**1. Introduction**

**1.1 Background:**

Opening a restaurant, similar to any other business, involves as extensive decicision-making structure. Some decisions that are made can be reversed given time, such as a menu, a theme, and even a business strategy regarding the type of customers being catered to. However, one decision that can reasonably be considered permanent is the choice of location for a restaurant.

**1.2 The Problem:**

The decision to open a restaurant in a neighborhood is not a decision that should be taken lightly. Once picked, a bad location can be disastrous, necessitating a need to get everything not just alright, but perfect. Needless to say, picking a bad location has been the doom of many a restaurant, and is even touted as one of the major reasons restaurants fail[[1]](#endnote-1). Therefore, it stands to be questions: is it possible to use data to determine which locations are ideal for opening a restaurant?

**1.3 Intended Beneficiaries:**

This project is intended to benefit anyone that is looking to start operating a restaurant in a neighborhood in New York by providing them insights regarding those neighborhoods in New York where restaurants tend to be more successful.

**2. Data Description:**

**2.1 Data Sources:**

The data for this project has been sourced from the FourSquare API, particularly from the neighborhoods from around New York City. The data used herein has been sourced only from FourSquare, although the data pre-processing techniques used herein are easily applicable to a far more diverse range of data sources.

**2.2 Data Description:**

The dataset schema looks as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| venue.id | venue.name | venue.location.lat | venue.location.lng | Likes |
| String | String | Float | Float | Int |

**2.3 Data cleaning:**

Most of the data provided by the FourSquare API was in String format. For instance, the number of likes that each restaurant got was preceded by a string sequence. The first n characters were removed from all rows corresponding to the “likes” column in order to obtain numeric data. The latitude and longitude columns are float types and did not require any further cleaning. There was no missing data. Each venue had a valid name, id, co-ordinates, and likes. In total, there were 65 total venues on the FourSquare API database in NYC.

**3. Exploratory Data Analyis:**

The goal of this project is to cluster restaurants based on their coordinates and the number of likes that they have received. The initial dataset that was extracted from the FourSquare API shows the following distribution of restaurants:

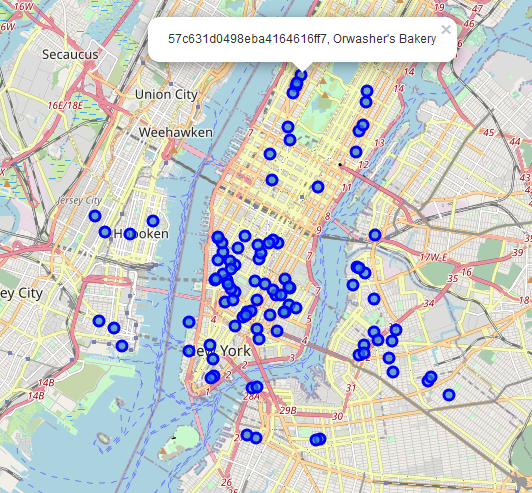


Figure 1 Restaurant locations in NYC marked with Venue Id and Venue Name

Shown below is a histogram plot of the number of restaurants and the likes received by each restaurant:

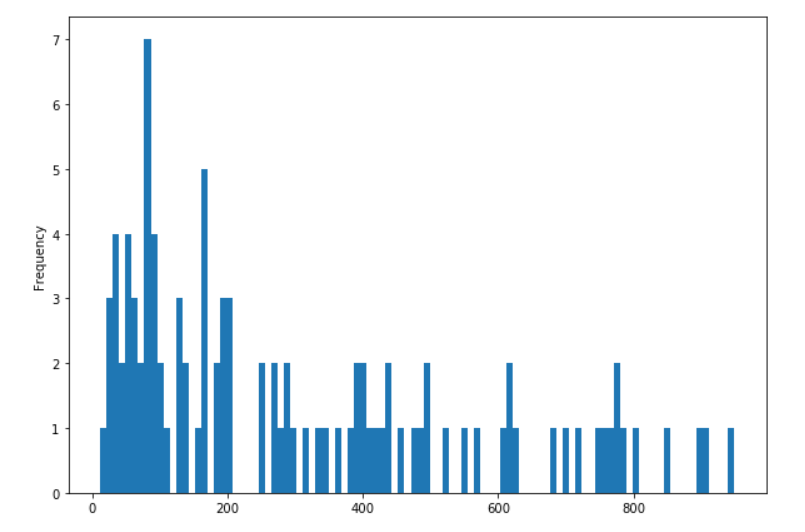


Figure 2 Histogram plot between restaurants and number of likes received

Figures 1 and 2 are representative of the dataset. Figure 1 shows the distribution of the location of the restaurants on a map, whereas Figure 2 shows the distribution of the number of likes each restaurant gets.

An important point of note is present in Figure 2. The histogram is skewed to the right. This means that there is a higher probability for a restaurant to get a lower number of likes. However, some restaurants have managed to receive an incredibly large number of likes, compared to the median.

This has potentially promising implications, as it is possible that these restaurants with a high number of likes could be part of a cluster. If this is the case, then we can show that the location of the restaurant will directly influence the number of likes the restaurant gets. However, if these locations are distributed seemingly randomly, then perhaps there is not much to be gained from location, at least for the dataset being used currently.

**4. Methodology:**

The methodology used herein was that of k means clustering.

**4.1 Determining the number of clusters – Elbow method**

First, we determine how many clusters would be ideal. This was done using the below code:

**sse = []**

**list\_k = list(range(1, 10))**

**for k in list\_k:**

**km = KMeans(n\_clusters=k)**

**km.fit(dtst\_clustering)**

**sse.append(km.inertia\_)**

**plt.figure(figsize=(6, 6))**

**plt.plot(list\_k, sse, '-o')**

**plt.xlabel(r'Number of clusters \*k\*')**

**plt.ylabel('Sum of squared distance')**

dtst\_clustering here is simply the dtst dataframe but with all the String fields (venue.name, venue.id) dropped. The output of the above code is as follows:

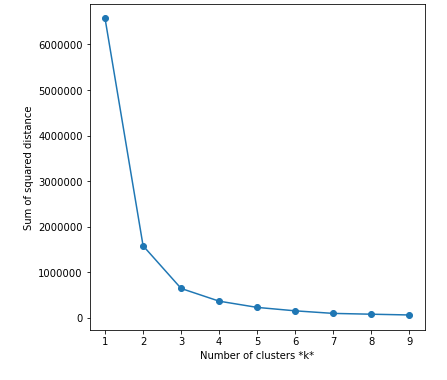


Figure 3 Elbow Method plot

As we can see, the curve starts flattening around k=3. Therefore, we will have 3 clusters in the kmeans clustering method.

**4.2 Clustering**

The clustering is done using the following code:

**kclusters = 3**

**dtst\_clustering=dtst.drop('venue.name',1)**

**dtst\_clustering=dtst\_clustering.drop('venue.id',1)**

**# run k-means clustering**

**kmeans=KMeans(n\_clusters=kclusters, random\_state=0).fit(dtst\_clustering)**

**# check cluster labels generated for each row in the dataframe**

**kmeans.labels\_[0:10]**

**dtst.insert(0, 'Cluster Labels', kmeans.labels\_)**

This will add cluster labels to the dataframe dtst. The cluster labels will be either 0, 1, or 2, as the number of clusters we are using is 3.

Each cluster is then examined using the command:

**dtst.loc[dtst['Cluster Labels'] == 0, dtst.columns[[1] + list(range(5,dtst.shape[1]))]]**

The above code is for cluster labels of 0. This is done for each cluster (labels 0, 1, and 2). On examining each cluster, a pattern seems to emerge: Cluster 0 has the lowest number of likes (tops out at low 100s), Cluster 1 has the highest number of likes (mid 500s and up), and Cluster 2 has an average number of likes(low 100s to 500s).

**4.3** **Adding cluster markers to the map**

Using Folium, we add clusters to the map shown above, to get the following view:

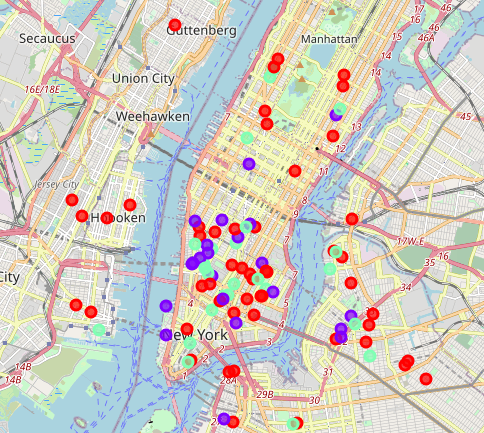


Figure 4Map of Manhattan with clusters added

In the above map, Cluster 0 (lowest) is red, Cluster 1 (highest) is in purple, whereas Cluster 2 (average) is in green.

**5 Results and Discussion:**

From the above map, we can make out areas on the map where purple and green clusters are densely packed, forming ideal candidates for the location of a new restaurant, such as the following:

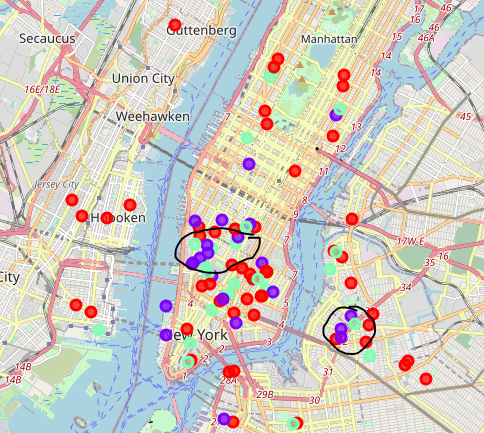


Figure 5 Possible areas for opening restaurant

**6 Conclusion**

Therefore, there does appear to be some correlation between the location of a restaurant and how well it is liked. Thus, we can conclude that there is promising evidence of a restaurant’s location being related to its popularity.

**7. Future direction**

The current dataset used was a fairly small dataset and was suitable for a personal project. However, even the small dataset has provided a couple of locations that look promising in a small radial span. With larger datasets, possibly numbering in the thousands, it might lead to even more promising locations. This combined with rental data of venues could actually help entrepreneurial restaurateurs to get good deals on great locations.

1. <https://www.thebalancesmb.com/ten-reasons-restaurants-fail-2888628> [↑](#endnote-ref-1)