Report - Week 2 Part 1

01.07.2021

1 Final Report – The Battle of Neighborhoods

1.1 Finding a Better Place in Downtown Toronto, Toronto

1.1.1 1) Introduction:

The goal of this project is to assist people in discovering better amenities in their area. It will assist individuals in making informed and efficient decisions on which great neighbourhood to choose from among the many in Downtown Toronto, Toronto.

Many individuals are travelling to different parts of Canada, and they need to do a lot of research to find affordable housing and a place which is closer to their workplace. This project is for folks who want to live in a more desirable neighbourhood and near to IT companies. For easy access to cafes, schools, supermarkets, medical and food stores, malls, theatres, hospitals, and like-minded people, among other things.

This project aims to produce a feature analysis for people migrating to Downtown Toronto in order to find the best area by comparing communities. The features include median housing price and proximity to a nearby workplace based on ratings, local crime rates, road connectivity, weather conditions, emergency management, water resources (both fresh and waste water), sewage sys- tems, and recreational amenities.

It will assist people in becoming familiar with the area and community before relocating to a new city, state, nation, or location for job or to begin a new life.

1.1.2 2) Data and API:

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

We're going to use the Downtown Toronto dataset, which we scraped from Wikipedia in Week3. Latitude and longitude, as well as zip codes, make up this dataset.

We'll require information on various venues throughout that borough's various neighbourhoods. We'll use "Foursquare" locational information to get that information. Foursquare is a location data provider that provides information about a variety of locations and activities in a given area. Names of venues, their locations, menus, and even images are examples of this type of informa- tion. As a result, the foursquare location platform will serve as the single data source, as the API provides access to all of the essential data.

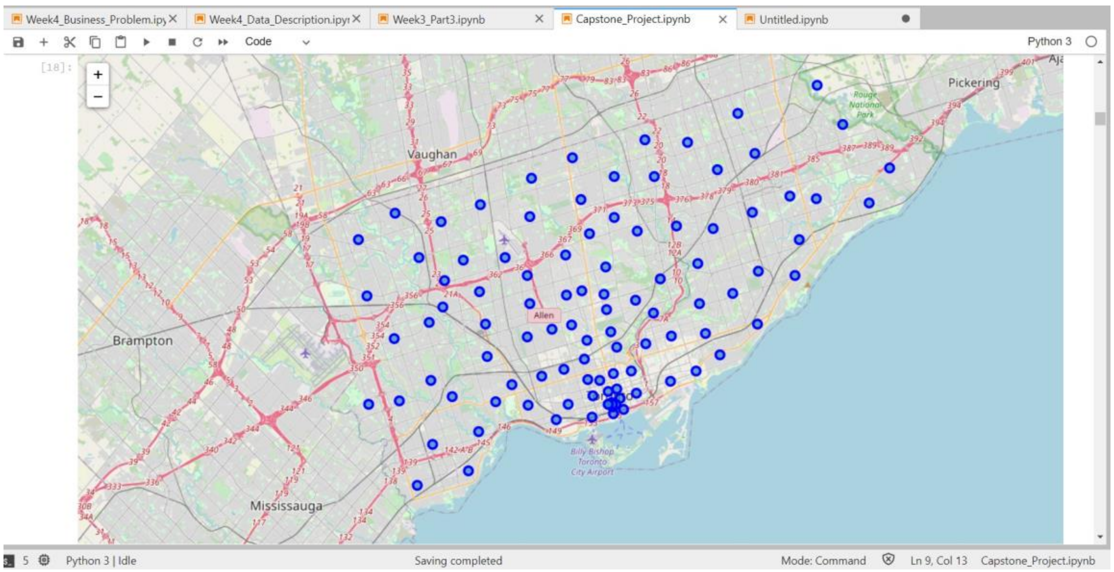
We next connect to the Foursquare API to obtain information about venues inside each neighbour- hood after finding the list of neighbourhoods. The radius for each neighbourhood has been set at

100 metres.

The Foursquare data comprised information on venues within a certain distance of the postcodes' longitude and latitude. The following is the information received for each venue:

1. Name of the venue e.g. the name of a store or restaurant
2.Neighborhood
3.Neighborhood Latitude
4.Neighborhood Longitude 5.Venue
6.Venue Category 7.Venue
Latitude 8.Venue
Longitude

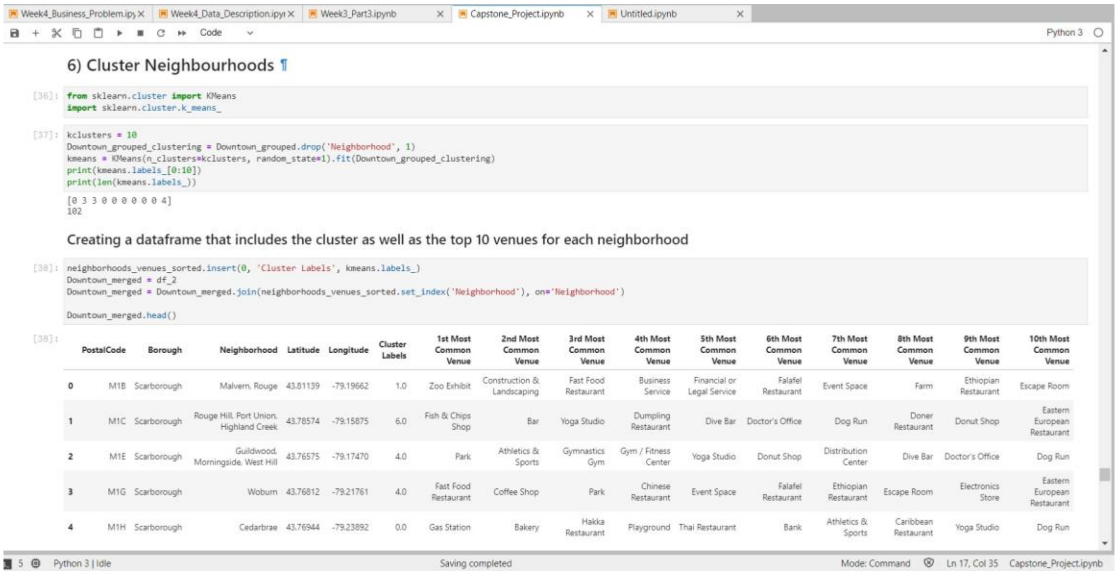
1.1.3 Map of Downtown Toronto



1.1.4 3) Methodology

Clustering approach To examine the similarities of two cities, we opted to look for similar neighbourhoods in big cities like New York and Toronto by exploring neighbourhoods, segmenting them, and grouping them into clusters. To do so, we must cluster data using the k-means cluster- ing algorithm, which is a type of unsupervised machine learning.

1.2 K-Means Clustering



Week4_Business_Problem.ipy | Week4_Data_Description.ipyı | Week3_Part3.ipynb | Capstone_Project.ipynb | Untitled.ipynb

Code | Python 3

6) Cluster Neighbourhoods

[36]:
```
from sklearn.cluster import KMeans
import sklearn.cluster.k_means_
```

[37]:
```
kclusters = 10
Downtown_grouped_clustering = Downtown_grouped.drop('Neighborhood', 1)
kmeans = KMeans(n_clusters=kclusters, random_state=1).fit(Downtown_grouped_clustering)
print(kmeans.labels_[0:10])
print(len(kmeans.labels_))
```

```
[0 3 3 0 0 0 0 0 0 4]
102
```

Creating a dataframe that includes the cluster as well as the top 10 venues for each neighborhood

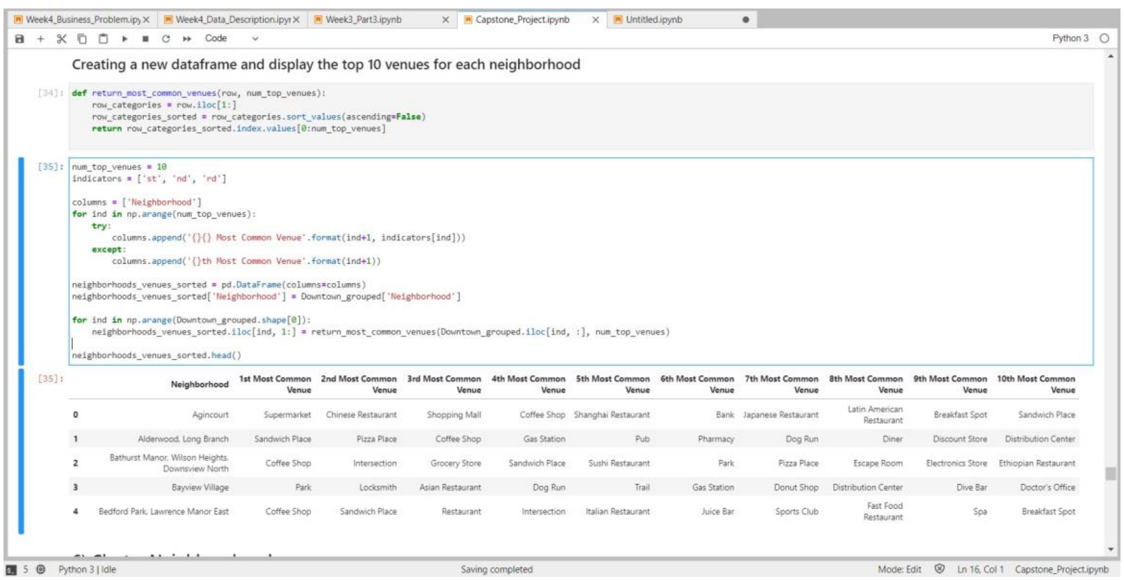
[38]:
```
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
Downtown_merged = df_2
Downtown_merged = Downtown_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

Downtown_merged.head()
```

[38]:

	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
0	M1B	Scarborough	Malvern, Rouge	43.81139	-79.19662	1.0	Zoo Exhibit	Construction & Landscaping	Fast Food Restaurant	Business Service	Financial or Legal Service	Falafel Restaurant	Event Space	Farm	Ethiopian Restaurant	Escape Room
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek	43.78574	-79.15875	6.0	Fish & Chips Shop	Bar	Yoga Studio	Dumpling Restaurant	Dive Bar	Doctor's Office	Dog Run	Doner Restaurant	Donut Shop	Eastern European Restaurant
2	M1E	Scarborough	Guildwood, Morningside, West Hill	43.76575	-79.17470	4.0	Park	Athletics & Sports	Gymnastics Gym	Gym / Fitness Center	Yoga Studio	Donut Shop	Distribution Center	Dive Bar	Doctor's Office	Dog Run
3	M1G	Scarborough	Woburn	43.76812	-79.21761	4.0	Fast Food Restaurant	Coffee Shop	Park	Chinese Restaurant	Event Space	Falafel Restaurant	Ethiopian Restaurant	Escape Room	Electronics Store	Eastern European Restaurant
4	M1H	Scarborough	Cedarbrae	43.76944	-79.23892	0.0	Gas Station	Bakery	Hakka Restaurant	Playground	Thai Restaurant	Bank	Athletics & Sports	Caribbean Restaurant	Yoga Studio	Dog Run

5 Python 3 | Idle | Saving completed | Mode: Command | Ln 17, Col 35 | Capstone_Project.ipynb

1.3 Most Common venues near Neighborhood



Week4_Business_Problem.ipy | Week4_Data_Description.ipyı | Week3_Part3.ipynb | Capstone_Project.ipynb | Untitled.ipynb

Code | Python 3

Creating a new dataframe and display the top 10 venues for each neighborhood

[34]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)
    return row_categories_sorted.index.values[0:num_top_venues]
```

[35]:
```
num_top_venues = 10
indicators = ['st', 'nd', 'rd']

columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = Downtown_grouped['Neighborhood']

for ind in np.arange(Downtown_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(Downtown_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head()
```

[35]:

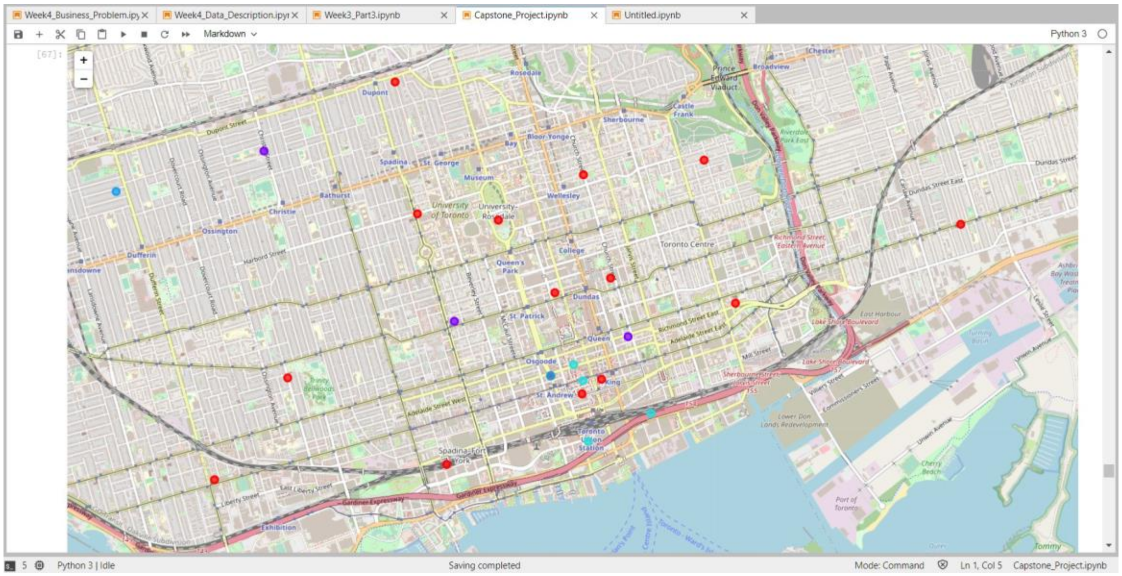
	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Agincourt	Supermarket	Chinese Restaurant	Shopping Mall	Coffee Shop	Shanghai Restaurant	Bank	Japanese Restaurant	Latin American Restaurant	Breakfast Spot	Sandwich Place
1	Alderwood, Long Branch	Sandwich Place	Pizza Place	Coffee Shop	Gas Station	Pub	Pharmacy	Dog Run	Diner	Discount Store	Distribution Center
2	Bathurst Manor, Wilson Heights, Downsview North	Coffee Shop	Intersection	Grocery Store	Sandwich Place	Sushi Restaurant	Park	Pizza Place	Escape Room	Electronics Store	Ethiopian Restaurant
3	Bayview Village	Park	Locksmith	Asian Restaurant	Dog Run	Trail	Gas Station	Donut Shop	Distribution Center	Dive Bar	Doctor's Office
4	Bedford Park, Lawrence Manor East	Coffee Shop	Sandwich Place	Restaurant	Intersection	Italian Restaurant	Juice Bar	Sports Club	Fast Food Restaurant	Spa	Breakfast Spot

5 Python 3 | Idle | Saving completed | Mode: Edit | Ln 16, Col 1 | Capstone_Project.ipynb

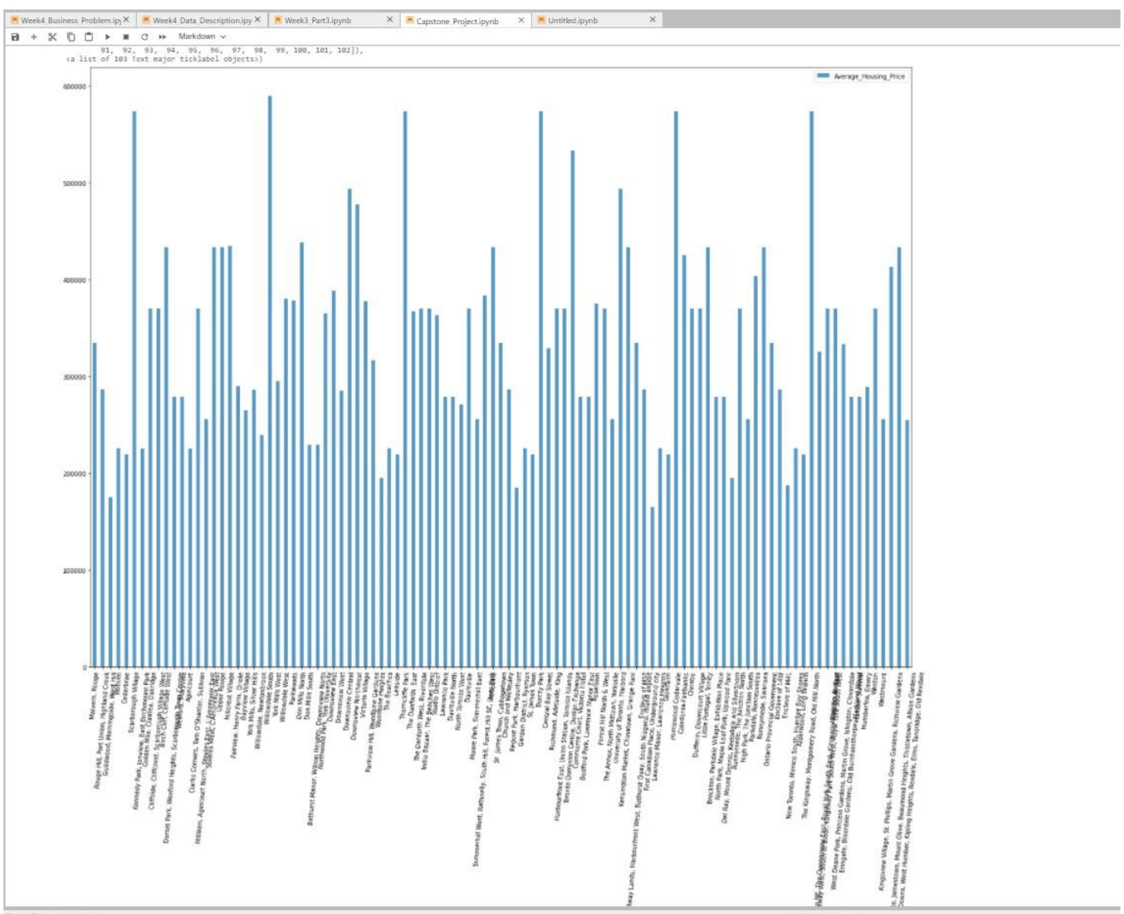
Workflow The attributes of nearby sites in the neighbourhoods would be mined using Foursquare API credentials. Due to http request constraints, the radius parameter should be ad- justed to 500 and the number of places per neighbourhood parameter should be set to 100.

1.3.1 4) Results

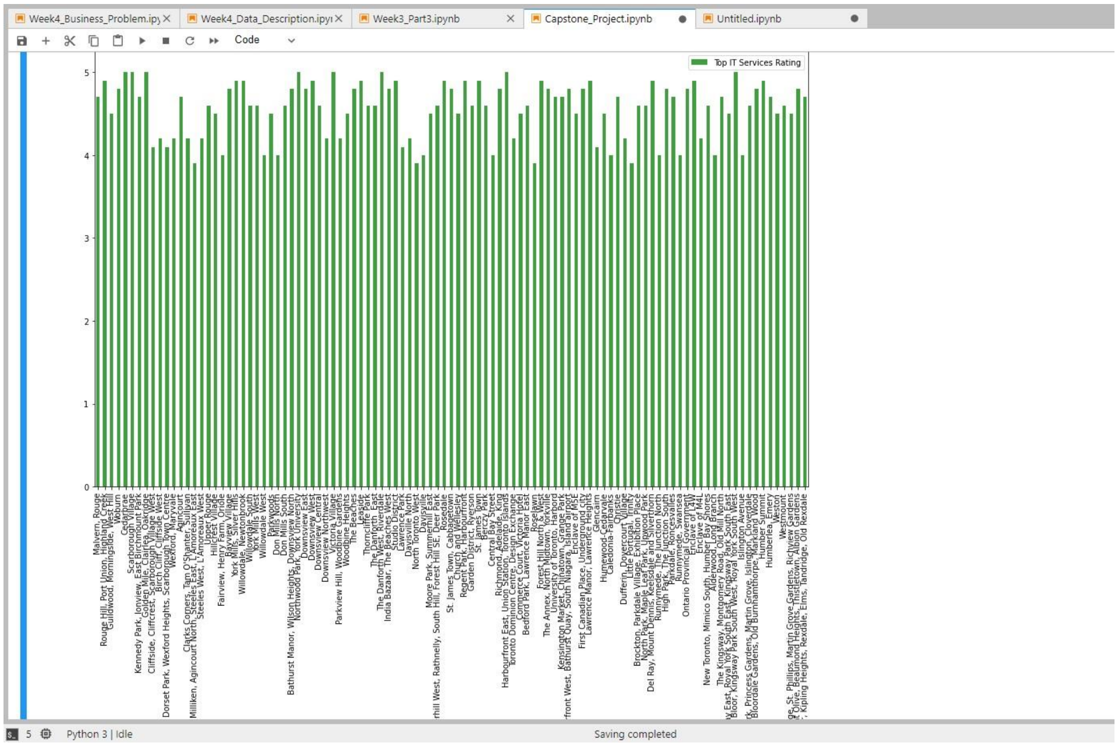
1.4 Map of clusters in Downtown Toronto



1.5 Average Housing Price by Clusters in Downtown Toronto



1.6 IT Companies Ratings by Clusters in Downtown Toronto



Location Downtown Toronto is the main central business district of Toronto, Ontario, Canada. It is also the home of the municipal government of Toronto and the Government of Ontario. The area is made up of Canada's largest concentration of skyscrapers and businesses that form Toronto's skyline.

Foursquare API This project would use the Four-square API as its primary data source because it contains a database of millions of locations, particularly their places API, which allows users to search for, share, and learn about businesses.

1.6.1 5) Discussion

The primary goal of this project is to recommend a better neighbourhood in a new city to those who are relocating there. Access to the airport, bus station, town centre, markets, and other neces- sities in the area.

1. Sorted list of houses in ascending or descending order by housing prices.
2. A list of IT Services that has been sorted by location, fees, rating, and reviews.

1.6.2 6) Conclusion

In this project, I used the k-means cluster algorithm to divide the neighbourhood into 10 differ- ent clusters and for 103 different latitude and longitude from a dataset with fairly comparable neighbourhoods. The above charts were used to deliver findings to a specific community based on typical house prices and school ratings.

Future Works: This project can be continued to improve the accuracy of finding the best property in Downtown Toronto. Best refers to the most efficient use of all available resources, as well as the most cost-effective use of those resources.

The following libraries were used to develop the project:

- Requests: Library to handle HTTP requests.
- Beautiful Soup: Library to parse HTML and XML documents.
- JSON: Library to handle JSON files.
- XML: To separate data from presentation and XML stores data in plain text format.
- Pandas: For creating and manipulating dataframes.
- Geocoder: To retrieve Location Data.
- Matplotlib: Python Plotting Module.
- Folium: Python visualization library would be used to visualize the neighborhoods cluster distribution of using interactive leaflet map.
- Scikit Learn: For importing k-means clustering.

[ ]:

```
```