

MARKET SEGMENTATION

PYTHON

TEAM AWESOME

Where to open a cafe in Toronto?



Toronto boasts a diverse array of neighborhoods, each with its own unique charm and characteristics.

The goal was to identify a neighborhood that provides optimal business opportunities, a robust customer base, and minimal competition.

OBJECTIVE

The objective was to analyze the distribution of markets, assess the existing competition, evaluate the population density and median household income within each neighborhood.

How can I find reliable sources for population and income data?

Which API service best meets the needs?

Which statistical analysis method is best suited for analyzing categorical data?

PROJECT OVERVIEW

Data sources

3
weighed & scrutinized

Wikipedia
Statistics Canada
Google Nearby Places API

Processes

5,000,000+
Lines of data processed

Web scraped
Read, wrote csv
Cleaned, Transformed
API calls
Data models pandas
One Hot encoding
K Means
Plots, Interpretation

Effort

80+
hours of learning

Reading HTML
Evaluating and confirming data
Writing more functions
modeling data
one hot encoding
Silhouette score
k-clusters
Data analysis mindset

THE RECOMMENDATIONS

Rouge Hill/Port Union/Highland Creek

This neighborhood in **Scarborough** experiences high foot traffic due to the presence of numerous transit stations, offices, and gas stations, all of which can significantly boost sales for a cafe.

Population

35,642

Median Household Income

\$ 109,785

Median Per postal code Population

- 26128 census year 2021

Median Per postal code Income

- \$65508 census year 2021

Median Per postal code Count of cafes - 1

Lawrence Park

This neighborhood in **Central Toronto** enjoys high foot traffic due to the numerous businesses, offices, clinics, and shopping plazas, all of which is good for cafe sales.

Population

16,058

Median Household Income

\$ 137,758

Median Per postal code Population

census year 2021

Median Per postal code Income

census year 2021

Median Per postal code Count of cafes - 1

Google Nearby Places

Milliken / Agincourt North / Steeles East

This neighborhood in **Scarborough** experiences high foot traffic due to the presence of numerous transit stations, offices, shopping plazas, churches, all of which will help the sales of a cafe.

Population

50,825

Median Household Income

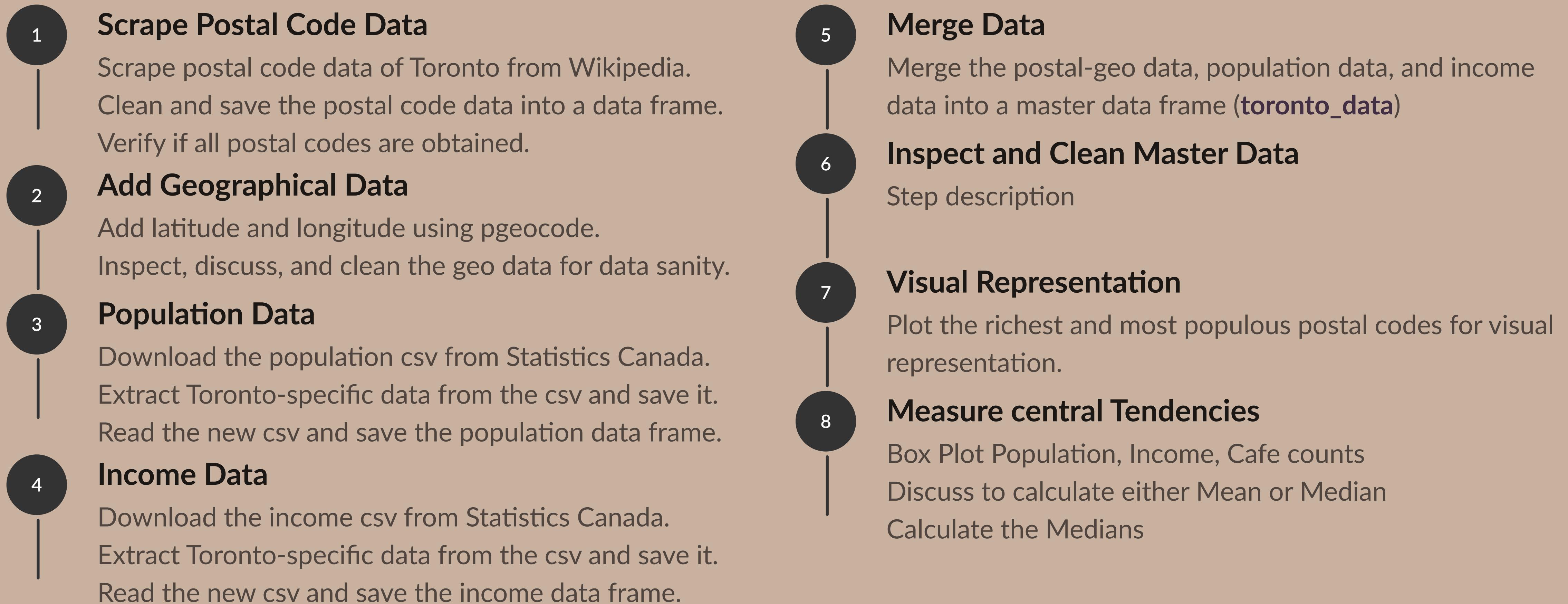
\$ 64,576

Approach

A commentary, describing the steps enabling the analysis.



WEB SCRAPING AND SECONDARY DATA SOURCES



API CALLS AND RE-USEABLE FUNCTIONS

9

API Selection

Analyze Google Places and Foursquare Places.
Set up an API call, read response, read the documentation.

10

Function to get Required Dictionary

Get all Place Type Cafe from the responses.
Use `get_places_cafe` function, append to `toronto_data`.

11

Data Storage

Store the responses as a csv.
Read the csv and create `toronto_data_df`
Store postal codes, geo information, income, population,
counts of cafes, and names of cafes.

12

Visualization MAP

Plot `toronto_data_df` on a map to enable visualization of
cafe competition.

13

Function Iteration

Iterate the function into `get_places_all` to get all place types
for postal codes.

14

Data Filtering, Function Iteration 2

Iterate `get_places_all` to filter out places with
'point_of_interest' as the first in type list and get the second
type if available (this place type is not of business use).

15

Final Data Storage

Store the response as a csv so that no more API calls are
needed while working.
Read the csv and create a data frame `toronto_data_places`.



ONE-HOT ENCODING AND K-MEANS

16

One-Hot Encoding

Perform **one-hot encoding** on the 'Place Type' column to prepare categorical data for the machine learning model.

17

Grouping and Analysis

Group by 'Postal Code' and calculate the mean of each one-hot encoded column to analyze the distribution of different place types across various postal codes in Toronto.

Identify the most common place types for each postal code in Toronto as frequencies.

18

Function Creation for Common Places

Create a function to identify the most common places for rows in the coded and grouped Data Frame, make a **lists of top 10 most common places for each postal code**.

19

Clustering

Use **Silhouette Score** to establish the number of clusters.

Set up and run the **K-means clustering algorithm**.

20

Final Data Frame

Final Data Frames using, dropping columns not necessary for conclusion and discussion. **Display all 4 clusters.**

21

Plot Stacked Bar Chart

Cluster Selection

Select the most suitable cluster for the cafe business environment using the stacked bar charts.

22

Prime Location In the cluster

Score the postal codes for Population and Income and sort the result to get the Top 3.



THE REPORT

*Here is the data analysis report
resulting from the market
segmentation study.*

CENTRAL TENDENCIES

Median Per postal code Population	- 26128	census year 2021
Median Per postal code Income	- \$65508	census year 2021
Median Per postal code Count of cafes - 1		Google Nearby Places

Population:

The median is around 25,000.

IQR spans from 15,000 to 40,000.

There is one outlier above 70,000.

Median Household Income:

The median is around 60,000.

IQR spans from 50,000 to 80,000.

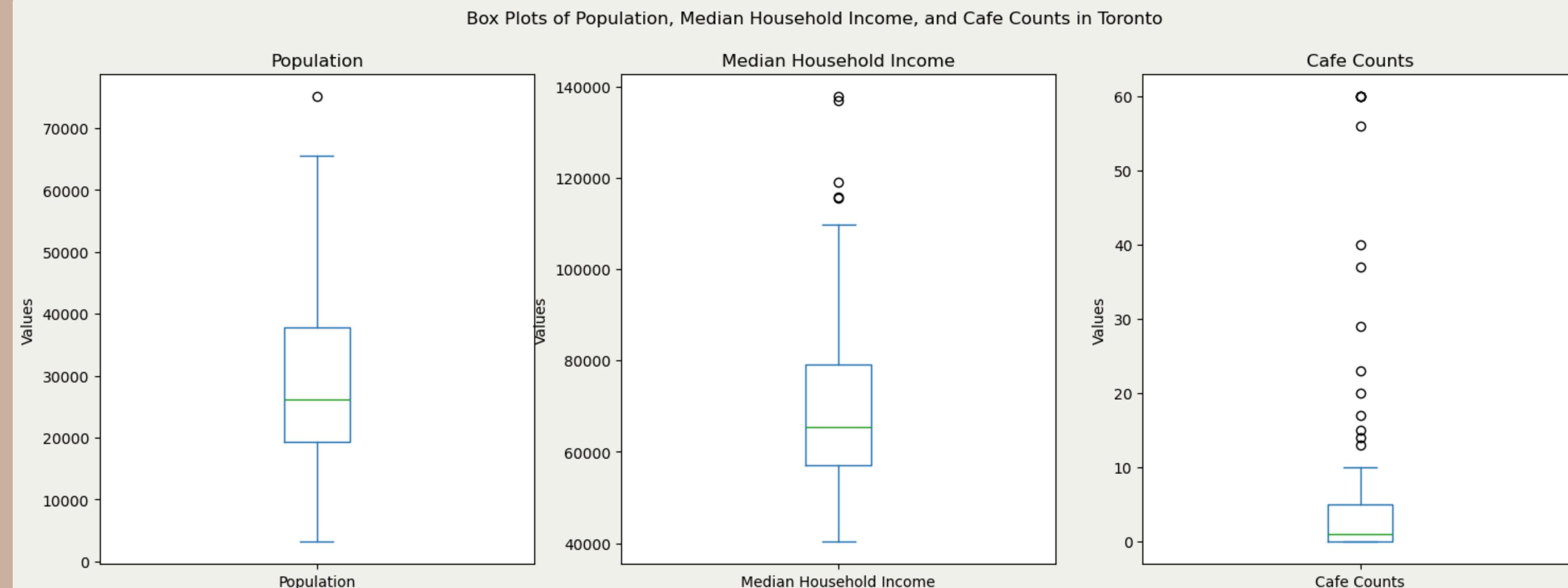
There are several outliers above 100,000.

Cafe Counts:

The median is around 2.

IQR spans from 1 to 6.

There are numerous outliers above 10,
with the highest being around 60.

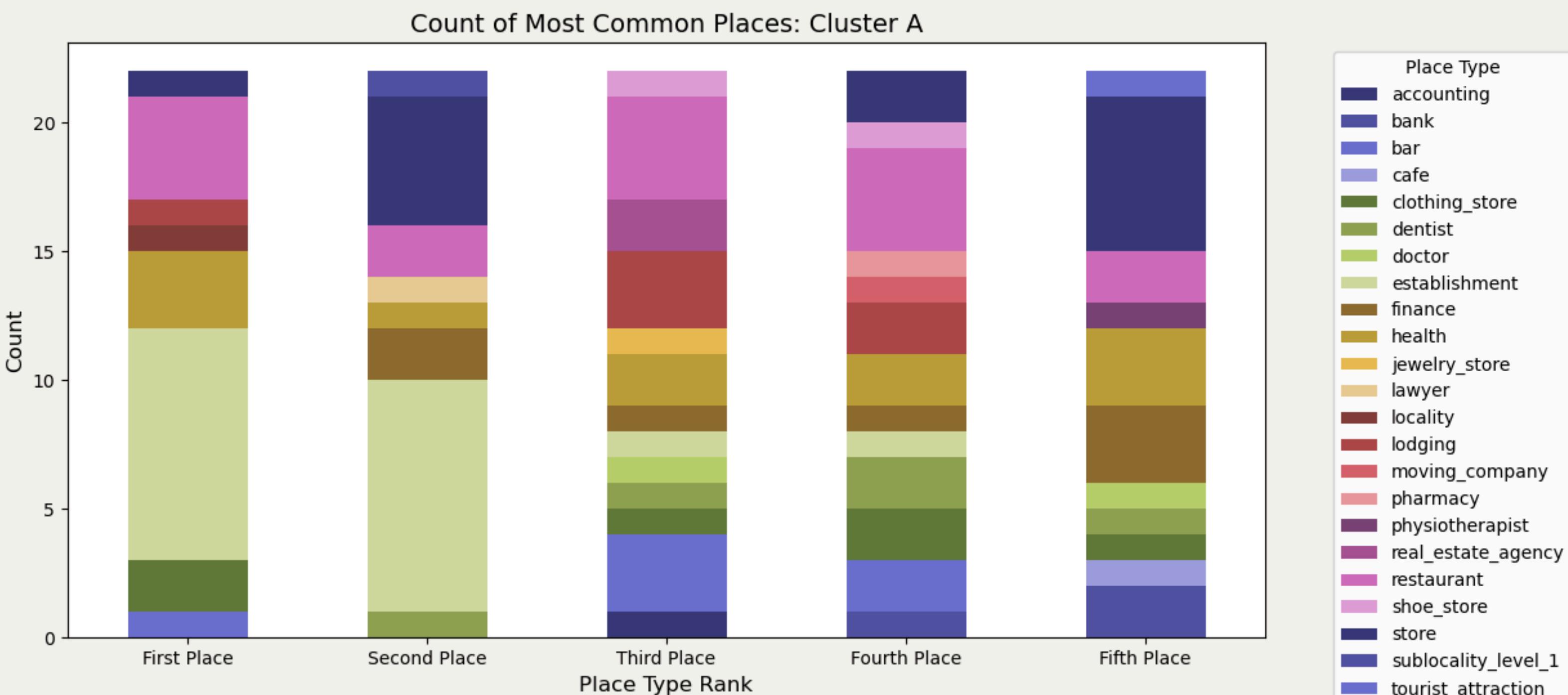


CLUSTERS

Cluster A:

A Business heavy area with restaurants and motels.

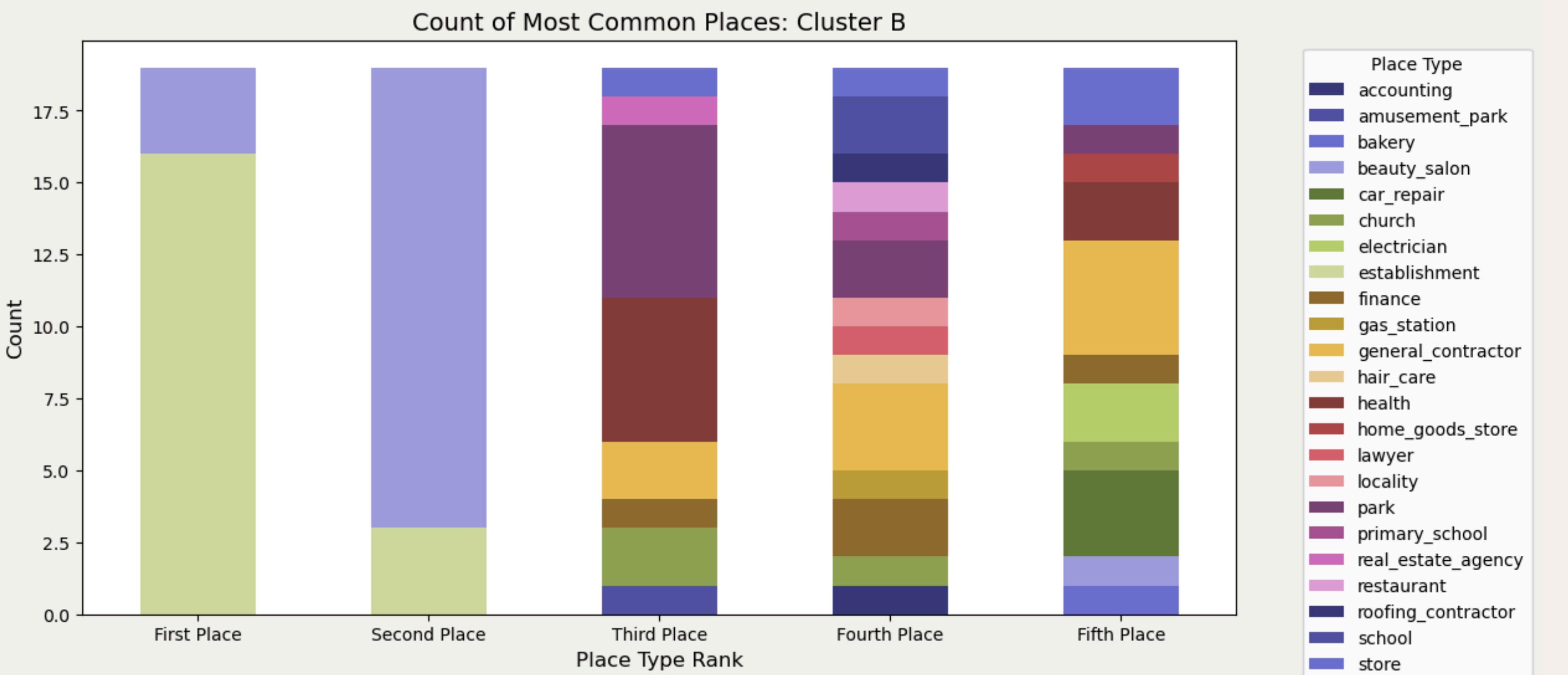
High concentration of cafes in postal codes M5B, M5C, M5E, M5G, and M5H.



Cluster B:

High footfall due to transit points,
making it ideal for new cafes.

Few cafes, with small counts in postal codes M1H, M1V, and M9P.

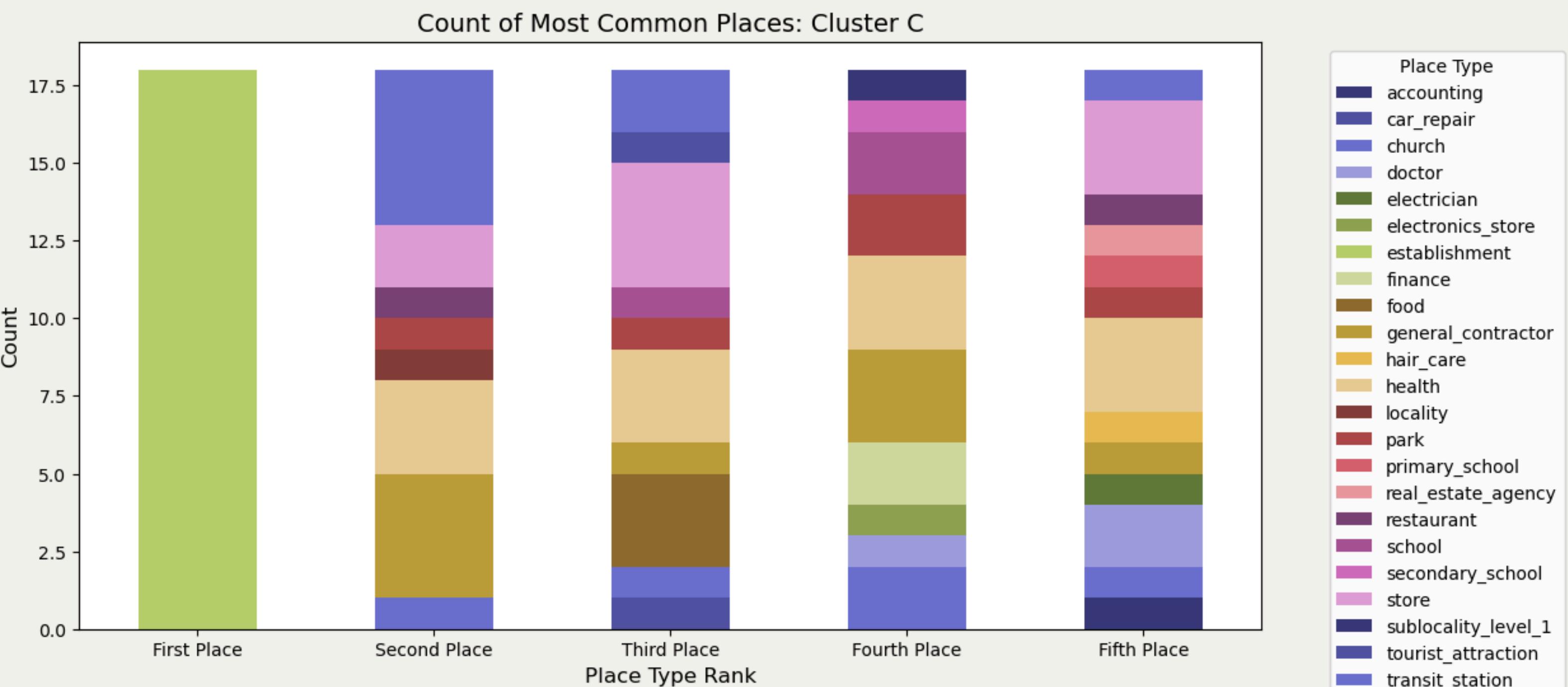


CLUSTERS

Cluster C:

A Business heavy area with shops and doctor clinics.

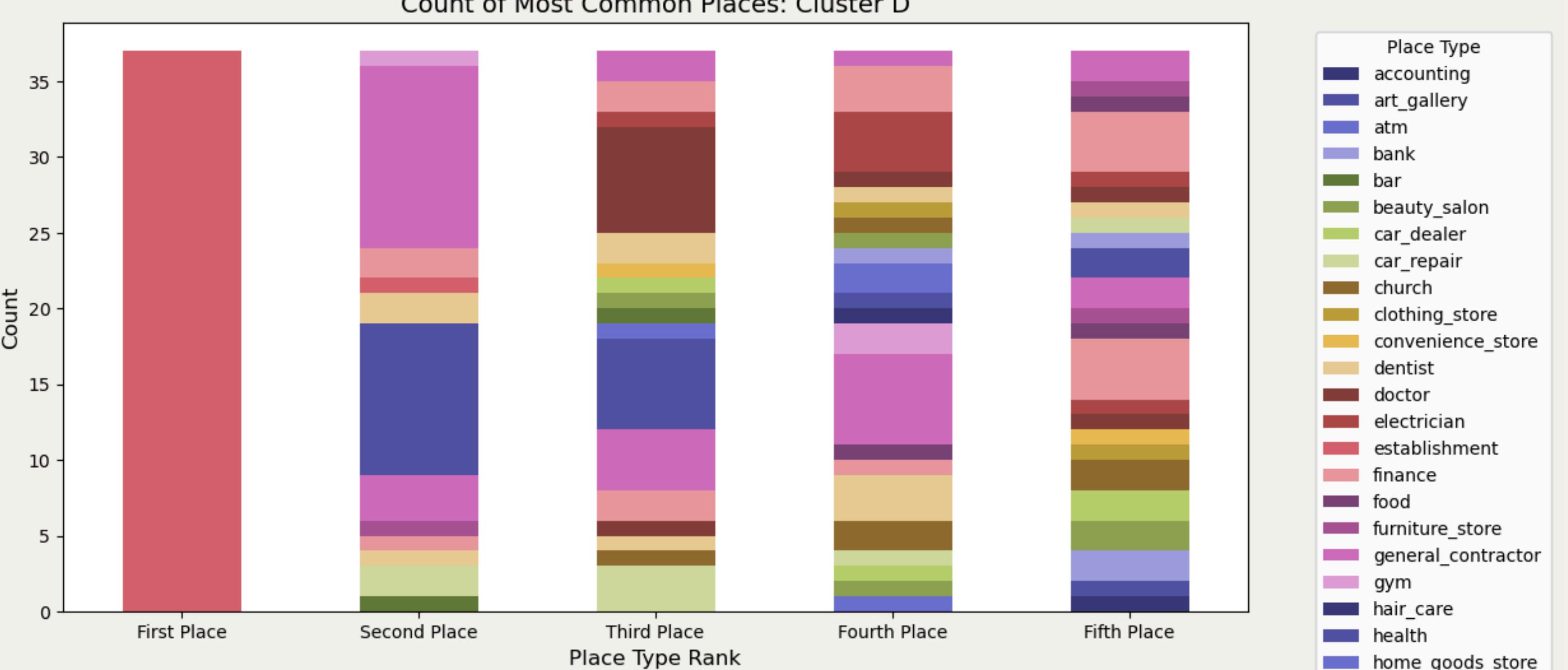
Moderate number of cafes in postal codes M3C, M4E, M5J, and M6K.



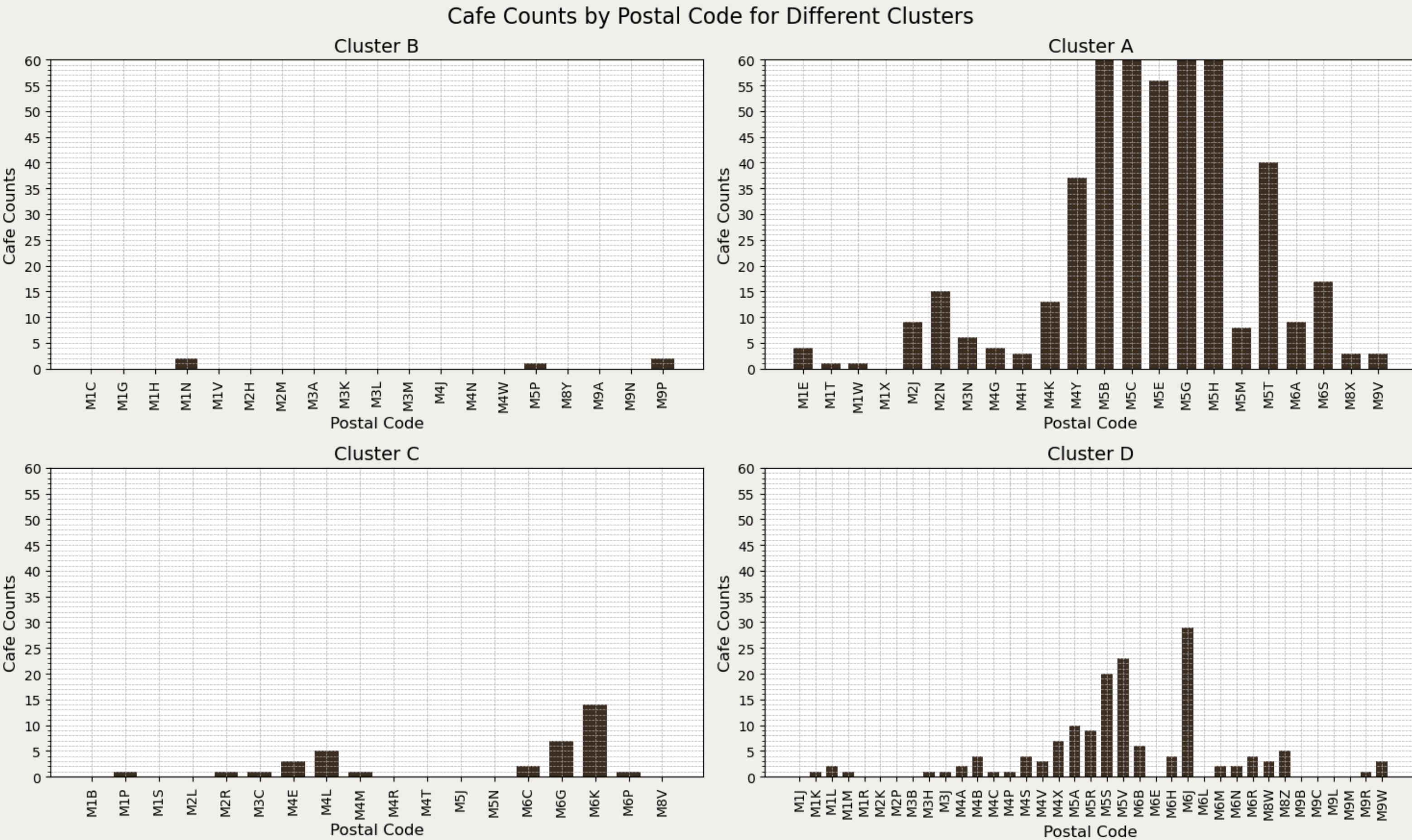
Cluster D:

A Industrial heavy area with some shops and clinics.

Notable number of cafes in postal codes M5V, M6G, and M6H.



COMPETITION



Cluster Selection

Cluster B Is the recommended cluster as it has high volume of businesses and offices, with lots of transit points. Such high footfall will yield high cafe sale volume.



RECOMMENDATIONS

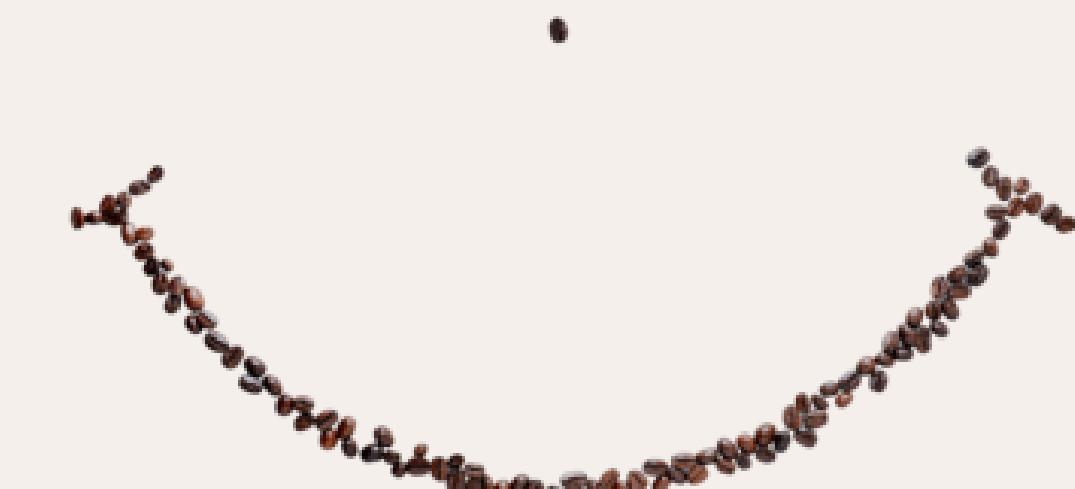
Rouge Hill / Port Union / Highland Creek

High household income: \$109,785, High population: 35,642



Lawrence Park

Very high household income: \$137,758, Sustainable population: 16,058



Milliken / Agincourt North / Steeles East

Comparable household income: \$64,576, Very high population: 50,825

	Postal Code	Borough	Neighborhood	Population	Median Household Income	Cafe Counts	1st Most Common Place Type	2nd Most Common Place Type	3rd Most Common Place Type	4th Most Common Place Type	5th Most Common Place Type	6th Most Common Place Type	7th Most Common Place Type	8th Most Common Place Type	9th Most Common Place Type	10th Most Common Place Type	Score
Top	M1C	Scarborough	Rouge Hill / Port Union / Highland Creek	35642.0	109785.0	0	transit_station	establishment	finance	gas_station	car_repair	general_contractor	accounting	car_wash	hair_care	real_estate_agency	50758.1
Second	M4N	Central Toronto	Lawrence Park	16058.0	137758.0	0	establishment	transit_station	health	lawyer	general_contractor	accounting	spa	gym	park	food	49358.0
Third	M1V	Scarborough	Milliken / Agincourt North / Steeles East / L...	50825.0	64576.0	0	transit_station	establishment	health	roofing_contractor	car_repair	church	finance	physiotherapist	pharmacy	primary_school	44786.9

THE CODE

*Here is a showcase of the work
that support the commentary of
this data analysis.*

Data Cleaning for Postal Code

This code shows how the combined data from Wikipedia has been separated in three columns as postal codes, borough and neighborhood.

Use of lambda function to simplify the code and put that value for the designed list.

```
# Iterate through each column in the DataFrame
for col in wiki:
    # Retrieve the first 3 characters of each string in the column and store in a list
    col_values = wiki[col].astype(str).apply(lambda x: x[:3]).tolist()
    postalcode_list.append(col_values)

    # Retrieve the substring from the 4th place value till the bracket open symbol "("
    substr_values = wiki[col].astype(str).apply(lambda x: x[4:x.find('(')] if '(' in x else x[4:]).tolist()
    borough_list.append(substr_values)

    # Retrieve the values within the brackets
    bracket_values = wiki[col].astype(str).apply(lambda x: x[x.find('(')+1:x.find(')')]) if '(' in x and ')' in x else ''
    neighborhood_list.append(bracket_values)

return postalcode_list, borough_list, neighborhood_list
```

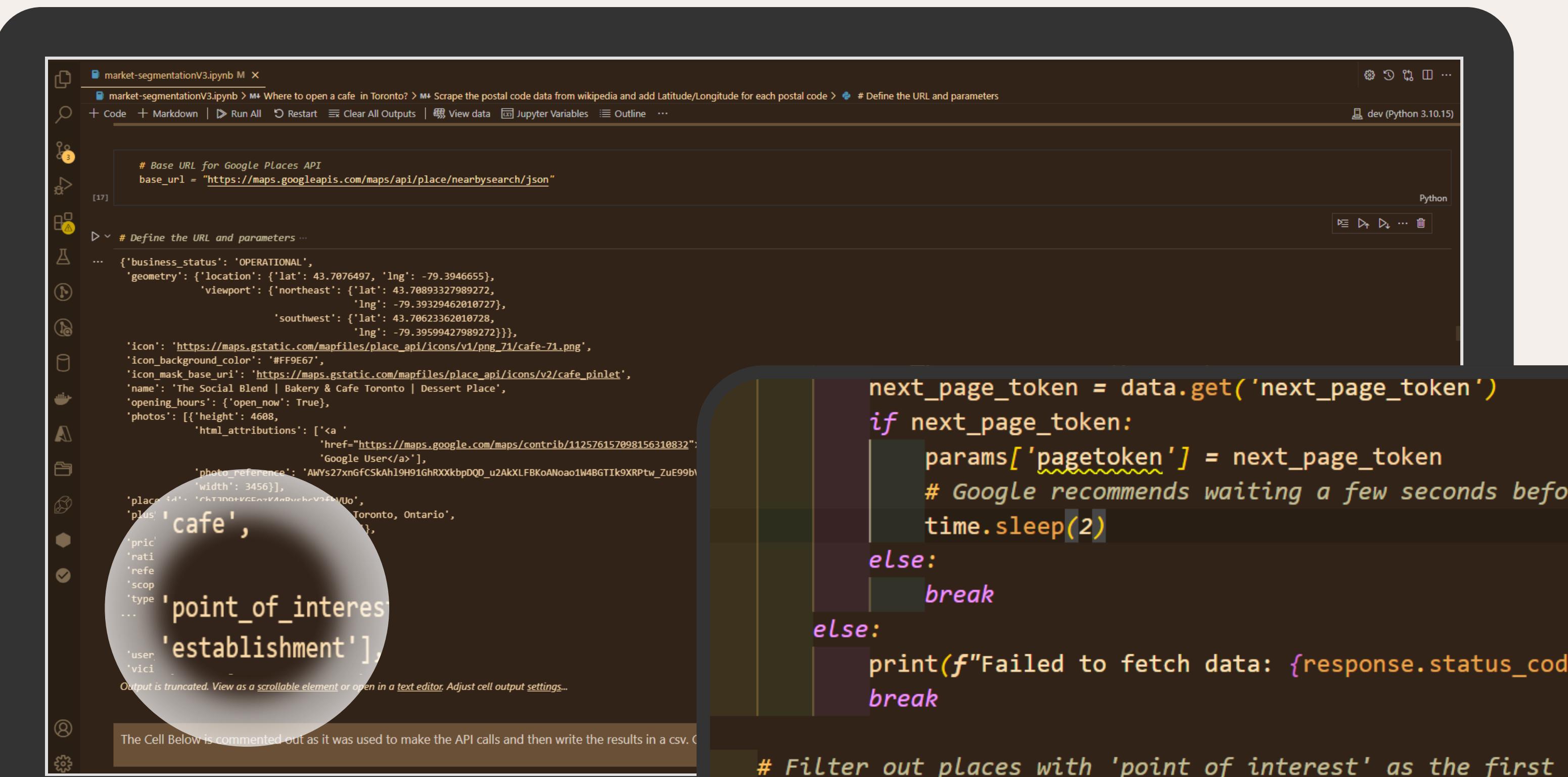


1	M1B Scarborough (Malvern / Rouge)	M2A Not assigned	M3A North York (Parkwoods)	M4A North York (Victoria Village)	M5A Downtown Toronto (Regent Park / Harbourfront)	M6A North York (Lawrence Manor / Lawrence Height...)	M7A Queen's Park (Ontario Provincial Government)	M8A Not assigned	M9A Etobicoke (Islington Avenue)
2		M2B Not assigned	M3B North York (Don Mills) North	M4B East York (Parkview Hill / Woodbine Gardens)	M5B Downtown Toronto (Garden District, Ryerson)	M6B North York (Glencairn)	M7B Not assigned	M8B Not assigned	M9B Etobicoke (West Deane Park / Princess Gard...
3		M2C Not assigned	M3C North York (Don Mills) South (Flemington P...	M4C East York (Woodbine Heights)	M5C Downtown Toronto (St. James Town)	M6C York (Humewood-Cedarvale)	M7C Not assigned	M8C Not assigned	M9C Etobicoke (Erigate / Bloordale Gardens / ...)
4	M1G Scarborough (Woburn)	M2G Not assigned	M3G Not assigned	M4G East York (Leaside)	M5E Downtown Toronto (Berczy Park)	M6E York (Caledonia-Fairbanks)	M7E Not assigned	M8E Not assigned	M9E Not assigned
5	M1H Scarborough (Cedarbrae)	M2H North York (Hillcrest Village)	M3H North York (Bathurst Manor / Wilson Height...	M4H East York (Thorndiffe Park)	M5H Downtown Toronto (Richmond / Adelaide / King)	M6H West Toronto (Dufferin / Dovercourt Village)	M7H Not assigned	M8H Not assigned	M9H Not assigned
6	M1J Scarborough (Scarborough Village)	M2J North York (Fairview / Henry Farm / Oriole)	M3J North York (Northwood Park / York University)	M4J East York East (The Danforth)	M5J Downtown Toronto (Harbourfront East / Union...)	M6J West Toronto (Little Portugal / Trinity)	M7J Not assigned	M8J Not assigned	M9J Not assigned
							M7K Not assigned	M8K Not assigned	M9K Not assigned
							M7L Not assigned	M8L Not assigned	M9L North York (Humber Summit)
							M7M Not assigned	M8M Not assigned	M9M North York (Humberlea / Emery)
							M7N Not assigned	M8N Not assigned	M9N York (Weston)
									M9P Etobicoke



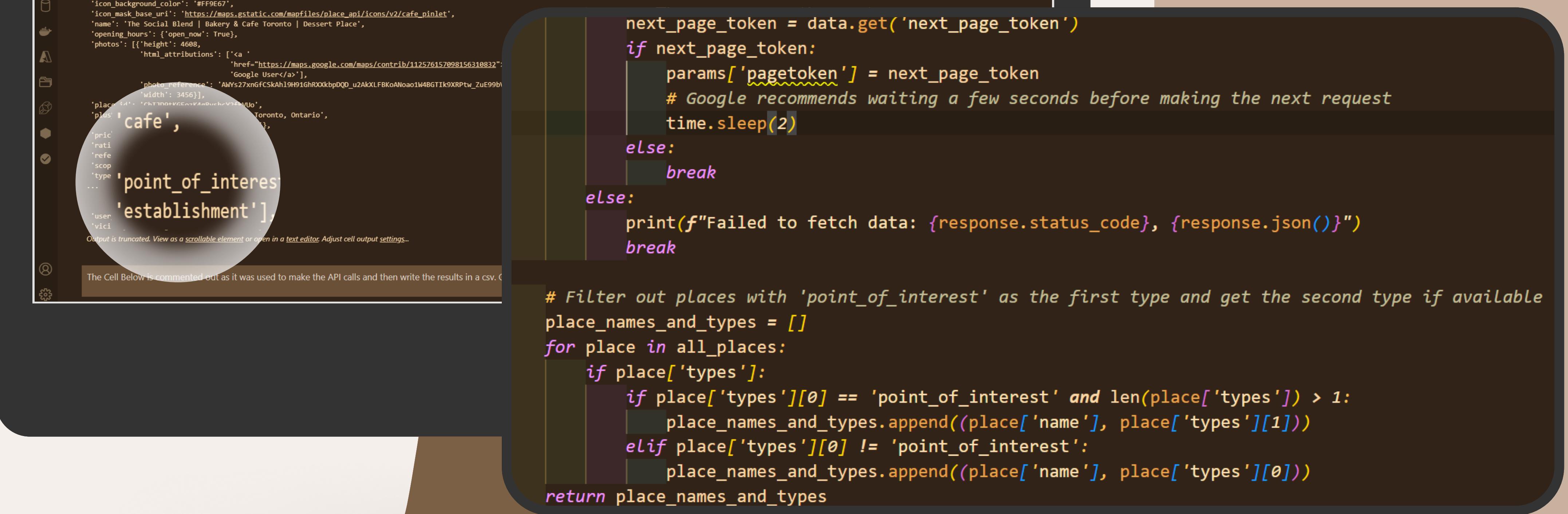
API call function

Iterated the API call function to get all “places for all of the postal codes. Filter out the ‘point_of_interest’ as it is not a place and is a google tag. This makes no sense for the study that we are doing.



```
# Base URL for Google Places API
base_url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json"

# Define the URL and parameters ...
... {'business_status': 'OPERATIONAL',
'geometry': {'location': {'lat': 43.7076497, 'lng': -79.3946655},
'vertex': {'northeast': {'lat': 43.70893327989272,
'lng': -79.39329462010727},
'southwest': {'lat': 43.70623362010728,
'lng': -79.39599427989272}}},
'icon': 'https://maps.gstatic.com/mapfiles/place_api/icons/v1/png_71/cafe-71.png',
'icon_background_color': '#FF9E67',
'icon_base_uri': 'https://maps.gstatic.com/mapfiles/place_api/icons/v2/cafe_pinlet',
'name': 'The Social Blend | Bakery & Cafe Toronto | Dessert Place',
'opening_hours': {'open_now': True},
'photos': [{'height': 4608,
'html_attributions': ['<a href="https://maps.google.com/maps/contrib/112576157098156310832" Google User/>'],
'photo_reference': 'AeWys27xngfCskAh19H91GhRXKbpDQD_u2AkXLFBKoANoao1W4BGTk9XRptw_ZuE99b',
'place_id': 'ChIJN0KCkVADuIcVJlo',
'plus': 'Toronto, Ontario',
'type': 'cafe',
'width': 3456}],
'price_level': 2,
'rating': 4.5,
'reference': 'AeWys27xngfCskAh19H91GhRXKbpDQD_u2AkXLFBKoANoao1W4BGTk9XRptw_ZuE99b',
'scopes': [],
'types': ['establishment'],
'users': [],
'version': 1}
...
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
The Cell Below is commented out as it was used to make the API calls and then write the results in a csv. C
```



```
next_page_token = data.get('next_page_token')
if next_page_token:
    params['pagetoken'] = next_page_token
    # Google recommends waiting a few seconds before making the next request
    time.sleep(2)
else:
    break
else:
    print(f"Failed to fetch data: {response.status_code}, {response.json()}")
    break

# Filter out places with 'point_of_interest' as the first type and get the second type if available
place_names_and_types = []
for place in all_places:
    if place['types']:
        if place['types'][0] == 'point_of_interest' and len(place['types']) > 1:
            place_names_and_types.append((place['name'], place['types'][1]))
        elif place['types'][0] != 'point_of_interest':
            place_names_and_types.append((place['name'], place['types'][0]))
return place_names_and_types
```



Silhouette Score Method

The silhouette score is a measure of how similar an object is to its own cluster compared to other clusters. Higher silhouette scores indicate better-defined clusters.

```
# Silhouette Score
silhouette_scores = []
for i in range(2, 11): # Silhouette score is not defined for a single cluster
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    score = silhouette_score(X, kmeans.labels_)
    silhouette_scores.append(score)

# Plot the Silhouette Scores
fig, ax = plt.subplots()
fig.patch.set_facecolor('#F4F0EC') # Set the face color of the figure
ax.plot(range(2, 11), silhouette_scores, color="#3e3023") # Set the color of the plotted line
ax.set_title('Silhouette Score Method')
ax.set_xlabel('Number of clusters')
ax.set_ylabel('Silhouette Score')
plt.show()
```



A screenshot of a Jupyter Notebook interface titled "market-segmentationV3.ipynb". The code cell contains Python code for calculating silhouette scores and plotting them. The code uses the KMeans algorithm from scikit-learn to find clusters and the silhouette_score function to calculate the silhouette score for each cluster. The plot shows the silhouette score for each number of clusters from 2 to 11. The x-axis is labeled "Number of clusters" and the y-axis is labeled "Silhouette Score". The background of the notebook is dark brown.

```
market-segmentationV3.ipynb M
market-segmentationV3.ipynb Where to open a cafe in Toronto? Scrape the postal code data from wikipedia and add Latitude/Longitude for each postal code # Define the URL and parameters
+ Code + Markdown | Run All Restart Clear All Outputs View data Jupyter Variables Outline ...
toronto_onehot_grouped_clustering = toronto_onehot_grouped.drop('Postal Code', axis=1)

# Use your DataFrame's data
X = toronto_onehot_grouped_clustering

# Silhouette Score
silhouette_scores = []
for i in range(2, 11): # Silhouette score is not defined for a single cluster
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    score = silhouette_score(X, kmeans.labels_)
    silhouette_scores.append(score)

# Plot the Silhouette Scores
fig, ax = plt.subplots()
fig.patch.set_facecolor('#F4F0EC') # Set the face color of the figure
ax.plot(range(2, 11), silhouette_scores, color="#3e3023") # Set the color of the plotted line
ax.set_title('Silhouette Score Method')
ax.set_xlabel('Number of clusters')
ax.set_ylabel('Silhouette Score')
plt.show()
```

The screenshot shows a Jupyter Notebook interface with several code cells and output sections. The notebook title is "market-segmentationV3.ipynb".

Code Cells:

```

ClusterA = toronto_final.loc[toronto_final['Cluster Labels'] == 0]
ClusterA = ClusterA.drop(['Cluster Labels'], axis=1)

# set number of clusters
kclusters = 4

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(toronto_onehot_grouped_clustering)

# check cluster Labels generated for each row in the dataframe
kmeans.labels_[0:10]

array([2, 1, 0, 1, 1, 3, 3, 3, 3, 1])

# add clustering Labels
toronto_onehot_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

toronto_onehot_merged = toronto_data_df

# merge toronto_grouped with toronto_data to add Latitude/Longitude for each neighborhood
toronto_onehot_merged = toronto_onehot_merged.join(toronto_onehot_sorted.set_index('Postal Code'), on='Postal Code', how='right')

toronto_onehot_merged.head() # check the last columns!

```

Output Sections:

Cafe Counts

Postal Code	Borough	Neighborhood	Establishments
M1E	Scarborough	Guildwood / Morningside / West Hill	4 establishment
M1T	Scarborough	Clarks Corners / Tam O'Shanter / Sullivan	1 establishment
M1W	Scarborough	Steeles West / L'Amoreaux West	
M1X	Scarborough	Upper Rouge	14810.0
M2J	North York	Fairview / Henry Farm / Oriole	61761.0
M2N	North York	Willowdale	75100.0
M3N	North York	Downsview	40846.0
M4G	East York	Leaside	19598.0
M4H	East York	Thorncliffe Park	18698.0
M4K	East Toronto	The Danforth West / Riverdale	30913.0

1st Most Common Place Type

Place Type	Count
coffee shop	14810.0
fast food	61761.0
supermarket / grocery	75100.0
bank	40846.0
post office	19598.0
bar / night club	18698.0
gas bar	30913.0
convenience store	65948.0
pharmacy	14810.0
grocery store	61761.0
liquor store	75100.0
superstore	40846.0
fast food	19598.0
coffee shop	18698.0
supermarket / grocery	30913.0
bank	65948.0
post office	14810.0
bar / night club	61761.0
gas bar	75100.0
convenience store	40846.0
pharmacy	19598.0
liquor store	18698.0
superstore	30913.0

2nd Most Common Place Type

Place Type	Count
coffee shop	61761.0
fast food	75100.0
supermarket / grocery	40846.0
bank	19598.0
post office	18698.0
bar / night club	30913.0
gas bar	65948.0
convenience store	14810.0
pharmacy	61761.0
liquor store	75100.0
superstore	40846.0
fast food	19598.0
coffee shop	18698.0
supermarket / grocery	30913.0
bank	65948.0
post office	14810.0
bar / night club	61761.0
gas bar	75100.0
convenience store	40846.0
pharmacy	19598.0
liquor store	18698.0
superstore	30913.0

3rd Most Common Place Type

Place Type	Count
supermarket / grocery	75100.0
fast food	61761.0
bank	40846.0
post office	19598.0
bar / night club	18698.0
gas bar	30913.0
convenience store	14810.0
pharmacy	61761.0
liquor store	75100.0
superstore	40846.0
fast food	19598.0
coffee shop	18698.0
supermarket / grocery	30913.0
bank	65948.0
post office	14810.0
bar / night club	61761.0
gas bar	75100.0
convenience store	40846.0
pharmacy	19598.0
liquor store	18698.0
superstore	30913.0

4th Most Common Place Type

Place Type	Count
bank	40846.0
post office	19598.0
bar / night club	18698.0
gas bar	30913.0
convenience store	14810.0
pharmacy	61761.0
liquor store	75100.0
superstore	40846.0
fast food	19598.0
coffee shop	18698.0
supermarket / grocery	30913.0
bank	65948.0
post office	14810.0
bar / night club	61761.0
gas bar	75100.0
convenience store	40846.0
pharmacy	19598.0
liquor store	18698.0
superstore	30913.0

5th Most Common Place Type

Place Type	Count
post office	19598.0
bar / night club	18698.0
gas bar	30913.0
convenience store	14810.0
pharmacy	61761.0
liquor store	75100.0
superstore	40846.0
fast food	19598.0
coffee shop	18698.0
supermarket / grocery	30913.0
bank	65948.0
post office	14810.0
bar / night club	61761.0
gas bar	75100.0
convenience store	40846.0
pharmacy	19598.0
liquor store	18698.0
superstore	30913.0

6th Most Common Place Type

Place Type	Count
gas bar	30913.0
convenience store	14810.0
pharmacy	61761.0
liquor store	75100.0
superstore	40846.0
fast food	19598.0
coffee shop	18698.0
supermarket / grocery	30913.0
bank	65948.0
post office	14810.0
bar / night club	61761.0
gas bar	75100.0
convenience store	40846.0
pharmacy	19598.0
liquor store	18698.0
superstore	30913.0

7th Most Common Place Type

Place Type	Count
convenience store	14810.0
pharmacy	61761.0
liquor store	75100.0
superstore	40846.0
fast food	19598.0
coffee shop	18698.0
supermarket / grocery	30913.0
bank	65948.0
post office	14810.0
bar / night club	61761.0
gas bar	75100.0
convenience store	40846.0
pharmacy	19598.0
liquor store	18698.0
superstore	30913.0

K-mean Cluster

Using the k-mean clusters, divided the dataset into 4 clusters based on the most common places types that comes into the assigned postal codes.



NEXT STEPS

The next iteration of the code will focus on expanding this data model both horizontally and vertically!

Horizontally, the ethnicity data from Statistics Canada's 2021 census can be incorporated, adding another dimension to the market clustering model.

Additionally, more geospatial data can improve postal code zoning. The current radius-based method leads to overlaps, this will enhance accuracy.

Vertically, a decision tree to automatically identify the best market clusters for a cafe, taking into account various dimensions and suggesting optimal markets can be developed .