Clustering Similar Neighborhood in Different Cities

1 Introduction

1.1 Background

Different cities in the world are filled with numerous kinds of venues that in turn define the cultures of the cities. A city not only differs from another by means of global positioning, what it to showcase to its inhabitants or tourists has put a significant mark on differentiating it from the rest. Despite of having dissimilarities, it is somewhat possible to group together the similar kind of neighborhoods in different cities. It is possible to segment the different venues in a neighborhood according to venue category, and then to group neighborhoods together that incorporate similar kind of neighborhoods. Having grouped together similar kind of neighborhoods may serve as a variable to help make a decision when people consider moving out of a city to another.

1.2 Problem

Finding identical neighborhoods in different cities in order to help provide a perception of similar neighborhoods which may provide with a great deal of insights in order to make a decision of choosing a neighborhood that is far away, yet somewhat feels like home.

2 Data Acquisition and Cleaning

2.1 Data Sources

This project works with two sets of data. The first dataset consists of New York's different neighborhoods and their respective geometric coordinates, which can be found here. The second dataset consists of Toronto's different borough and their respective postcodes, which can be found here.

2.2 Data Cleaning

The first data source in the described link is in json format. It initially consisted of many different classes of data. Upon examining them, the data that we are interested in was found under 'features'

category. Further formatting of the json data finally resulted in a dataframe that consists of 4 columns, namely: Borough, Neighborhood, Latitude and Longitude.

The second data source is a Wikipedia page that contains Postcode of the city of Toronto in a wikitable. To scrape the data from the URL, BeautifulSoup has been used to extract the table data. After going through a few more steps, the dataframe was obtained which consists of: PostalCode, Borough and Neighborhood.

But the problem with this dataframe was, it has some values under the column 'Borough' which were not assigned in the first place. So, the rows with no assigned value in the 'Borough' column were dropped. Another problem was there were a few rows in the 'Neighborhood' column that too had no values assigned to it. As a solution, the value from the 'Borough' column of the respective row was copied into the 'Neighborhood' column.

2.3 Feature Selection

Now that we have obtained the different neighborhoods and their respective geometric coordinates for the city of New York and Toronto, it is time to come up with different venues that the different venues have to offer.

Foursquare API provides with an access to an enormous database consisting of venues from all around the world including rich variety of information such as addresses, tips, photos and comments. Having signed up for a Foursquare developer, using the Client ID and Client Secret, it is possible to make API requests in order in order to retrieve venue information.

By feeding a function with Neighborhood name and its geometric coordinates, using Foursquare API different venues (Restaurants, Coffee shops, etc) were extracted. After performing One-Hot-Encoding and grouping together the rows by neighborhoods, the NY dataset and Toronto dataset seemed to share 250 features. Both the dataframes were combined into a single dataframe in order to perform clustering operation.

2.4 Dimensionality Reduction

Principal Component Analysis is a dimension-reduction tool that can be used to reduce a large set of variables to a small set that still contains most of the information from the large set. Principal component analysis (PCA) is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components.

Before diving into clustering operation, first we performed dimensionality reduction using Principal Component Analysis on the dataframe in order to reduce the number of dimensions.

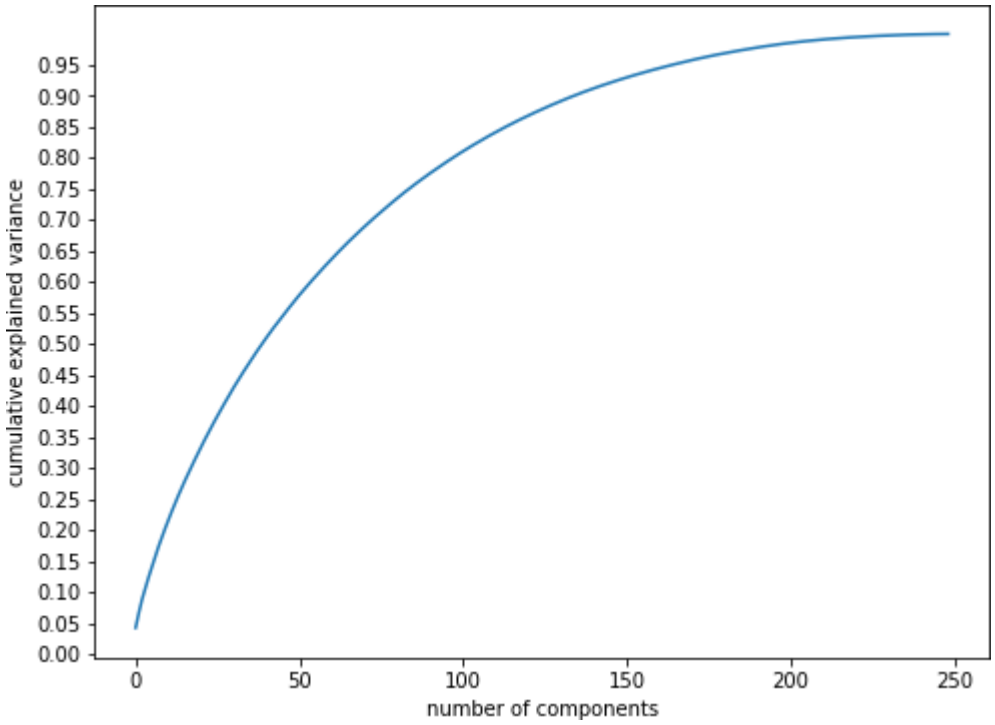


Figure 2.1 Selection of number of Principal Component

Having performed PCA, the number of features was reduced to 150 from 250 yet retaining the maximum variance of the dataset.

3 Methodology

The goal of this project is to group together the similar neighborhoods in the city of New York and Toronto. Since the dataset is unlabeled i.e. unsupervised, this

3.1 Determining Optimal Cluster Number

K-means is a simple unsupervised machine learning algorithm that groups a dataset into a user-specified number (k) of clusters. The algorithm is somewhat naive--it clusters the data into k clusters, even if k is not the right number of clusters to use. Therefore, when using k-means clustering, users need some way to determine whether they are using the right number of clusters. One method to validate the number of clusters is the elbow method. The idea of the elbow method is to run k-means clustering on the dataset for a range of values of k (say, k from 1 to 20), and for each value of k calculate the sum of squared errors (SSE). Then, plot a line chart of the SSE for each value of k. If the line chart looks like an arm, then the "elbow" on the arm is the value of k to be used. The idea is that we want a small SSE, but the SSE tends to decrease toward 0 as we increase k (the SSE is 0 when k is equal to the number of data points in the dataset, because then each data point is its own cluster, and there is no error between it and the center of its cluster). So our goal is to choose a small value of k that still has a low SSE, and the elbow usually represents where we start to have diminishing returns by increasing k.

The second method to find our optimal cluster number is silhouette analysis. Silhouette analysis is a way to measure how close each point in a cluster is to the points in its neighboring clusters. It is a neat way to find out the optimum value for k during k-means clustering. Silhouette values lies in the range of [-1, 1]. A value of +1 indicates that the sample is far away from its neighboring cluster and very close to the cluster it is assigned. Similarly, value of -1 indicates that the point is close to its neighboring cluster than to the cluster it is assigned. And, a value of 0 means it’s at the boundary of the distance between the two clusters. Value of +1 is ideal and -1 is least preferred. Hence, higher the value better is the cluster configuration.

3.2 Random Initialization

Since, K-means incorporates a heuristic approach, it does not ensure converging at global optima at each iteration. Depending upon how the initial position of the clusters were set, it maybe

converge into different local optima. In order to overcome this issue, numerous iterations on different random initializations were performed in order find the best set of convergence.

4 Results

4.1 Optimal Number of Clusters

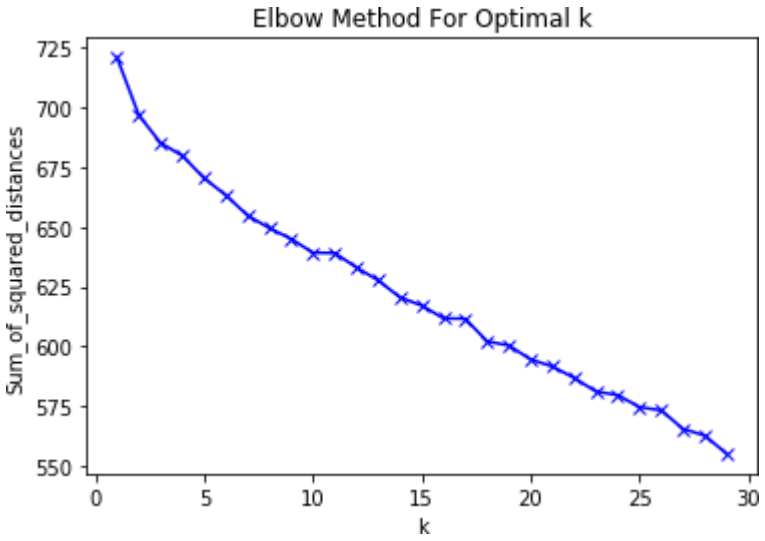


Figure 4.1 Elbow method to determine the number of K

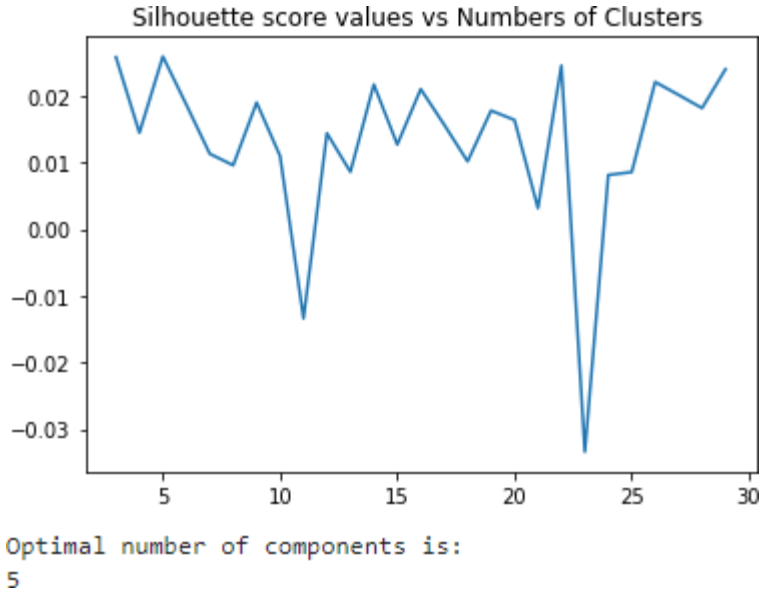


Figure 4.2 Silhouette Score to determine the number of K

The number of clusters being 5 experiences a decrease before it and a gradual regular decrease after it in figure 4.1. The Silhouette score confirms number of clusters being 5 has its peak in figure 4.2. The Silhouette Coefficient is calculated using the mean intra-cluster distance and the mean nearest-cluster distance for each sample. Figure 7 shows the elbow method, in which case, we can't see a distinctive elbow, but we can still come to a conclusion that 15 clusters would be a reasonable choice.

4.2 Visualizing the Clusters on the Map

We created folium maps to help obtain a visual perception of how the very different clusters look on the map when plotted on the map of the city of New York and Toronto.

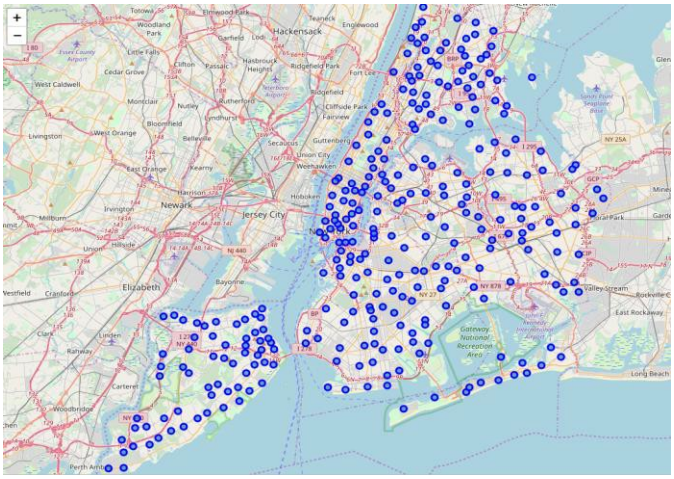


Figure 4.2 (a)

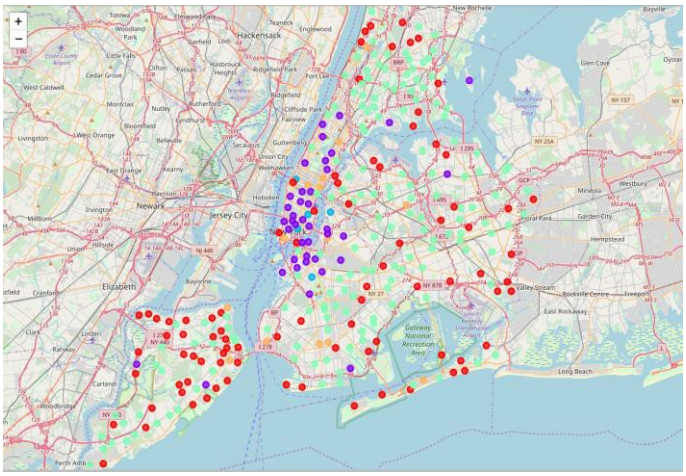
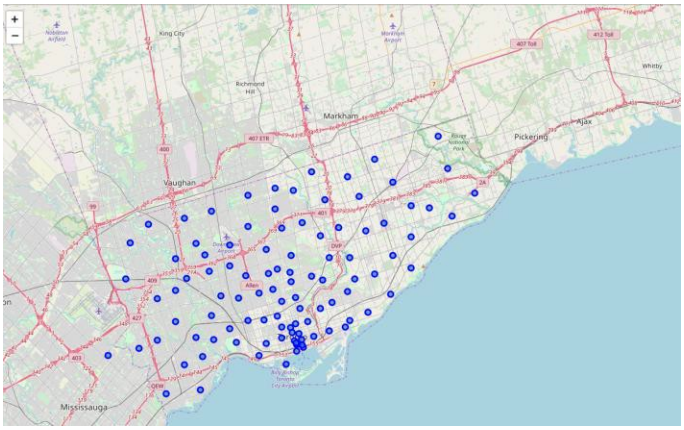


Figure 4.2(b)

Figure 4.2 (a) Venues pinned on NY before Clustering (b) Venues pinned on NY after Clustering

The very different venues in the city of NY have been clustered into 5 distinctive clusters represented by different colors in the Figure 4.2 (b). Now, to comply with the initial goal of finding similar neighborhood in a different city is being visualized in the Figure 4.3 (b). The points of similar color in the Figure 4.2 (b) and again, in the Figure 4.3 (c) represent similar neighborhood in terms of venue information that we obtained from Foursquare API.



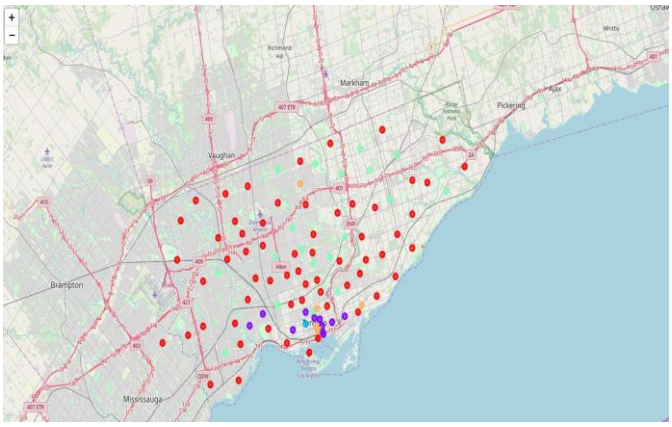


Figure 4.3 (a) Figure 4.3 (b)

Figure 4.2 (a) Venues pinned on Toronto before Clustering (b) Venues pinned on Toronto after Clustering

5 Discussion

Since this is an unsupervised clustering work, many different approaches can be adopted in order to achieve better results. The project was only done on the zip codes of New York and Toronto, each having 150 features, even after performing dimensionality reduction. Having more samples may result in a better clustering

For instance, for the outliers that are being observed on the maps could be defined by using DBSCAN algorithm.

Having dealt with location data on a deeper level, for instance at neighborhood level may result in better grouping of similar data points which eventually may result is better clustering. The study here is being ended by visualizing the data and clustering information on the map of the City of New York and Toronto.

6 Conclusion

People are frequently moving into new cities. And in this ever growing world filled with technology, having a neighborhood recommendation based on location data is something to be considered basic now-a-days. And the application of neighborhood segmentation lies beyond this application too. This can serve to be an impressive tool to better organize a city resources. Furthermore, it can be used as a tool for security measurement if combined with crime data.