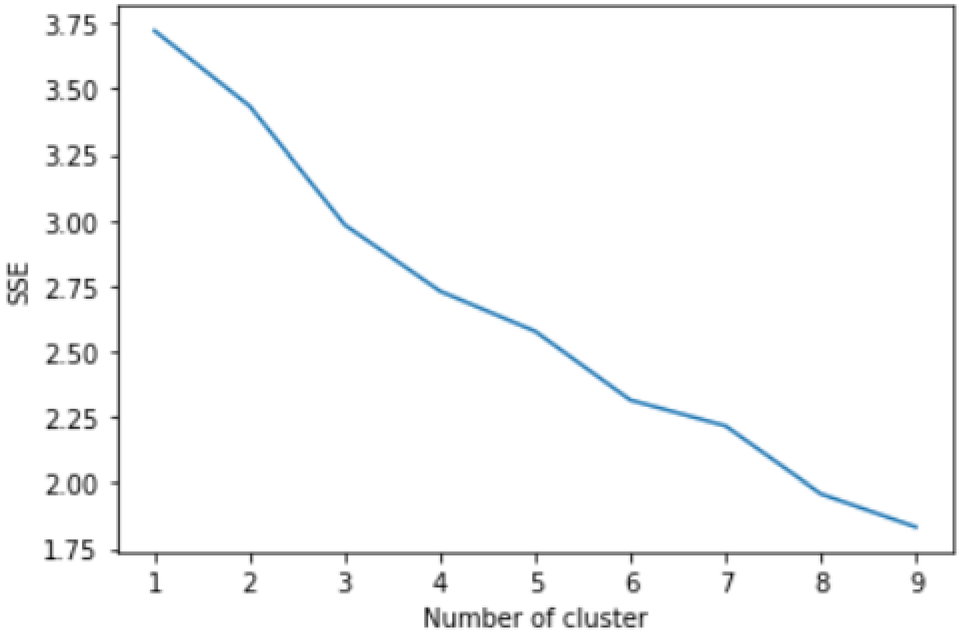
The result doesn’t mean that inquiry run all the possible results in neighborhoods. Actually, it depends on given Latitude and Longitude information, and here, we just run single Latitude and Longitude pair for each neighborhood. We can increase the possibilities with neighborhood information comprising more Latitude and Longitude information.

I created a table with a list of top 10 venue categories for each neighborhood as shown in the table below.

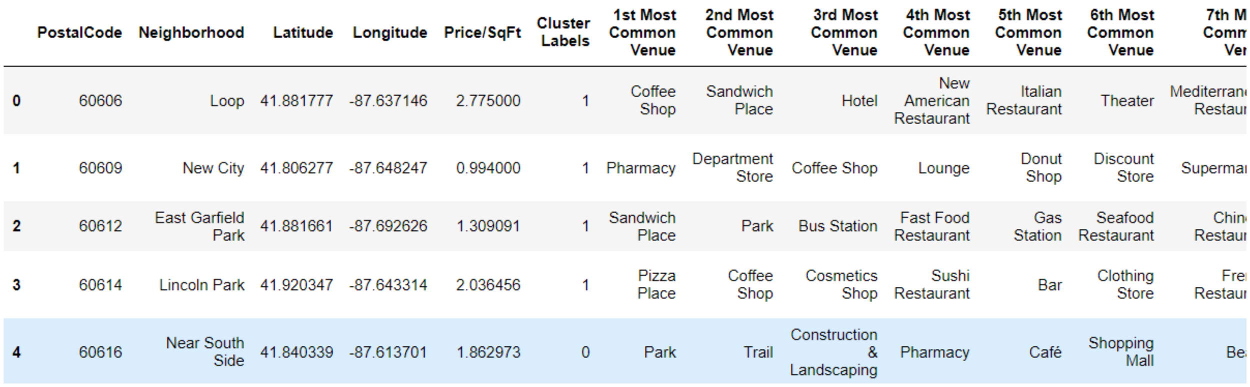


	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Albany Park	Park	Mexican Restaurant	Gym	Supermarket	Grocery Store	Train Station	Ice Cream Shop	Vietnamese Restaurant	Mediterranean Restaurant	Café
1	Ashburn	Liquor Store	American Restaurant	Furniture / Home Store	Train Station	Italian Restaurant	Gas Station	Park	Clothing Store	Automotive Shop	Exhibit
2	Auburn Gresham	Fried Chicken Joint	Discount Store	Currency Exchange	Lounge	Fast Food Restaurant	Seafood Restaurant	Mexican Restaurant	Greek Restaurant	Southern / Soul Food Restaurant	Pharmacy
3	Avalon Park	Fast Food Restaurant	Sandwich Place	African Restaurant	Lounge	Spa	Donut Shop	Mexican Restaurant	BBQ Joint	Discount Store	Fried Chicken Joint
4	Avondale	Restaurant	Korean Restaurant	Climbing Gym	Theater	New American Restaurant	Chinese Restaurant	Discount Store	Pub	Brewery	Liquor Store

Here, we can see there are a few common venue categories amongst the neighborhoods and hence, I used K-means clustering to find similar neighborhoods with this unsupervised learning technique. I used the elbow method to find the optimum number of clusters to decide the **K** value before performing the **K-Means.**



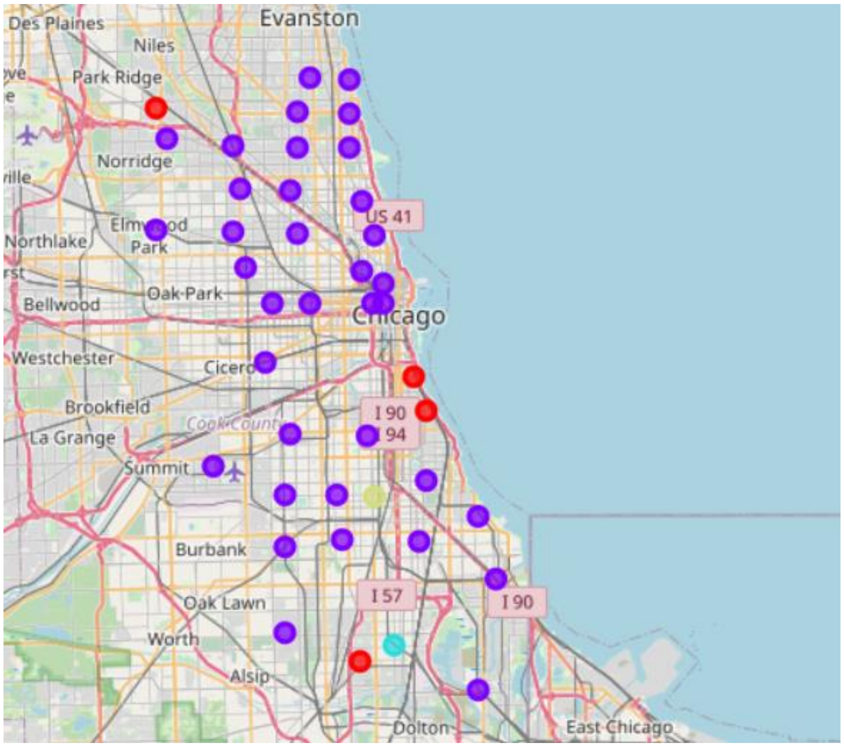
Here is my merged table with cluster labels for each neighborhood.



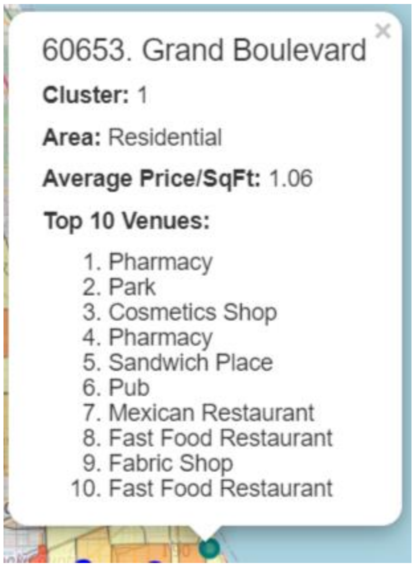
	PostalCode	Neighborhood	Latitude	Longitude	Price/SqFt	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th M Comm Ver
0	60606	Loop	41.881777	-87.637146	2.775000	1	Coffee Shop	Sandwich Place	Hotel	New American Restaurant	Italian Restaurant	Theater	Mediterran Restau
1	60609	New City	41.806277	-87.648247	0.994000	1	Pharmacy	Department Store	Coffee Shop	Lounge	Donut Shop	Discount Store	Supermai
2	60612	East Garfield Park	41.881661	-87.692626	1.309091	1	Sandwich Place	Park	Bus Station	Fast Food Restaurant	Gas Station	Seafood Restaurant	Chin Restau
3	60614	Lincoln Park	41.920347	-87.643314	2.036456	1	Pizza Place	Coffee Shop	Cosmetics Shop	Sushi Restaurant	Bar	Clothing Store	Fre Restau
4	60616	Near South Side	41.840339	-87.613701	1.862973	0	Park	Trail	Construction & Landscaping	Pharmacy	Café	Shopping Mall	Be

E. Results and Discussion

To show the result of the **K-Means Clustering** of Chicago's neighborhoods, I have visualized them on the map of Chicago as shown below.



In the summary section, one of my aims was also to visualize the Average Rental Prices per Square Foot with choropleth style map with pop-up markers having details like:



60653. Grand Boulevard

Cluster: 1

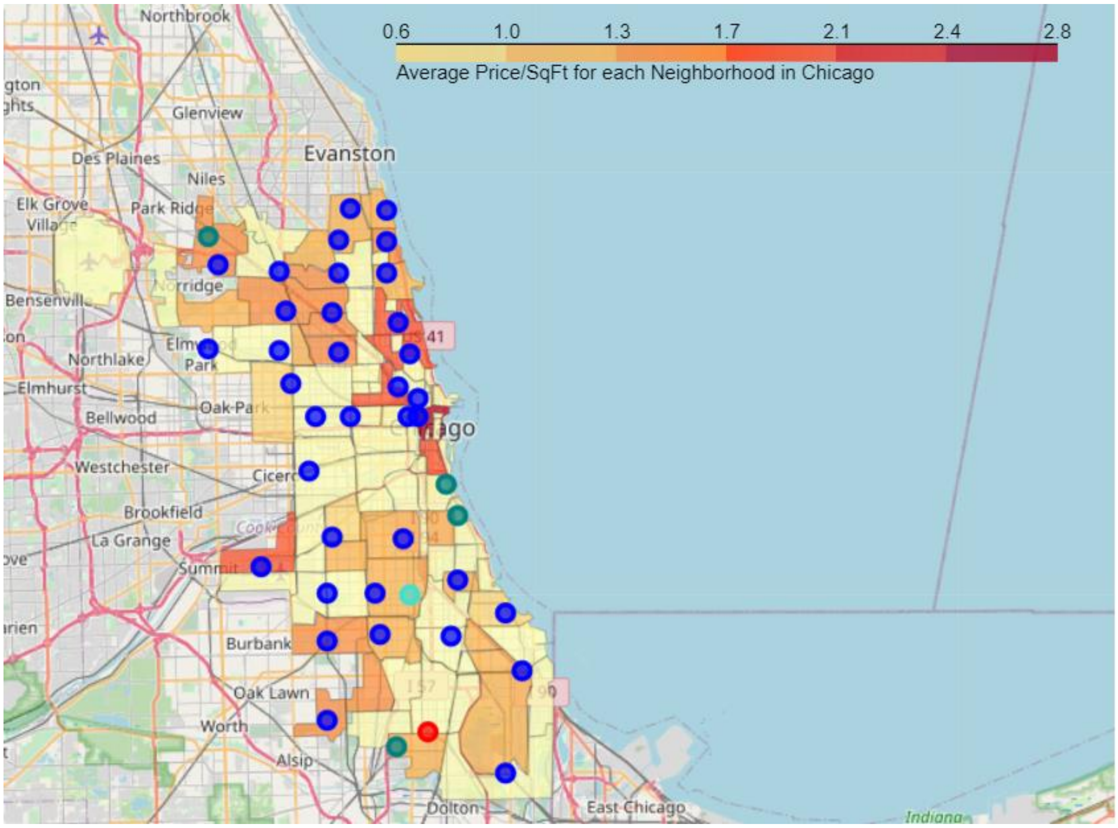
Area: Residential

Average Price/SqFt: 1.06

Top 10 Venues:

1. Pharmacy
2. Park
3. Cosmetics Shop
4. Pharmacy
5. Sandwich Place
6. Pub
7. Mexican Restaurant
8. Fast Food Restaurant
9. Fabric Shop
10. Fast Food Restaurant

- Postal Code - Neighborhood
- Cluster Number
- Area: Commercial/Touristic or Residential
- Top 10 Venues



With all the gathered data we can now create a choropleth map displaying the average price/sqft for each neighborhood as well as display information about the area type and top 10 locations for each neighborhood on the marker labels. With this map, one could determine for example that the Millennium Park Neighborhood is the most expensive to live in. However, by clustering we determined that there are several more similar districts where the price/sqft is significantly lower. Therefore, if someone wants to rent an apartment but cannot afford to live in the Millennium Park, they could look for apartments in the other neighborhoods which are similar in venues but has a much lower price for renting apartments.

F. Conclusion

The purpose of this project was to identify neighborhoods in Chicago close to commercial areas in order to aid city residents/people moving to Chicago in narrowing down the search for the optimal location for an apartment rental. By calculating venue density distribution from Foursquare data, we have first identified general neighborhoods that justify this, and then generated an extensive collection of locations that satisfy some basic requirements. The clustering of those locations was then performed in order to create major zones of interest (containing the greatest number of potential neighborhoods).

A final decision on affordable house rental locations will be made by users based on specific characteristics of neighborhoods, taking into consideration additional factors like price/sqft, social and economic dynamics of every neighborhood, etc.

G. References

- [1] Chicago - Wikipedia
- [2] City of Chicago Data Portal - Facilities Geographic Boundaries
- [3] Foursquare API
- [4] Chicago, IL - Houses for Rent
- [5] Google Maps - Geocodes API