The Battle of the Neighborhoods

RINI CHRISTY

April 21, 2019

Table of Contents

1. Introduction: Marketing Consultancy Problem

The marketing teams involved in Travel, Tourism & Housing are faced with the need to provide ever more quality information that is tailored to the individual customer's needs. In this era of information overload, it's easy to get overwhelmed with the information on all the seemingly available choices, without being able to get the best use out of all the information

Customers today are highly aware of the market and as a result the industry has become highly competitive. This makes it essential to understand the client's need in the shortest possible time and present the options most suitable for their tastes.

For my project, I present the example of **a travel advisor who was approached by a client looking for a neighborhood to stay for a short term, in and around Toronto. The client has provided some information about his tastes and preference towards some neighborhoods he likes/ dislikes the most. The challenge for this advisor lies in understanding & exploiting client information that lacks linear structure and using it to recommend location clusters. Additionally, the sheer volume of predictions of possible neighborhoods also poses a challenge.**

The Toronto area is spread over 630.2 square kilometers and like any big city has a mix of neighborhoods with varying density of shops, establishments, open areas and recreation facilities. One of the big factors in such a search for suitable neighborhood is obviously the available budget, but there is also the question of the entire makeup of the neighborhood. For the purposes of this project, the budget is not being taken into consideration and the focus will be on the facilities and infrastructure. To bring all this information on a visual scale which is easily understandable and also scalable with the ability to focus or zoom in further will help the client get a cleaner understanding of the options matching their outlook and lifestyle. This project would be extremely helpful to people considering moving to Toronto, as they would have all the information needed to make a decision on where in Toronto to live.

2. Methodology: Data Collection to Clustering, Recommendation & Classification

2.1 Project Objective

To Explore, cluster, recommend the neighborhoods of Toronto using k-means Clustering, Recommender Systems and then classify the clients' profile based on their neighborhood preferences, applying various Machine Learning Algorithms.

2.2 Data collection

This project focuses first on applying well-known machine learning algorithms to the dataset available from Wikipedia, Statistic Canada Census Official website & Foursquare. The first task is to define the data requirements for the Segmentation, Recommendation & Classification approach for the Toronto area. This data is then modified as well as combined with various algorithms in the subsequent steps so that it produces the correct classifications matching those on the Customer Profile dataset.

2.3 Data Definition

The content, format, and representations of the data needed for clustering, recommendation, and classification are defined & the explicit combination of information extraction and machine learning are executed. In this phase the data requirements are revisited and decisions are made as to whether or not the collection requires more or less data.

Exploratory Data Analysis (EDA) techniques such as descriptive statistics and visualization can be applied to the data set, to assess the content, quality, and initial insights about the data. Gaps in data will be identified and plans to either fill or make substitutions will have to be made.

2.4 Neighborhood Clustering

Clustering Algorithm produces taxonomy of location properties, namely neighborhood, with use of k-means method. The developed approach separates groups (clusters) of neighborhood with similar characteristics, which do not depend on spatial location. The segmented neighborhood data segregates the neighborhoods based on venue categories to recommend a potential buyer or renter or traveler based on their preference and with the description of a property of interest available.

After clustering for each category, the cluster centers were chosen to represent the category and they become the new training sets for Classification Algorithm.

2.5 Neighborhood Recommendation

Recommending products to consumers is a popular application of machine learning, especially when there exists substantial data about the customer’s preferences. Even though peoples’ tastes may vary, they generally follow patterns. There are similarities in the things that people tend to like ie., they tend to like things in the same category or things that share the same characteristics.

One of the main advantages of using recommendation systems to my case study is that client gets a broader exposure to many different neighborhoods he might be interested in. The idea was that people who prefer a particular destination are more likely to select a neighborhood from the same neighborhood clusters. Not only does this provide a better experience for the client but it benefits the consultant, as well, with increased potential revenue and better results for its customers.

Content-based Recommender systems try to figure out what a client's favorite aspects of a neighborhood are, and then make recommendations on neighborhoods that share those aspects. The recommendation in a content-based system is based on client's liking for a particular neighborhood and the nature of the venue categories contained in that neighborhood.

2.6 Final Prediction - Client Profile match

The match of the client profile with the most appropriate neighborhood cluster is the final stage where the information is now distilled to provide a wide range of information in a highly focused manner. This is achieved by executing classification algorithms with KNN, Decision tree, Support Vector Machine. Finally the evaluation of each classification model is done with a view to ensure its accuracy.

2.7 Final Visual Representation

Clients often feel the need to switch to different sources in search of complete information, but visual tree has been found to be more effective in customer engagement as it gives a personalized comprehensive information at a glance. The Decision Tree visualization will ease the client's understanding of the information, giving them more flexibility in discovering, using, modifying, and updating the available information according to their taste & preferences and thereby enhancing their Decision-making process.

3. Data Collection

3.1 Wikipedia List of Postal Codes of Canada

The data required for this project is collected from the following websites:

1. Wikipedia List of Postal Codes of Canada
2. Geospatial Data
3. Canada Census population data from 2016
4. Foursquare Data

`

I start with scraping the Wikipedia page in order to obtain the data that is in the table of postal codes and to transform the data into a pandas dataFrame, then use geocoder to fetch the coordinates data and merge with neighborhood data. Finally, I will apply the Foursquare API to explore venues for all neighborhoods in Toronto and analyze and visualize the clustering of neighborhoods.

In order to explore and cluster the neighborhoods in Toronto, the Toronto neighborhood data of postal codes of each neighborhood along with the borough name and neighborhood name, is obtained from the Wikipedia page,

https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M,

Using BeautifulSoup package in Python the table on the Wikipedia page is scrapped which is then wrangled, cleaned, and then read into a structured format like pandas dataFrame. I use the 'request' and 'BeautifulSoup' libraries to get the 'lxml' file and then find the all table tags, build a loop to extract all data and store as dictionary for subsequent dataFrame generation. Once the data is in a structured format, the analysis is done to explore and cluster the neighborhoods in the city of Toronto.

3.2 Geospatial Data

From the csv file, (http://cocl.us/Geospatial_data). a dataFrame containing the geographical coordinates corresponding to each postal code is created.

3.3 Statistics Canada Census data

The demographic status of Toronto can be explored extensively after reading the Statistics Canada Census data for Population, 2016 into the Pandas dataFrame. Canada Census population data, 2016 from (https://www12.statcan.gc.ca/census-recensement/2016/dp-pd/hlt-fst/pd-pl/Tables/CompFile.cfm?Lang=Eng&T=1201&OFT=FULLCSV) is read into a pandas dataFrame and filter for the columns and rows of interest (Population, 2016 for Ontario & Postal codes of Toronto).

3.4 Foursquare location data

Location data describing places and venues, such as their geographical location, their category, working hours, full address, and so on needs to be gathered. Once all the data ingredients are collected, I will have a better understanding of what I will be working with in the data collection stage. The data should be such that for a given location in the form of its geographical coordinates (or latitude and longitude values), one is able to determine what types of venues exist within a defined radius from that location. So for a given location I will be able to tell whether there are restaurants nearby, or other facilities, institutions such as schools, banks, parks, or gyms, or community centers and also the density of these facilities in each neighborhood.

Geocoding: To utilize the Foursquare location data, I need to get the latitude and the longitude coordinates of each neighborhood. To convert the Toronto address into latitude and longitude values, a search engine for OpenStreetMap data geocoding tool named Nominatim is employed.

Using the Foursquare API, I searched for the type of venues or stores around the Nominatim returned location coordinates. By making the call to the database entering the developer account credentials, which are my Client ID and Client Secret as well as what is called the version of the API.

Again because I'm searching for different type of venues, I pass the latitude and longitude coordinates along with the search query radius & limits. This completes the URI to make the call to the database and in return a .JSON file format of the venues that match the query with its name, unique ID, location, and category information is downloaded.

I get a .JSON file of the venues with its name, unique ID, location, and category. Apply the get_category_type function from the Foursquare lab, followed by cleaning the .JSON file and structuring it into a pandas dataFrame.

To repeat the same process to all the neighborhoods in Toronto getNearbyVenues function is created and thus retrieves all Toronto Neighborhoods with its Venues & Venue categories.

Now I am ready to explore the venues in the city of Toronto.

4. Results & Discussion

4.1 Exploratory Data Analysis (EDA)

Various exploratory data analysis (EDA) are done with the data using Matplotlib, Seaborn, Population, Total private dwellings, and Private dwellings used by residents and they show a very high significant positive correlation among each other. With the population keeps increasing, the number of dwellings inhabited by both residents and non-residents keep increasing. A joint plot (**Figure 1**) with histogram and the regression analysis of change in private dwelling with population in the city of Toronto shows a correlation coefficient of 0.907 and the relationship is shown with the linear regression equation:

Total number of private dwellings = 0.398 * Population + 902

I can easily predict an increase in the number of dwellings with an increase in population in future years in Toronto with this modelled equation.
A PairGrid combining kernel density estimation (kde) plot, histogram and scatter plot of population, total private dwellings, & dwellings occupied by private residents (**Figure 2**) is plotted.

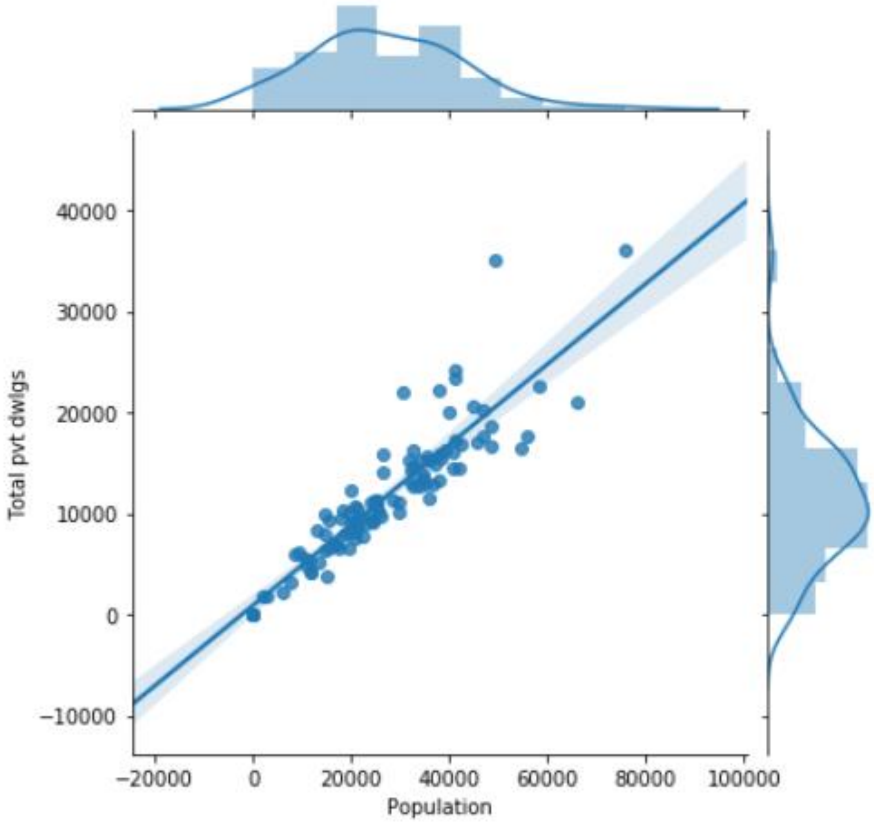
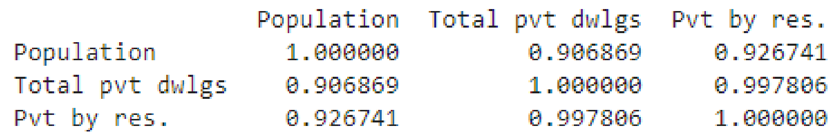


Figure 1: Joint plot of relationship of total private dwellings with population in Toronto.

*The correlation coefficients for the relationship among different pairs are listed below (*Table 1*):*



```
                 Population  Total pvt dwlgs  Pvt by res.
Population         1.000000         0.906869     0.926741
Total pvt dwlgs    0.906869         1.000000     0.997806
Pvt by res.        0.926741         0.997806     1.000000
```

Table 1: Pair-wise correlation coefficients

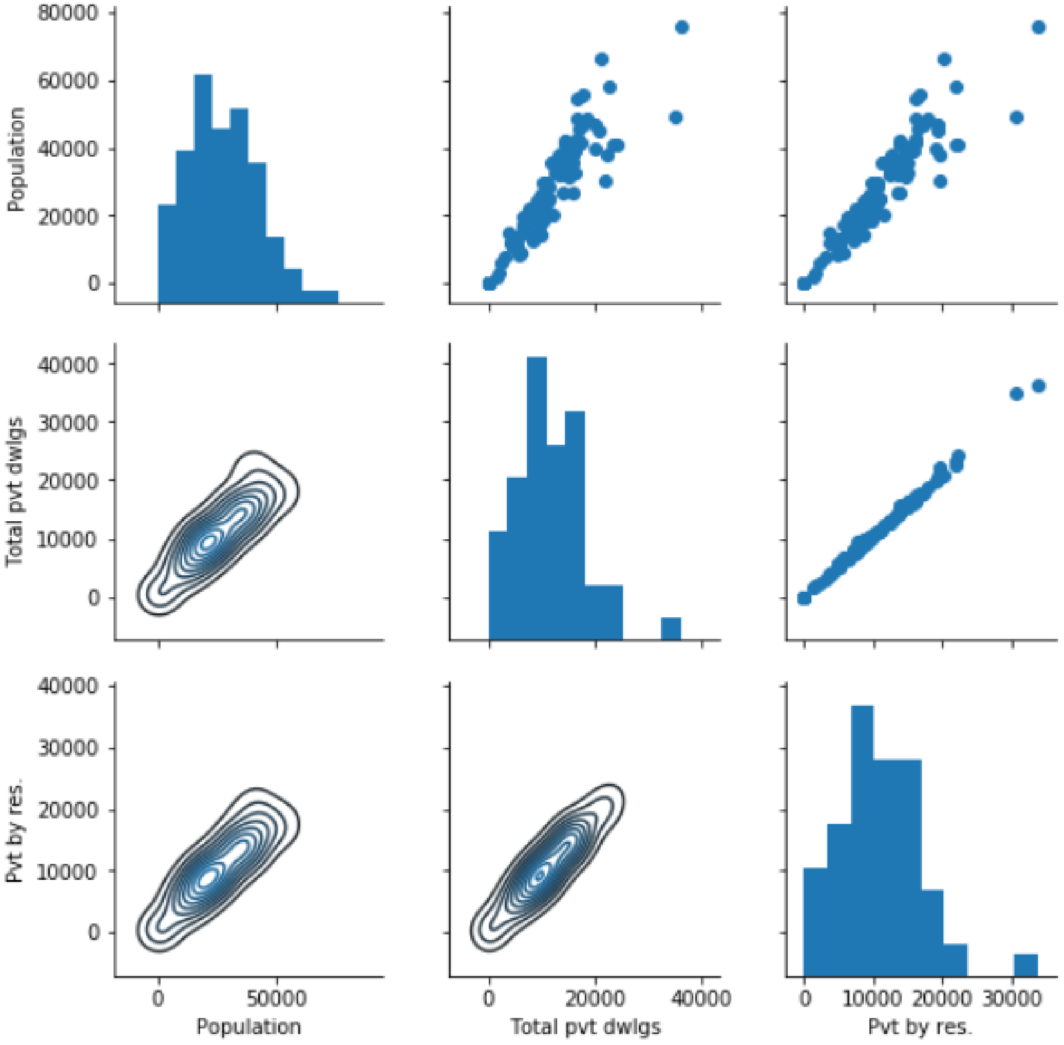


Figure 2: A PairGrid combining kernel density estimation (kde) plot, histogram and scatter plot of population, total private dwellings, & dwellings occupied by private residents.

- **Choropleth Map to show the Distribution of Population across the City of Toronto**

Geographical data visualization is a fundamental tool for communicating results related to geospatial analyses, and for generating hypotheses during exploratory data analysis. The constantly increasing availability of geolocated data from various spatial databases implies the need for new tools for exploring, mining and visualizing large-scale spatial datasets. The python programming language has been gaining attention as a data analysis tool in the scientific community thanks to the clarity and simplicity of its syntax, and due to an abundance of third-parties libraries e.g. within many disciplines including scientific computing, machine learning.

Now, I create a Choropleth map (Figure 3) for Toronto population data obtained from Statistics Canada census website. As this requires structured data in .GEOJSON format as an input, I generated one, fetching a shape file of Census Boundary Files data for Canadian FSAs (Forward Sortation Area—the first three digits of the Canadian Postal Code) from a publicly accessible API endpoint provided by Statistics Canada. The .shp file downloaded is then converted to .GEOJSON file using QGis, filtering all Toronto CFSAUID (FSA) corresponding to the postal codes in the dataFrame, Toronto_data.

Linking the FSA stored in the key feature.properties.CFSAUID, a choropleth map of Toronto population, which provides a detailed demographic information about each neighborhood is generated. Choropleth map is superimposed on top of Folium which is a Python map rendering Library that is handy to visualize spatial data in an interactive manner, straight within the notebooks environment. It is quite straight forward provided a GeoJSON file for the area is available. The default map style is the open street map, which shows a street view of an area when it is zoomed in. First a map centered around Toronto is created by passing in the latitude and the longitude values of Toronto using the location parameter. With Folium the initial zoom level can be set and here I use the zoom start parameter as 10. The zoom level can be changed easily after the map is rendered by zooming in or zooming out.

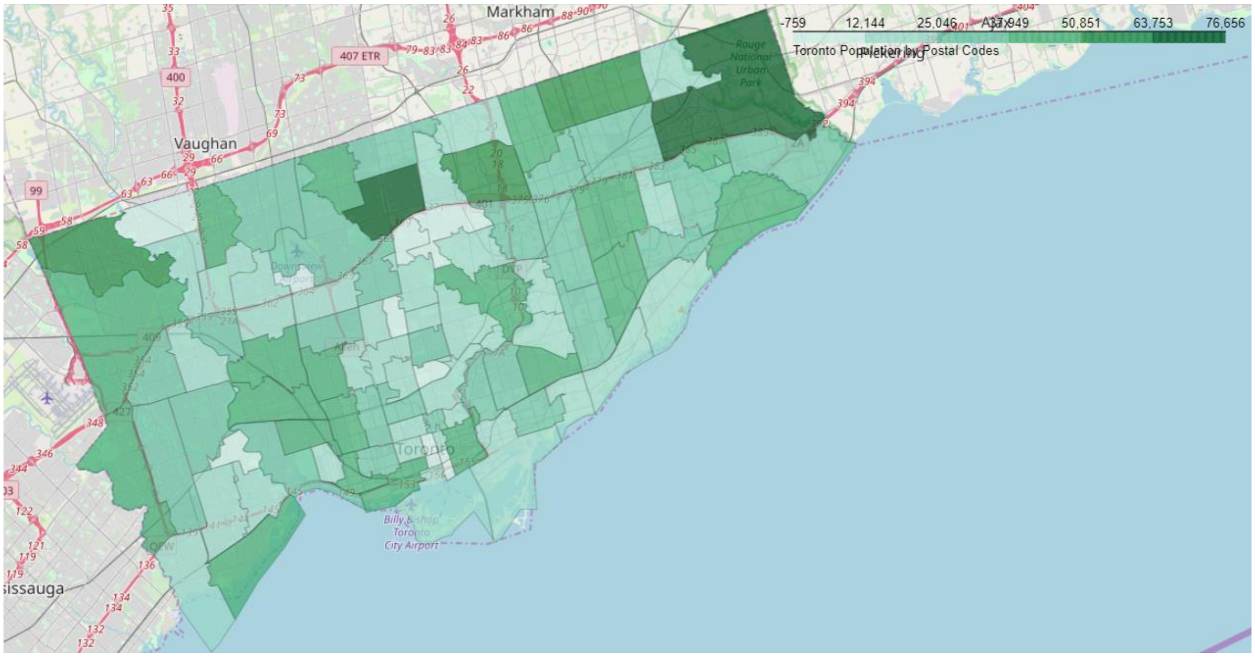

-759
12,144
25,046
37,949
50,851
63,753
76,656
Toronto Population by Postal Codes
Markham
Vaughan
Pickering
Rouge National Urban Park
Toronto
Billy Bishop Toronto City Airport
issauga

Figure 3: Choropleth Map showing the distribution of Population in the city of Toronto

- **Heat Map of Latitudes & Longitudes of Points of Interest**

A Heat map of points of interest (Figure 4) is also generated using latitude and longitude values obtained

from the merged data of Wikipedia, statcan & csv file data. Circular markers are added to specify the location of the neighborhoods. To do that, a for loop is used to iterate through each row and then adding them to the map I created as before for choropleth. The columns to be iterated are specified and hence all the blue circular marks are superimposed on top of the map. Labels are also added to these markers in order to let other people know what they actually represents. This is done by using the marker function and the pop up parameter to pass in neighborhood, borough names to add to this marker.

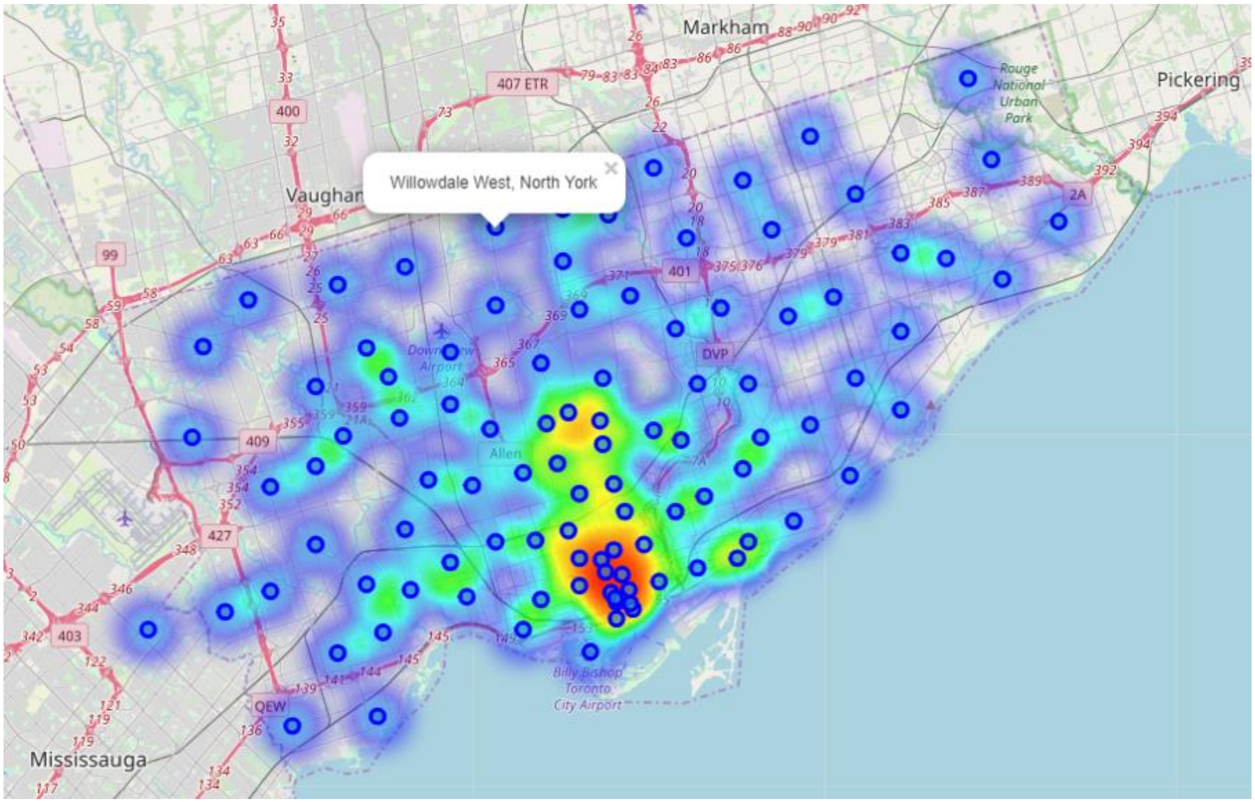
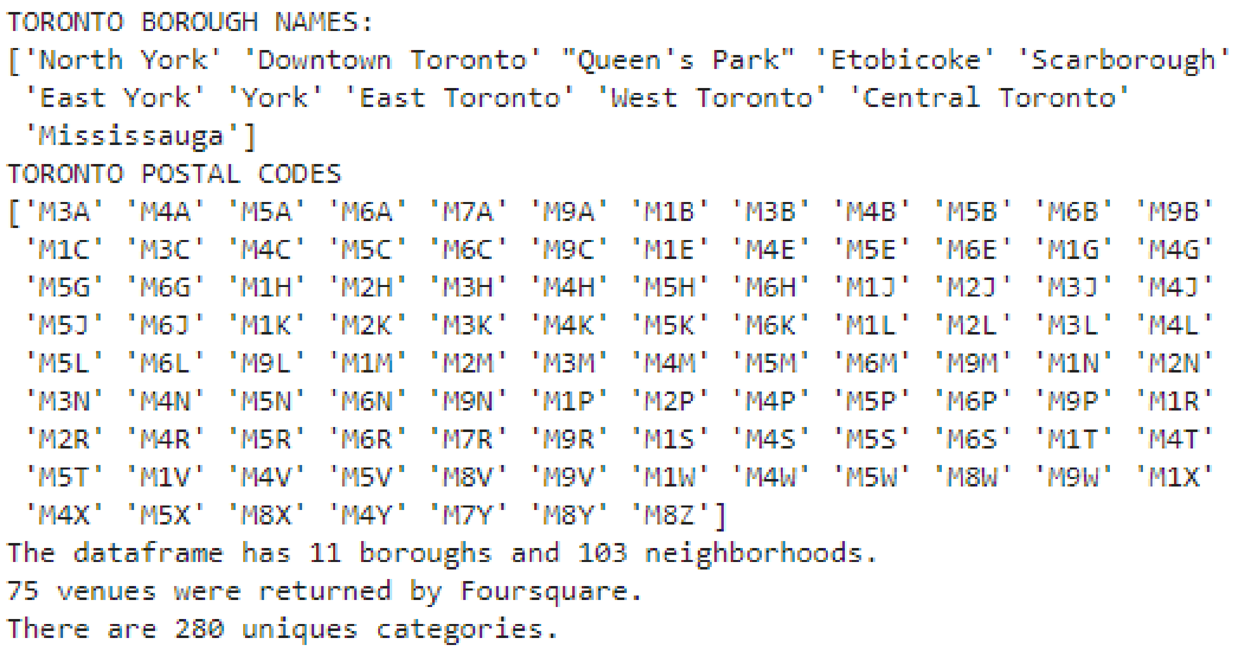

Willowdale West, North York
Markham
Pickering
Vaughan
Mississauga
407 ETR
Rouge National Urban Park
Downsview Airport
Allen
DVP
QEW
Billy Bishop Toronto City Airport

Figure 4: Heat map showing Latitudes & Longitudes of points of interest in the Toronto area.

Detailed numerical analysis carried out for Toronto showed 11 boroughs, 103 Neighborhoods with unique Postal Codes, 72 venues consisting of 280 uniques categories. The list of Toronto borough names & postal codes are given below (**Table 2**). Most common venues were in the Coffee Shop, Café, Restaurant, Pizza Place, Italian Restaurant categories (Figure 5).



```
TORONTO BOROUGH NAMES:
['North York' 'Downtown Toronto' "Queen's Park" 'Etobicoke' 'Scarborough'
 'East York' 'York' 'East Toronto' 'West Toronto' 'Central Toronto'
 'Mississauga']
TORONTO POSTAL CODES
['M3A' 'M4A' 'M5A' 'M6A' 'M7A' 'M9A' 'M1B' 'M3B' 'M4B' 'M5B' 'M6B' 'M9B'
 'M1C' 'M3C' 'M4C' 'M5C' 'M6C' 'M9C' 'M1E' 'M4E' 'M5E' 'M6E' 'M1G' 'M4G'
 'M5G' 'M6G' 'M1H' 'M2H' 'M3H' 'M4H' 'M5H' 'M6H' 'M1J' 'M2J' 'M3J' 'M4J'
 'M5J' 'M6J' 'M1K' 'M2K' 'M3K' 'M4K' 'M5K' 'M6K' 'M1L' 'M2L' 'M3L' 'M4L'
 'M5L' 'M6L' 'M9L' 'M1M' 'M2M' 'M3M' 'M4M' 'M5M' 'M6M' 'M9M' 'M1N' 'M2N'
 'M3N' 'M4N' 'M5N' 'M6N' 'M9N' 'M1P' 'M2P' 'M4P' 'M5P' 'M6P' 'M9P' 'M1R'
 'M2R' 'M4R' 'M5R' 'M6R' 'M7R' 'M9R' 'M1S' 'M4S' 'M5S' 'M6S' 'M1T' 'M4T'
 'M5T' 'M1V' 'M4V' 'M5V' 'M8V' 'M9V' 'M1W' 'M4W' 'M5W' 'M8W' 'M9W' 'M1X'
 'M4X' 'M5X' 'M8X' 'M4Y' 'M7Y' 'M8Y' 'M8Z']
The dataframe has 11 boroughs and 103 neighborhoods.
75 venues were returned by Foursquare.
There are 280 uniques categories.
```

Table 2: Toronto Borough Names & Neighborhood Postal codes

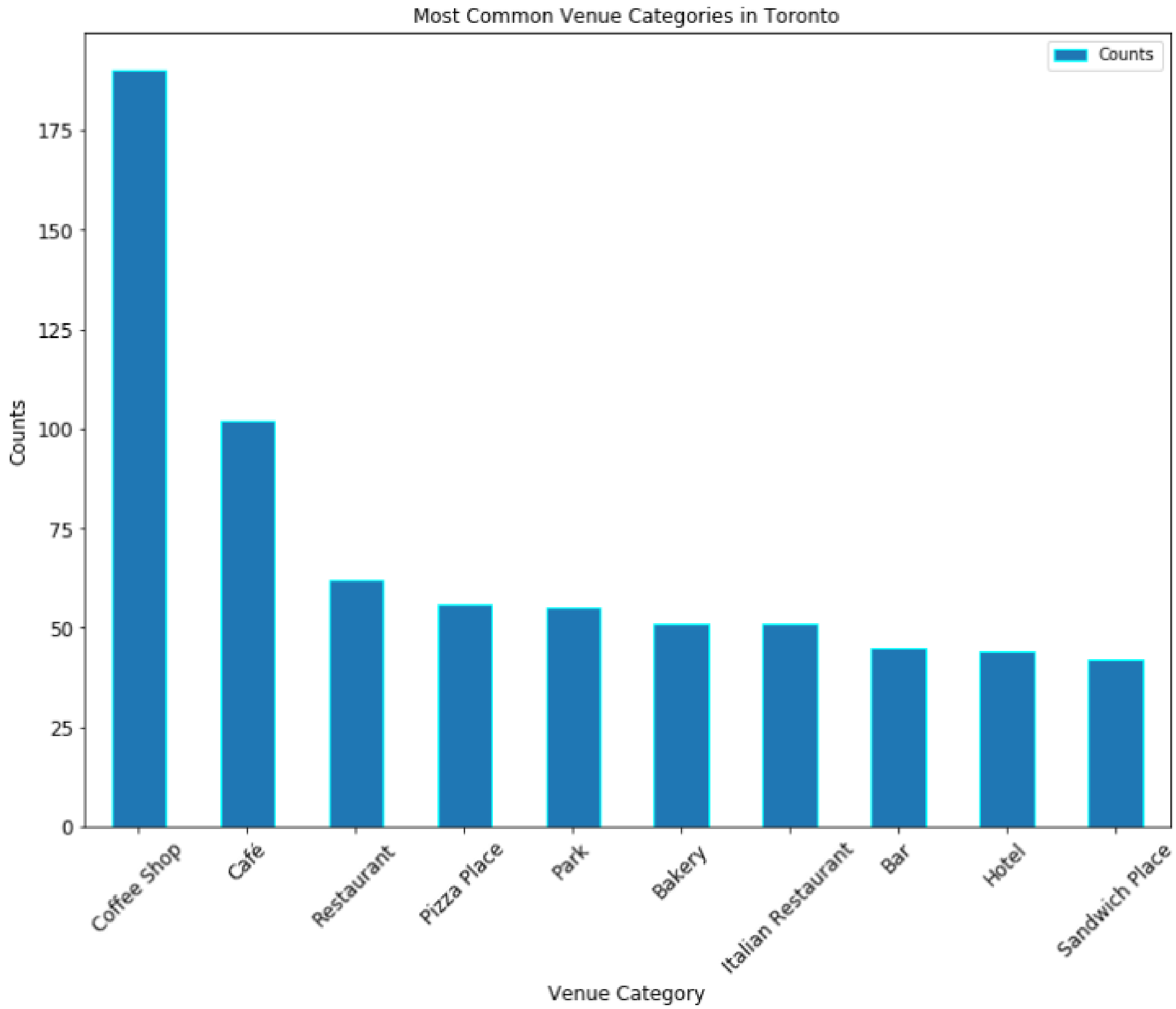


Figure 5: Most Common Venue Categories in the City of Toronto

- **Visual Representation using Word Cloud showing most prominent locations & venues.**

A visual representation of text data using a Python package for generating word clouds. Word cloud (or Tag cloud) is used to display a list of neighborhood & venue categories, the importance of each being shown with both font size and color. The more a specific word appears in the data, the bigger and bolder it appears in the word cloud. This format is useful for quickly perceiving the most prominent locations and venues. Word clouds are commonly used to perform high-level analysis and visualization of text data. Next, I use the stop words to remove "Neighborhood", "Category", "Latitude", "Longitude", "Venue". Word cloud generated (Figure 6) is then superimposed onto an image of Toronto map which I downloaded from Google.

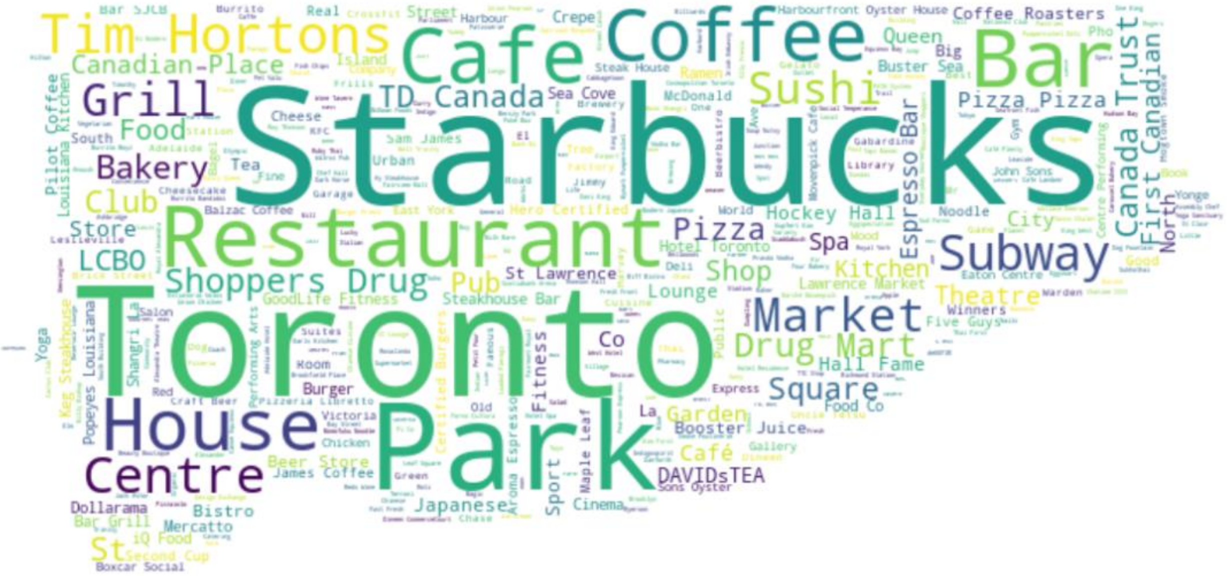
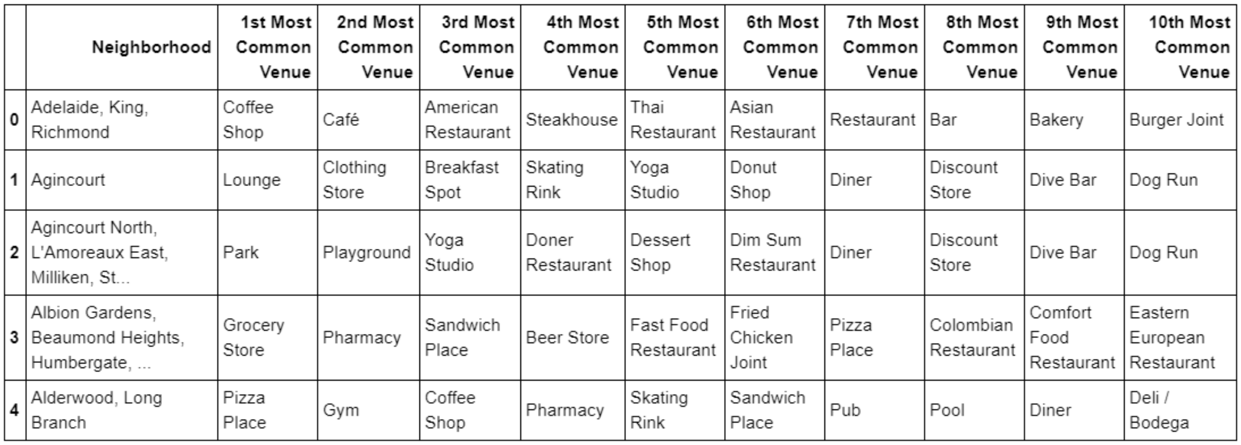


Figure 6: Word Cloud representation of venues in the neighborhood of Toronto

- **Analyze Each Neighborhood**

Dummy variables are created for each category, followed by one hot encoding. The rows are grouped by neighborhood and by taking the mean of the frequency of occurrence of each category.

The original dataFrame is reduced from 2247 to 100. A new dataFrame (Table 3) is created to display the top 10 venues for each neighborhood after sorting venues in descending order.



	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	Adelaide, King, Richmond	Coffee Shop	Café	American Restaurant	Steakhouse	Thai Restaurant	Asian Restaurant	Restaurant	Bar	Bakery	Burger Joint
1	Agincourt	Lounge	Clothing Store	Breakfast Spot	Skating Rink	Yoga Studio	Donut Shop	Diner	Discount Store	Dive Bar	Dog Run
2	Agincourt North, L'Amoreaux East, Milliken, St...	Park	Playground	Yoga Studio	Doner Restaurant	Dessert Shop	Dim Sum Restaurant	Diner	Discount Store	Dive Bar	Dog Run
3	Albion Gardens, Beaumond Heights, Humbergate, ...	Grocery Store	Pharmacy	Sandwich Place	Beer Store	Fast Food Restaurant	Fried Chicken Joint	Pizza Place	Colombian Restaurant	Comfort Food Restaurant	Eastern European Restaurant
4	Alderwood, Long Branch	Pizza Place	Gym	Coffee Shop	Pharmacy	Skating Rink	Sandwich Place	Pub	Pool	Diner	Deli / Bodega

Table 3: Top 10 venues for each neighborhood in the City of Toronto

4.2 Cluster Neighborhoods

This part investigates the way clustering algorithms can be used to identify individual neighborhood properties in the City of Toronto. Clustering algorithms have been used to group the records acquired from different sources. With data related to neighborhood available in specific boroughs in a city, the category of establishments, the frequency of the venue category, and the association between the two can give quality insight into areas within a city or a locality. Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

K-means clustering to segment the neighborhoods into different clusters is the first algorithm I am going to use in this project. Kmeans algorithm is an iterative algorithm that tries to partition the dataset into K-pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation I have within clusters, the more homogeneous (similar) the data points are within the same cluster. The goal of the k-means algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided.

When using k-means clustering, to determine the right number of clusters to be used, it needs to be validated by 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 10 as in the run below), and for each value of k calculate the sum of squared errors (SSE). From the plot of line chart of the SSE for each value of k, which looks like an arm, the "elbow" on the arm is selected as the best value of k.

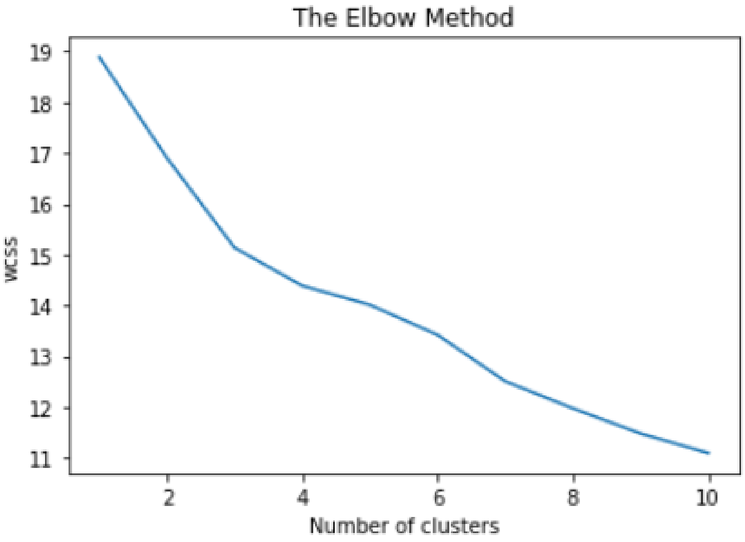


Figure 7: The elbow method to choose the best value of K

The elbow method shows K=3 (Figure 7) as the most suitable value, and hence I run k-means to cluster the neighborhood into 3 clusters.

To visualize the resulting clusters I use Folium map to superimpose color coded clusters on top of it (Figure 8), after entering the same location coordinates and parameters as before in section 4.1 for EDA.

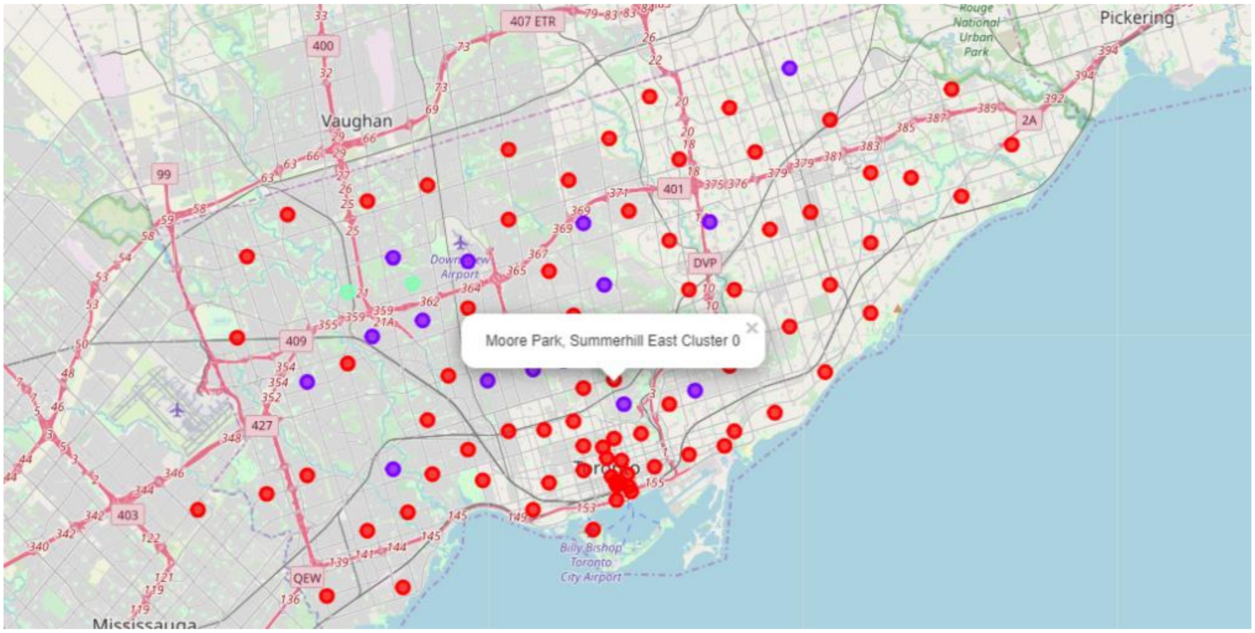

Moore Park, Summerhill East Cluster 0
Vaughan
Pickering
Rouge National Urban Park
Downsview Airport
Toronto
Billy Bishop Toronto City Airport
Mississauga
407 ETR
401
DVP
QEW
427
409
400
403
2A

Figure 8: Color coded cluster map of Toronto after segmenting with K-means clustering

- **Examine Clusters**

To examine each cluster and determine the discriminating venue categories that distinguishes each cluster to name them accordingly.
Cluster 0: UpScale
Cluster 1: MidScale
Cluster 2: DownScale
The clustering algorithm applied to 277 effective establishments for 103 neighborhoods of Toronto using Foursquare data provides support for the conjecture that there exists a few major "families" of neighborhoods, with members exhibiting similar behavior with each other but markedly differing from other segments. I find strong evidence of associations among neighborhoods such as Adelaide, King, Richmond, & west Toronto.
By means of box plot the distinction in number of venue categories in each cluster can fully be understood (Figure 9).

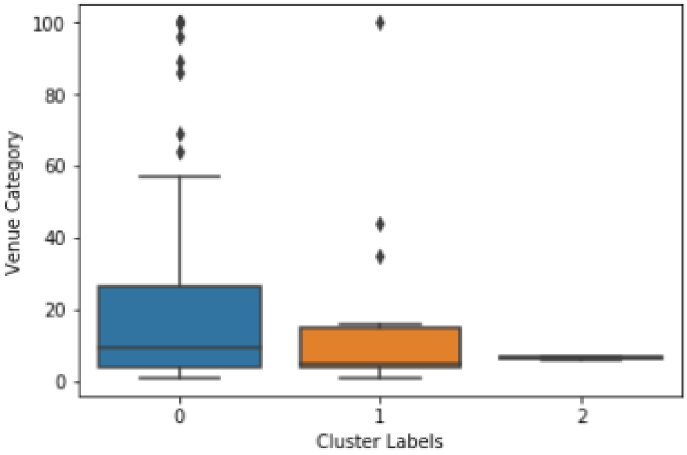
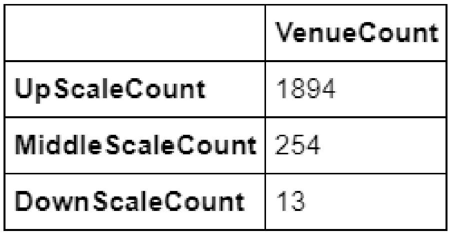


Figure 9: Distribution of venue categories in each cluster.

K-means clustering segmented 2 Neighborhoods in DownScale segment.
K-means clustering segmented 12 Neighborhoods in MidScale segment.
K-means clustering segmented 37 Neighborhoods in UpScale segment.



	VenueCount
UpScaleCount	1894
MiddleScaleCount	254
DownScaleCount	13

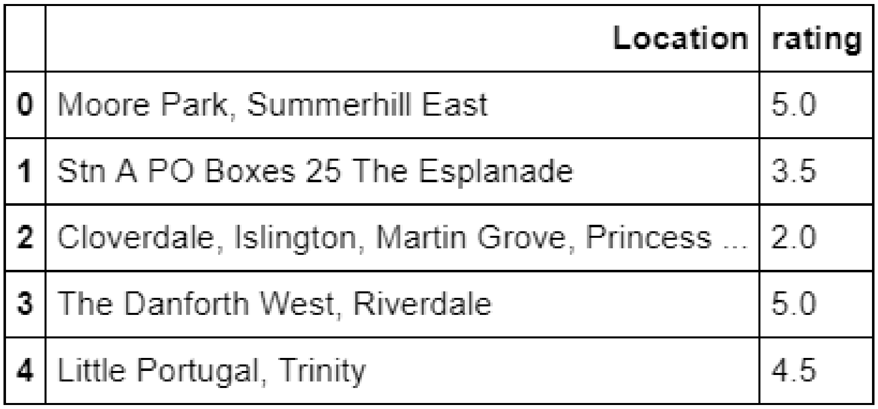
Table 4: Number of Venues in each cluster

This concludes the methodology for investigating the robustness of the clustering algorithm, and to develop a means for testing the significance of neighborhood associations. While the analysis is limited to venue categories and neighborhood data, the results provide a guideline for the further application of cluster analysis to other types of geographical, demographic & socio-economical information.

4.3 Content Based Recommender systems Algorithm for Identifying suitable localities

The recommendation in a content-based system is based on client's liking for a particular neighborhood and the nature of the venue categories contained in that neighborhood. The task of the recommender engine is to recommend ten of the several neighborhoods of Toronto to this client. To achieve this, I have to build the client profile.

I start this by first extracting the category table from the original Toronto dataframe to get the category of every Venue in the neighborhood. From the client's ratings for the neighborhood that he has already experienced with, a vector called "CustomerInput" is created (Table 5).

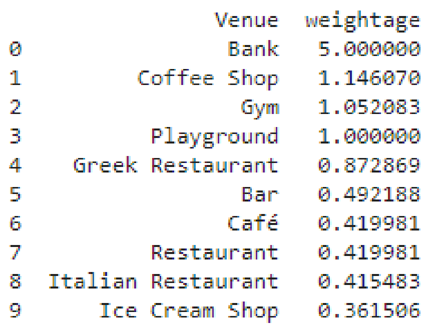


	Location	rating
0	Moore Park, Summerhill East	5.0
1	Stn A PO Boxes 25 The Esplanade	3.5
2	Cloverdale, Islington, Martin Grove, Princess ...	2.0
3	The Danforth West, Riverdale	5.0
4	Little Portugal, Trinity	4.5

Table 5: "CustomerInput" table showing the preference of client for each Neighborhood in Toronto

I have already encoded the "CategoryTable" neighborhoods through the one-hot encoding approach where venue categories of neighborhood were used as the feature set and this is going to be represented as the neighborhood feature set matrix after extracting the customer preferred neighborhood from it. I use the client preferred neighborhood to make the matrix, which represents the neighborhood feature set matrix.

Now I'm ready to start learning the clients's preferences! To do this, I'm going to turn each category into weights. I can do this by using the clients's ratings and multiplying them into the CustomerPreferredCategory table and then summing up the resulting table by column. This operation is actually a dot product between a matrix and a vector, so I can simply accomplish by calling Pandas's "dot" function. If I multiply these two matrices I can get the weighted feature set for the neighborhood. This matrix is also called the weighted category matrix and represents the interests of the client for each categories based on the neighborhood that he has already experienced (Table 6).



```
                Venue  weightage
0                Bank   5.000000
1         Coffee Shop   1.146070
2                 Gym   1.052083
3          Playground   1.000000
4    Greek Restaurant   0.872869
5                 Bar   0.492188
6                Café   0.419981
7          Restaurant   0.419981
8  Italian Restaurant   0.415483
9      Ice Cream Shop   0.361506
```

Table 6: Table showing the category weightage generated by the Recommender system for each venue category from the Client rating information.

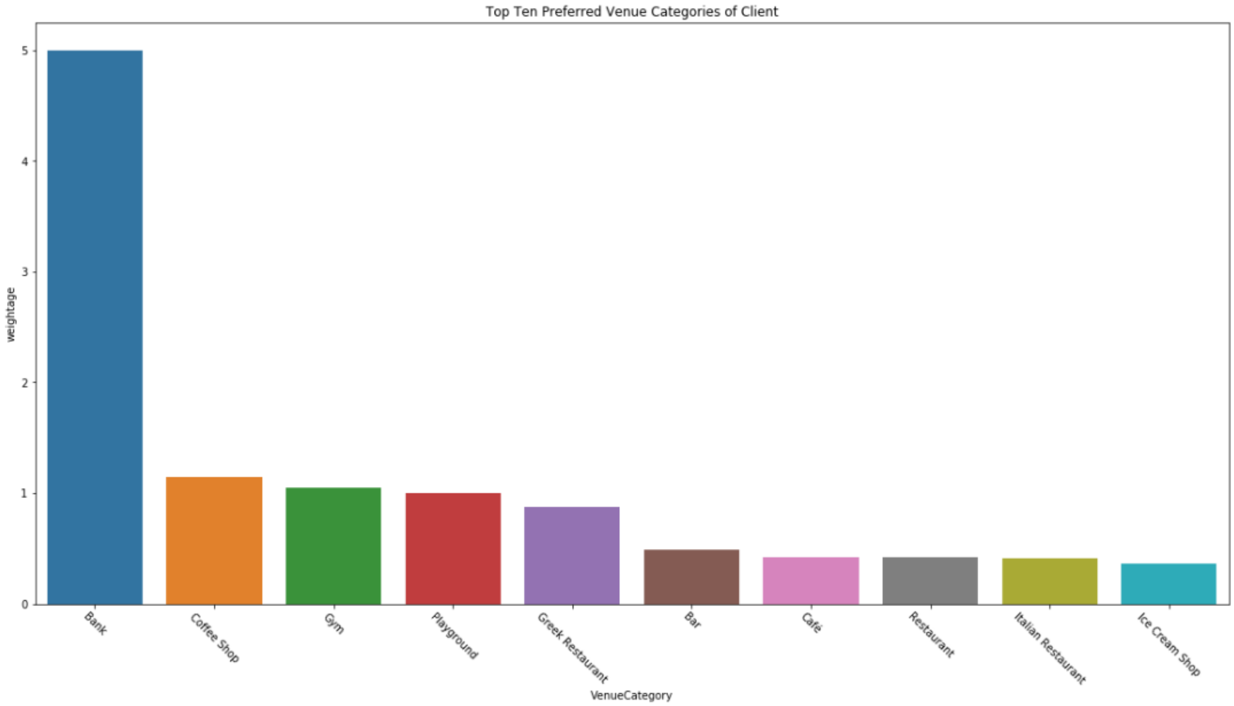
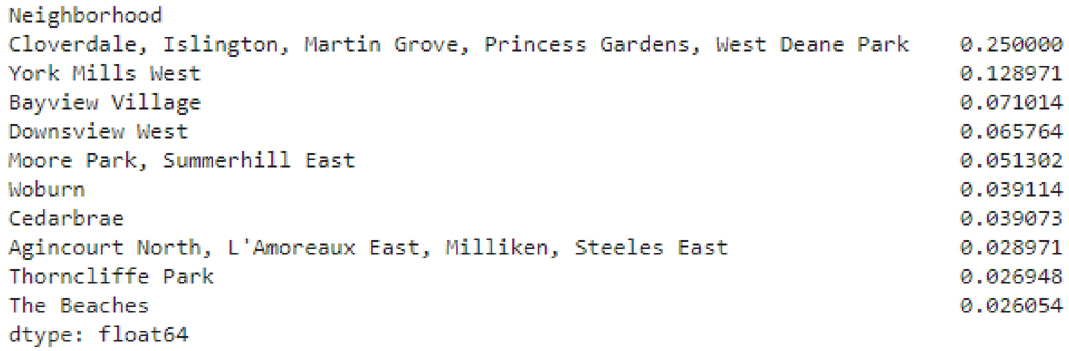


Figure 10: Top Ten Client preferred Venue Categories generated by the Recommender system

Now, given the weighted category matrix, the profile of the client is shaped, aggregated and then normalized to find the “Customer profile”. It clearly indicates that he likes neighborhood with Banks, Coffee Shop, Gym, Playground, Greek Restaurant, Bar, Cafe', Restaurant, Italian Restaurant, Ice Cream Shop etc more than other categories listed (Figure 10).

I use this profile to figure out what neighborhood is proper to recommend to this client. Now, I have the weights for every of the client's preferences. This is known as the Customer Profile. Using this, I can recommend neighborhoods that satisfy the client's preferences. For this I start by extracting the category table from the original dataFrame to encode all other neighborhoods as well. Thus, I am in the position where I have to figure out which of them is most suited to be recommended to the client. With the customer input profile and the complete list of location and their categories in hand, I am going to take the weighted average of every neighborhood based on the input profile and recommend the top ten neighborhoods that most satisfy it.

To do this, I simply multiply the “CustomerProfile” matrix by the “CategoryTable” Matrix and then take the weighted average, which results in the “Recommendation Table” (the weighted neighborhood matrix) given below (Table 7).

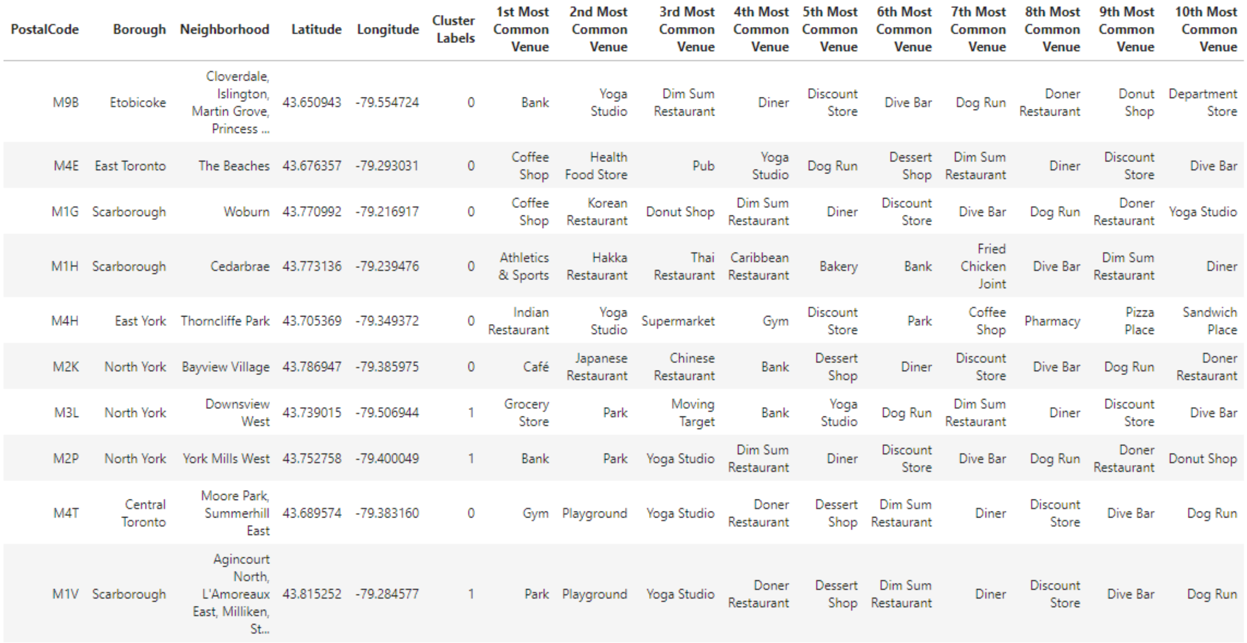


```
Neighborhood
Cloverdale, Islington, Martin Grove, Princess Gardens, West Deane Park    0.250000
York Mills West                                                           0.128971
Bayview Village                                                           0.071014
Downsview West                                                            0.065764
Moore Park, Summerhill East                                               0.051302
Woburn                                                                    0.039114
Cedarbrae                                                                 0.039073
Agincourt North, L'Amoreaux East, Milliken, Steeles East                  0.028971
Thorncliffe Park                                                          0.026948
The Beaches                                                               0.026054
dtype: float64
```

Table 7: The Weighted Neighborhood matrix showing the weightage given to each Neighborhood by the Recommender system based on Client rating.

It shows the weight of each venue category with respect to the Client Profile. Now, if I aggregate these weighted ratings, I get the active client's possible interest level in these neighborhoods. In essence, it's my recommendation lists, which I can sort to rank the neighborhoods and recommend them to the client.

Now here's the final recommendation table(Table 8)!



PostalCode	Borough	Neighborhood	Latitude	Longitude	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 Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
M9B	Etobicoke	Cloverdale, Islington, Martin Grove, Princess ...	43.650943	-79.554724	0	Bank	Yoga Studio	Dim Sum Restaurant	Diner	Discount Store	Dive Bar	Dog Run	Doner Restaurant	Donut Shop	Department Store
M4E	East Toronto	The Beaches	43.676357	-79.293031	0	Coffee Shop	Health Food Store	Pub	Yoga Studio	Dog Run	Dessert Shop	Dim Sum Restaurant	Diner	Discount Store	Dive Bar
M1G	Scarborough	Woburn	43.770992	-79.216917	0	Coffee Shop	Korean Restaurant	Donut Shop	Dim Sum Restaurant	Diner	Discount Store	Dive Bar	Dog Run	Doner Restaurant	Yoga Studio
M1H	Scarborough	Cedarbrae	43.773136	-79.239476	0	Athletics & Sports	Hakka Restaurant	Thai Restaurant	Caribbean Restaurant	Bakery	Bank	Fried Chicken Joint	Dive Bar	Dim Sum Restaurant	Diner
M4H	East York	Thorncliffe Park	43.705369	-79.349372	0	Indian Restaurant	Yoga Studio	Supermarket	Gym	Discount Store	Park	Coffee Shop	Pharmacy	Pizza Place	Sandwich Place
M2K	North York	Bayview Village	43.786947	-79.385975	0	Café	Japanese Restaurant	Chinese Restaurant	Bank	Dessert Shop	Diner	Discount Store	Dive Bar	Dog Run	Doner Restaurant
M3L	North York	Downsview West	43.739015	-79.506944	1	Grocery Store	Park	Moving Target	Bank	Yoga Studio	Dog Run	Dim Sum Restaurant	Diner	Discount Store	Dive Bar
M2P	North York	York Mills West	43.752758	-79.400049	1	Bank	Park	Yoga Studio	Dim Sum Restaurant	Diner	Discount Store	Dive Bar	Dog Run	Doner Restaurant	Donut Shop
M4T	Central Toronto	Moore Park, Summerhill East	43.689574	-79.383160	0	Gym	Playground	Yoga Studio	Doner Restaurant	Dessert Shop	Dim Sum Restaurant	Diner	Discount Store	Dive Bar	Dog Run
M1V	Scarborough	Agincourt North, L'Amoreaux East, Milliken, St...	43.815252	-79.284577	1	Park	Playground	Yoga Studio	Doner Restaurant	Dessert Shop	Dim Sum Restaurant	Diner	Discount Store	Dive Bar	Dog Run

Table 8: The recommendation Table which ranks the neighborhoods that clients may like the most showing client's possible interest level.

Seven out of ten recommended neighborhood fall into Up scale neighborhood with three in the Middle Scale market (Table 8) and the recommended locations are displayed in a simple mplleaflet map below (Figure 11):

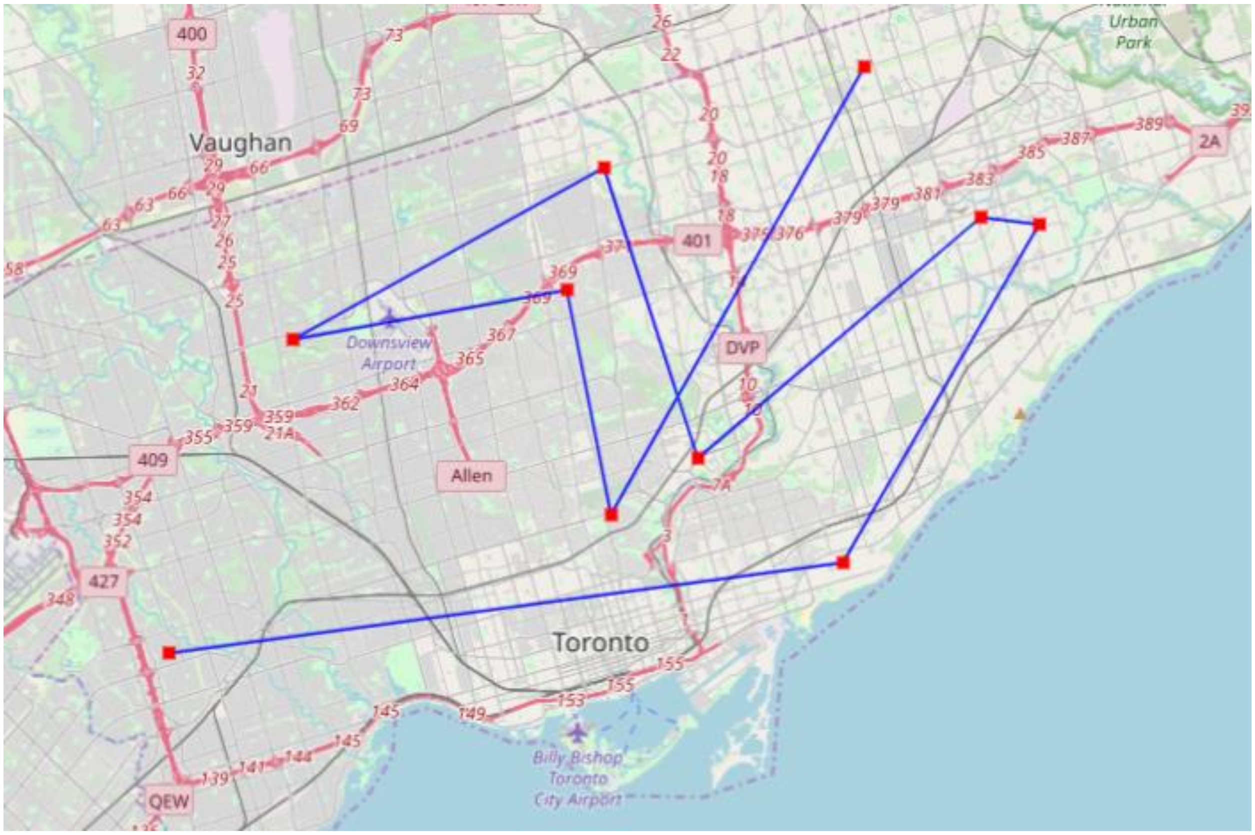


Figure 11: Simple mplleaflet map showing areas which have greater chances of being preferred starting from the south end of the map and moving up along the line.

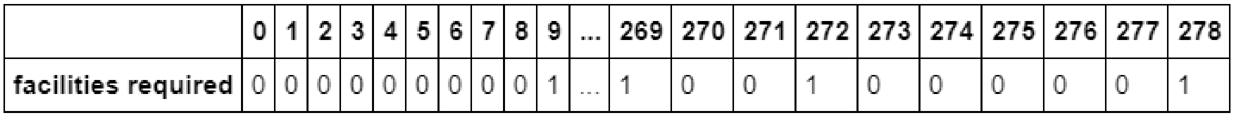
So the best match for the Client's profile would be a neighborhood located at Etobicoke borough, which is at the South end of Toronto, followed by East Toronto, Scarborough, East York, North York etc. I can say that the Cloverdale, Islington, Martin Grove, Princess Gardens, West Deane Park belonging to Up scale cluster has the highest score in my list and it's proper to recommend these to the client.

4.4 Classification of Client Profile

Multiple models using Multiclass classifier can be built to identify the combination of categories leading to neighborhood’s cluster category outcome and applied at Client profile to classify the client profile according to his points of interests. For the case study, K-nearest neighbors, Decision Tree classifier with entropy criterion, Decision Tree classifier with gini criterion, Extra Tree classifier with gini criterion, Support Vector Machine using SVM OVR, SVM_OneVsOne, SVM_Outputcode,& RadiusNeighbors Classifier are used. From this information, the analysts can obtain the best neighborhood cluster, or the likelihood of preference of segments of neighborhood for each client. These classification models are easy for non-data scientists to understand and apply, to score new clients for their points of interests. Consultants & advisors can readily see what facilities are making a client getting attracted to a particular locality. This gives a complete picture of the customer's interest and how it is evolving with the change of preference being applied and with the addition of new facilities to a particular neighborhood. This predictive modelling machine learning recipe for this classification project uses different Classification models available in scikit-learn as well as their tuning the hyper-parameters of these models. Finally, the trained & tuned models need to be finalized for making predictions on unexperienced data and the outcomes of the model is evaluated & presented.

- **Dataset and Features**

To start with the classification , I need to convert the “CustomerProfile” weights from section 4.2, into dummy variables with weights > 0.1 as 1 and < 0.1 as 0 and transposed so as to create the required facilities for the client in neighborhood search. This dataset is treated as the test set for the classification Algorithm to classify the client based on neighborhood cluster preference and so as to find the cluster the client is attracted to.



	0	1	2	3	4	5	6	7	8	9	...	269	270	271	272	273	274	275	276	277	278
facilities required	0	0	0	0	0	0	0	0	0	1	...	1	0	0	1	0	0	0	0	0	1

Table 9: Dummy variable table of category weightage table with “ones” represents the required facilities in a neighborhood search.

In short, the data from K-means clustering is to be taken as training set and those from Recommender system will be used as Test set to predict the possible outcome of Client preferred cluster group.

The classification focuses on using various venue categories to predict the cluster which the client’s profile matches with. The target field, called cluster labels, has 3 possible values that correspond to the three cluster groups obtained from K-means cluster algorithm and are named as follows: 0 – UpScale, 1- MidScale, 2- DownScale

The test dataset X_test is the "ClientProfile" dataset where the output values are to be predicted from the trained and tested classification table.

To train and test my models I choose "TorontoRename" as the dataset & that is going to be used to prepare the classification models, with the "cluster label" column as the target column (dependent variable) and the rest of the columns as feature columns (independent variables / predictor columns).

Train/Test Split

To make correct predictions on unknown data, it is important that my models have a high, out-of-sample accuracy. Train/Test Split method is used to achieve that and it involves splitting the dataset into training and testing sets respectively, which are mutually exclusive. After which, the model is trained with the training set and test with the testing set. Because the testing dataset is not part of the dataset that have been used to train the data, it is more realistic for real world problems.

The purpose of train_test_split method is to randomly breaks up the data. By specifying random_state=4, I will always get the same output for the same input. To avoid introducing a bias in test using train-data, the train-test split should be performed before the classification algorithm.

test_size=0.2 indicates that the model is going to split randomly this data set into 80% trainset and 20% testset. The model is first trained by X_trainset and y_trainset. I gather predictions from the trained model on the inputs from the test dataset X_testset and compare them to the withheld output values of the testset and the result gives the class of the unknown value.

Choice of classifiers

Since our dataset consists of 3 cluster labels, "Up scale", "Middle scale" & "Down scale", I have to choose the classifiers accordingly from those suitable for multiclassification. K-Nearest Neighbors (KNN), Decision Tree with both entropy and gini criteria, Extra Tree with gini criterion, Support Vector Machine using SVM OVR, SVM_OneVsOne, SVM_Outputcode,& RadiusNeighbors Classifier are found to be appropriate to be used as Multiclassifiers.

Accuracy evaluation

Accuracy classification score is a function that computes subset accuracy using the available metrics from sklearn. This function is equal to the jaccard_similarity_score function. Essentially, it calculates how closely the actual labels and predicted labels are matched in the test set. Also, different metrics to evaluate the model accuracy can be run, for example, using a confusion matrix to show the results.

To predict the Client profile cluster

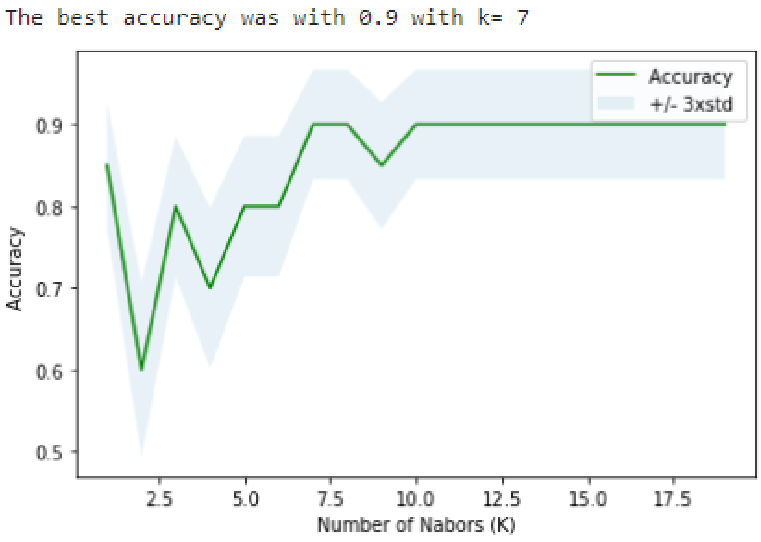
All the models are then ready to test the Client profile dataset to predict the cluster group of the Client profile.

The following Classifiers are employed to predict the best cluster for the client

1. **K-Nearest Neighbor Classifier**

To choose the best value of K

The model is run for different values of k starting from K=1, using the training part for modeling, and calculate the accuracy of prediction using all samples in the test set. Repeat the process, increasing the k, and see which k is the best for the model by Plotting model accuracy for Different number of Neighbors & choose the right value for K (Figure 12).



```
The best accuracy was with 0.9 with k= 7
```

Figure 12: Accuracy plot of K-Nearest Neighbor Classifier for each value of K

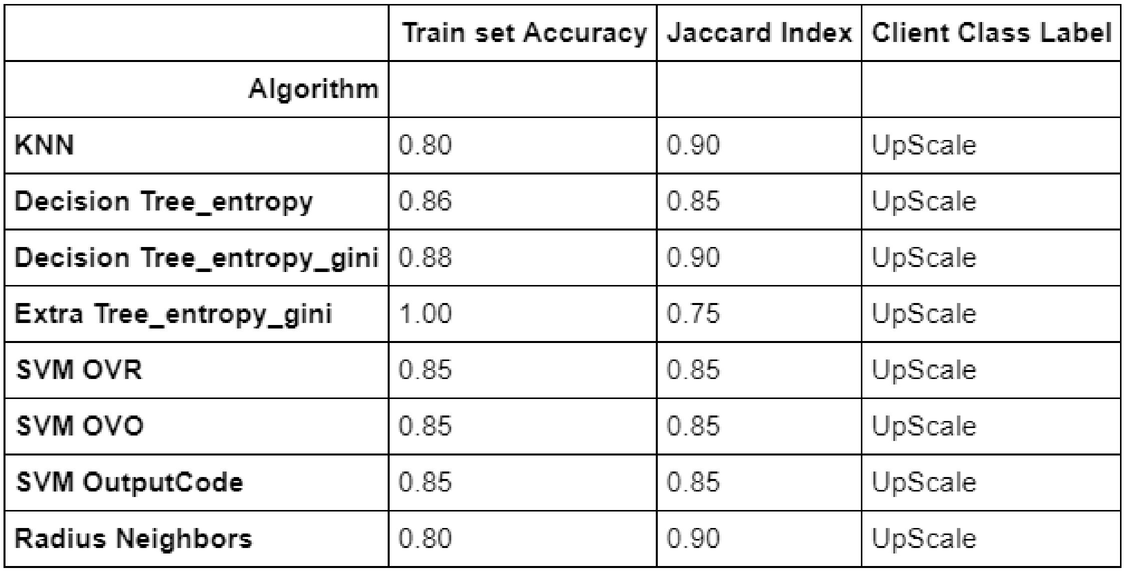
2. **Decision Tree classifier with entropy criterion**

Predictiveness is based on decrease in entropy - gain in information, or impurity.

3. **Extra Tree classifier with gini criterion**
4. **Support Vector Machine - One versus Rest (SVM OvR) Multiclass classifier**
5. **SVM_OneVsOne Multiclass classifier**
6. **SVM_Outputcode Multiclass classifier**
7. **RadiusNeighbors Classifier**

- **The Final Verdict**

The table of classification results along with the class to which the Client profile is identified is reported in the following table (Table 10).



	Train set Accuracy	Jaccard Index	Client Class Label
Algorithm			
KNN	0.80	0.90	UpScale
Decision Tree_entropy	0.86	0.85	UpScale
Decision Tree_entropy_gini	0.88	0.90	UpScale
Extra Tree_entropy_gini	1.00	0.75	UpScale
SVM OVR	0.85	0.85	UpScale
SVM OVO	0.85	0.85	UpScale
SVM OutputCode	0.85	0.85	UpScale
Radius Neighbors	0.80	0.90	UpScale

Table 10: Trainset accuracy, Jaccard Index, & Client cluster class predicted

All models predicted “Upscale” neighborhood for the client, in accordance with the recommendations of Recommender system.

4.5 Visual Representation using Decision Tree

Let me explore the decision tree machine learning algorithm a little bit more, and see if it can be used as the right technique to automate the process of identifying the category of a given neighborhood while simultaneously providing us with some insight on why a given category is believed to belong to a certain type of neighborhood.

To evaluate the model of Neighborhoods, I will split the dataset into a “trainset” and a “testset”. I will build the decision tree using the “trainset”. Then, I will test the model on the “testset” and compare the neighborhood that the model predicts to the actual neighborhood.

Here, I am creating a decision tree for the categories for just “UpScale” & “MidScale” clusters. The reason for this is because the decision tree does not run well when the data is biased towards these clusters. One option is to exclude the “Downscale” cluster from our analysis or just build decision trees for different subsets of the data. Let me go with the latter solution and build the decision tree using the data pertaining to the “UpScale” & “MidScale” clusters and name our decision tree “cluster_tree”.

I experimented with different numbers of decision trees, with each time getting different results. Special attention is made to prevent overfitting. Testing both “gini” and “entropy” criteria, I found entropy criteria the best working. That measures the quality of the splits at each node of the tree in such a way that after the split, the more homogeneous are the groups of neighbors, the more the entropy will be reduced and hence the disorder. The more homogeneous a node is compared to the parent node the lower the entropy is compared to the entropy of the parent node.

And the information gain is how much the entropy is managed to get reduced from before to after the split because it's the difference between the entropy of the parent node minus the entropy of the child node. And so for this I will choose an entropy criterion so that the tree makes it predictions about the cluster group of each neighborhood according to the entropy criterion (Figure 13).

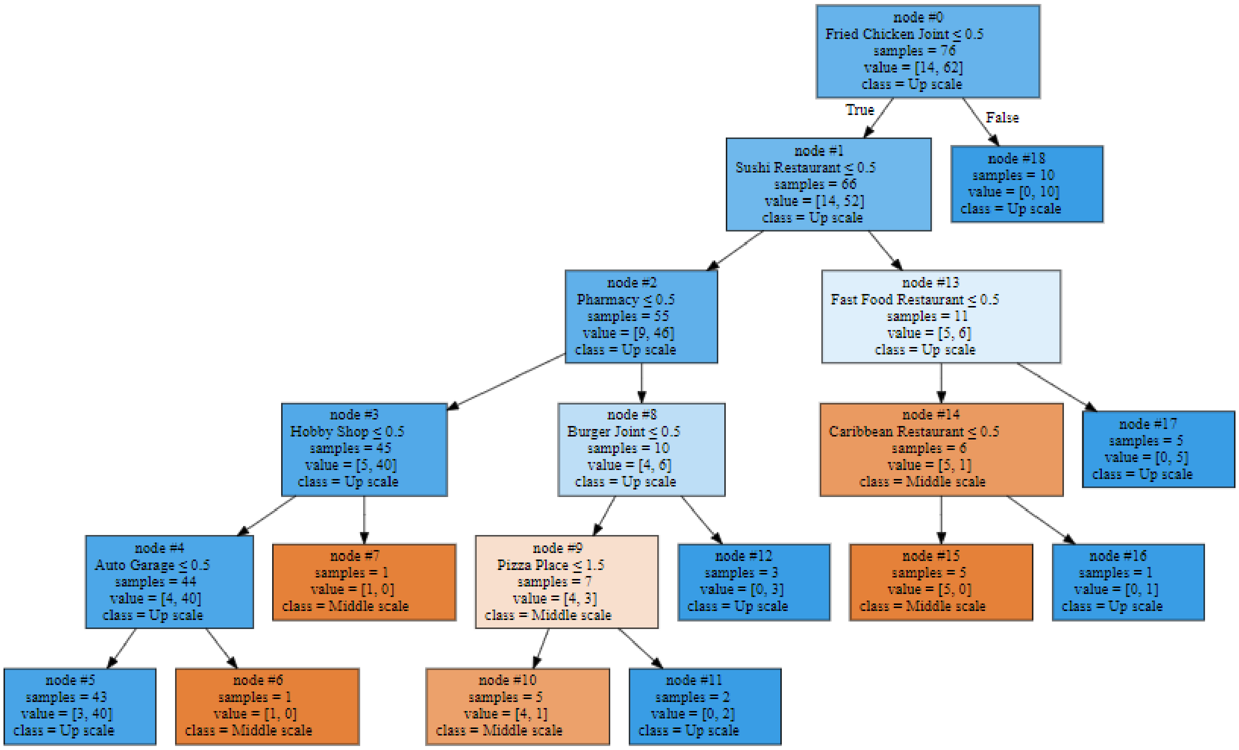


Figure 13: Decision Tree using entropy criterion and maximum depth of 5

Confusion Matrix

A good thing about the confusion matrix is that it shows the model’s ability to correctly predict or separate the classes. In the specific case of a binary classifier, such as this example, I can interpret these numbers as the count of true positives, false positives, true negatives, and false negatives (Figure 14).

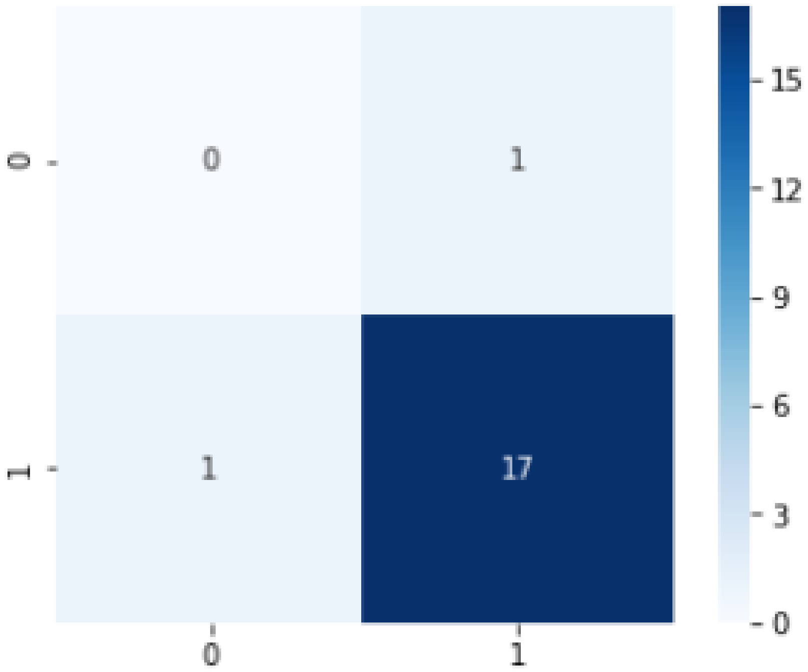


Figure 14: Heat Map of Confusion matrix to show true positives, false positives, true negatives, and false negatives.

4.6 Discussion:

With k-means clustering the neighborhoods are segregated into 3 main clusters. Applying content based Recommender system on those clusters and client's ratings on already experienced neighborhoods, I have successfully created a model to recommend a neighborhood that suit the client's taste. The categories of venues in the preferred neighborhood are given a weightage and are converted to dummy variable dataset to be used in various classification algorithms. All the classifiers I experimented on are in good agreement with very high accuracy, in predicting the class "UpScale" for this client profile. Now, I can recommend my client to focus on continuing search for neighborhood in this cluster so as to enjoy all the preferred facilities available. Finally, the decision tree can be of assistance in explaining ideas to someone not so familiar with any kind of technology. The model can be tweaked easily to accommodate any changes in the neighborhood facilities or in the client's preferences.

5. Conclusion

The main stages consisted of data collection, visualization, pre-processing, cluster with k-means algorithm, weighting term, application of Recommender system & finally classification with KNN, Decision tree, Support Vector Machine Algorithms.

In short, combination of Clustering, Recommender System Algorithm & Classification techniques to segregate, recommend, classify & predict the available Toronto Neighborhoods to potential clients based on their taste are employed sequentially to get the best possible outcome of their neighborhood search.

This project provides an excellent case study for applying machine learning algorithms to large data sets lacking obvious structure. This work shall be of great importance not only for large scale tourism industry like Expedia, Trip advisor, but also for small and medium sized travel agents, real estate agents, marketers, Government administrations, researchers, students and academicians. This project can be extended further to include the budget, demographic factors, socioeconomic status, various environmental parameters etc.

Finally, let me leave the topic with a comprehensive & comprehensible visual representation of Top Neighborhoods in Toronto.

Leaside Leaside
Processing Centre
Wellesley Church
Station Harbourfront
Portugal Trinity
First Canadian
East
Toronto
Park Kensington
Little Portugal
Victoria Hotel
Park Berczy
Don Mills
Annex North
India Bazaar
city First
Toronto Dominion
Grange Park
King Richmond
Exchange Toronto
Court Victoria
Dominion Centre
Kensington Market
Richmond Adelaide
Chinatown Grange
North Toronto
Junction South
St James
West India
Design Exchange
Park Junction
Esplanade Stn
Cabbagetown St
Victoria
Commerce Court
Church Wellesley
Town St
District Ryerson
Berczy Park
Toronto Harbord
PO Boxes
Beaches West
Regent Park
South Willowdale
Lawrence Manor
Oriole Fairview
Stn PO
High Park
West North
Harbourfront Regent
Danforth West
Ryerson Garden
Adelaide King
Bedford Park
West Riverdale
East Bedford
Islands Union
Garden District
Yorkville Annex
Union Station
Queen Park
Park Harbourfront
Henry Farm
Fairview Henry
Market Chinatown
Canadian Place
Mills South
James
Town
Underground city
Centre Design
Way Street
Central Way
Place Underground
Hotel Commerce
Willowdale South
Place Parkdale
Forest Hill
Harbourfront East
Park Don
Toronto Islands
Farm Oriole
Town Cabbagetown
Bazaar Beaches
Exhibition Place
Street Central
Runnymede Swansea
Trinity Little