The Battle of Neighborhoods: An E-Scooter business.

Introduction

City of Calgary in Canada recently announced that E-Scooters will be back in the city this summer 2021, and it will be operated by two companies Bird Canada and Neuron, both companies will provide and maintain 1500 E-Scooters altogether. This project aims at analysing Calgary traffic data (Pedestrian and Bike) and using FourSquare's API to determine the best locations to place these 1500 E-Scooters to maximize the company's customer satisfaction and profit.

Background

Micro-mobility transportation services are recently becoming a popular trend in Canada. One of the top cities in Canada, Calgary, is among the cities to adopt this shared micro-mobility services (e-Bike and e-Scooter). Benefits of electric scooter includes: Ease of use, Green House Gases (GHG) savings, it encourages people to walk and take public transport, thus improving physical health and it also saves time on short trips.

Business Problem

Almost every business is started with an intention to make profit. The capital to start and manage this E-Scooter business is huge and if the company is to generate profit, it must grow its customers/subscriber's base.

Also, since micro-mobility services is a fairly new program, it has faced several restrictions or regulations from the government, some of them include: no riding scooter on busy sidewalks or busy roads.

The business problem could be defined as:

1. How do we serve lots of customers and increase customer satisfaction by providing our scooters at the right location or hotspots?
2. What are the hotspots for E-Scooters that comply to governments regulations?

Thus, this project will utilize Calgary's traffic data to explore the busy roads and places with high pedestrian traffic, so scooters won't be placed in such places. FourSquare API will be used to determine hotspots for the scooters.

Target Audience

The target audience are mainly entrepreneurs or investors interested in shared micro-mobility services. This project will be of particular interest to Bird Canada and Neuron as they are about to launch in few weeks.

Data

To analyze data and obtain any meaningful insight from it, raw data must first be collected.

Calgary's pedestrian and bike traffic data were collected from the city of Calgary's Open Data Portal. Location information (Latitude and Longitude) include postal codes, borough and neighborhood in Calgary where collected (Web-Scrapped) from Wikipedia. Data regarding Calgary's top venues and location were collected using FourSquare API. Hence, the following are data sources used for this project:

1. (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_T) – To retrieve postal codes, borough, Latitude and Longitude data.
2. (https://data.calgary.ca/Transportation-Transit/Pedestrian-Traffic-Count/nz3v-2tau) – To retrieve pedestrian traffic count.
3. (https://data.calgary.ca/Transportation-Transit/Bike-Traffic-Count/pubj-un4t) – To retrieve bike traffic count.
4. (https://developer.foursquare.com) – To retrieve the top venues or hotspots in Calgary.

Data Cleaning and Feature Extraction

The first data was obtained from Wikipedia. Wikipedia webpage was web-scrapped using pandas read_html syntax. The table already had all the information we needed for our analysis; therefore we won't be using Geolocator to obtain both longitude and latitude.

The second data was a csv file downloaded from city Calgary's open data portal. The file had 31,862 rows and 102 columns. The columns contained information on: **Date** the data was collected, the **location** (address), time the information was collected from (12am to 11:45pm).

The third data was also csv file downloaded from city Calgary's open data portal. The file had 60,000+ rows and 102 columns. The columns contained information on: **Date** the data was collected, the **location** (address), time the information was collected from (12am to 11:45pm).

The last data will be retrieved using FourSquare API and a personal login credential. Data from the API will be used to determine the top venues and hotspots in the city. Neighbors will then be clustered based on their most popular services.

Data Analysis and Visualization

In analyzing the dataset, the following tasks were performed.

1. **Imported the required libraries for the analysis of the data**.

```
In [1]:
# IMPORTING NECESSARY LIBRARIES

from bs4 import BeautifulSoup # this module helps in web scrapping.
import requests  # this module helps us to download a web page
import pandas as pd
```

```
In [55]:
import json # library to handle JSON files

import numpy as np

from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests

from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

import folium # map rendering library
```

2. **Obtained the required data and cleaned it**. Some data were web-scrapped others were downloaded in csv format.

```
In [2]:
# DEFINING URL FOR SCRAPING CALGARY'S POSTAL-CODE, FROM WIKIPEDIA
url = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_T"
```

The data from the HTML table was scraped into a DataFrame using (read_html)

```
In [3]:
dataframe_list = pd.read_html(url, flavor='bs4')
```

```
In [4]:
len(dataframe_list)
```

```
Out[4]: 6
```

Dataframe containing Postal Code, Borough, Neighborhood, latitude and Longitude

```
In [6]:
dataframe_list[1]
```

Out[6]:

	Postal Code	Borough	Neighborhood	Latitude	Longitude
0	T1A	Medicine Hat	Central Medicine Hat	50.036460	-110.679250
1	T2A	Calgary	Penbrooke Meadows, Marlborough	51.049680	-113.964320
2	T3A	Calgary	Dalhousie, Edgemont, Hamptons, Hidden Valley	51.126060	-114.143158
3	T4A	Airdrie	East Airdrie	51.272450	-113.986980
4	T5A	Edmonton	West Clareview, East Londonderry	53.5899	-113.4413
...	...	...	...	...	...
175	T5Z	Edmonton	West Lake District	53.5966	-113.4882
176	T6Z	Not assigned	Not assigned	Not assigned	Not assigned
177	T7Z	Stony Plain	Not assigned	53.5202	-114.0135
178	T8Z	Not assigned	Not assigned	Not assigned	Not assigned
179	T9Z	Not assigned	Not assigned	Not assigned	Not assigned

180 rows × 5 columns

In [12]:
```
# Reading the csv file

bike = pd.read_csv('Bike_data.csv')
ped = pd.read_csv('ped.csv')
```

In [13]:
```
bike.head()
```

Out[13]:

	Date	User Type	Monitoring Location	Direction	Total	0000	0015	0030	0045	0100	...	2145	2200	2215	2230	2245	2300	2315	2330	2345	
0	2018/01/01	Bikes	10 Street North of 5 Ave NW	Northbound	23	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2018-01T00:00: Bikes Street N of
1	2018/01/01	Bikes	10 Street North of 5 Ave NW	Southbound	24	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	2018-01T00:00: Bikes Street N of
2	2018/01/01	Bikes	10 Street North of 5 Ave NW	TOTAL DIR	47	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	2018-01T00:00: Bikes Street N of
3	2018/01/02	Bikes	10 Street North of 5 Ave NW	Northbound	16	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2018-02T00:00: Bikes Street N of
4	2018/01/02	Bikes	10 Street North of 5 Ave NW	Southbound	12	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2018-02T00:00: Bikes Street N of

5 rows × 102 columns

Cleaning the data

Dropping NAN values from the bike data

In [15]:
```
bike.dropna(inplace=True)
```

In [16]:
```
bike = bike.reset_index(drop=True)
```

Grouping time from (00:00 to 23:45) into four groups [(12am-9am), (9am-2pm), (2pm-8pm) and (8pm-11:45pm)]

In [17]:
```
# Grouping bike time columns together

b1 = bike.columns[5:42]
b2 = bike.columns[42:62]
b3 = bike.columns[62:86]
b4 = bike.columns[86:101]

b1
```

Out[17]:
```
Index(['0000', '0015', '0030', '0045', '0100', '0115', '0130', '0145', '0200',
       '0215', '0230', '0245', '0300', '0315', '0330', '0345', '0400', '0415',
       '0430', '0445', '0500', '0515', '0530', '0545', '0600', '0615', '0630',
       '0645', '0700', '0715', '0730', '0745', '0800', '0815', '0830', '0845',
       '0900'],
      dtype='object')
```

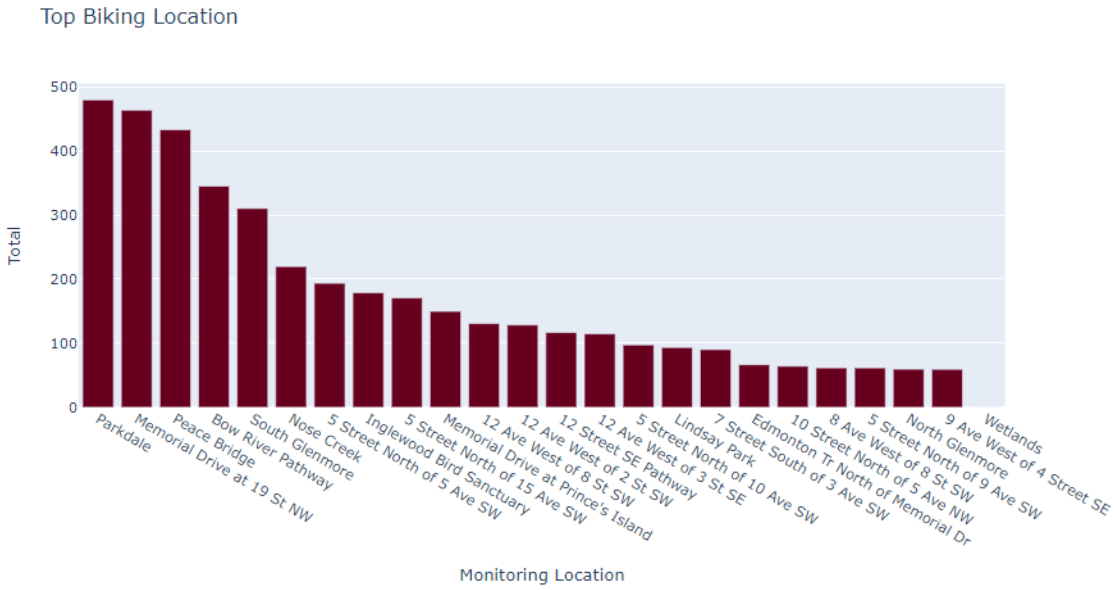
In [18]:
```
# Creating new bike columns by summming and grouping column together

bike ['12am-9am'] = bike[b1].sum(axis=1)
bike ['9am-2pm'] = bike[b2].sum(axis=1)
bike ['2pm-8pm'] = bike[b3].sum(axis=1)
bike ['8pm-11:45pm'] = bike[b4].sum(axis=1)
```

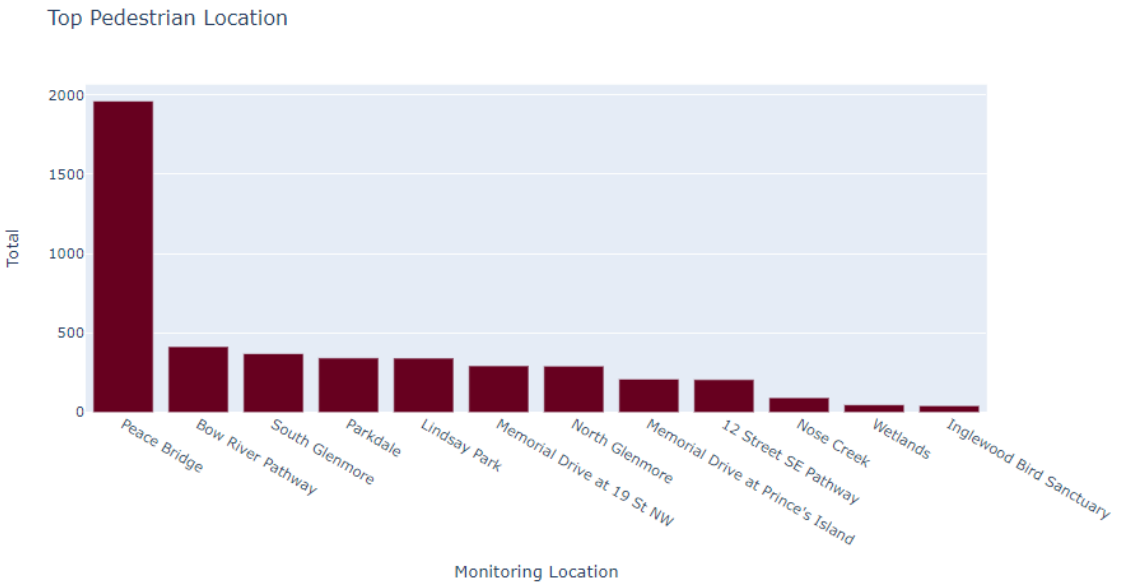
In [19]:
```
# Grouping ped time columns together

p1 = ped.columns[5:42]
p2 = ped.columns[42:62]
p3 = ped.columns[62:86]
p4 = ped.columns[86:101]

p2
```

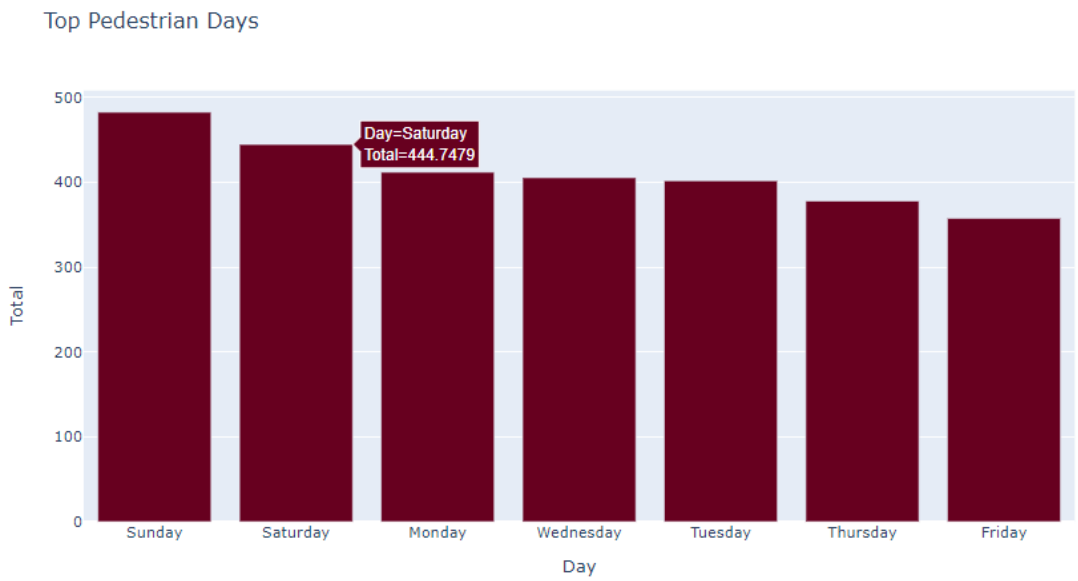
Out[19]:
```
Index(['0915', '0930', '0945', '1000', '1015', '1030', '1045', '1100', '1115',
       '1130', '1145', '1200', '1215', '1230', '1245', '1300', '1315', '1330',
       '1345', '1400'],
      dtype='object')
```

3. **Exploratory Data Analysis**.

Using visualization to understand the dataset and generating meaningful relationships that exist in the dataset.

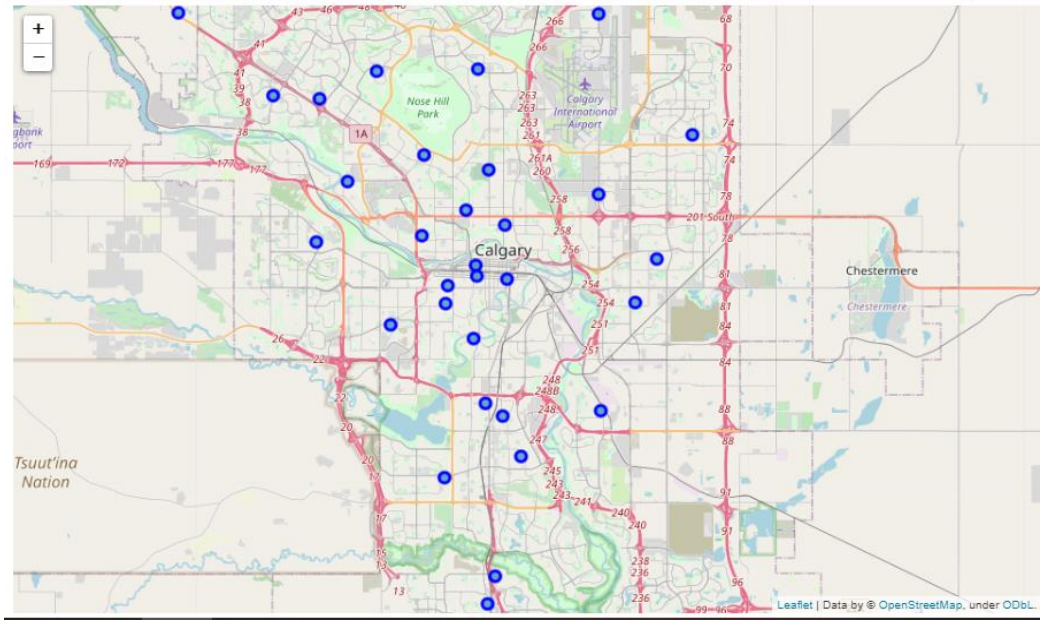






4. City of Calgary's map was created using Folium package and its neighborhood were superimposed on it.

```
Calgary_map = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(Calgary['Latitude'], Calgary['Longitude'], Calgary['Borough'], Calgary['Neighborh
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(Calgary_map)

Calgary_map
```



5. Using FourSquare API, venues in Calgary were generated.

```
Creating Calgary venues, using the Calgary data with the defined function
```

```
calgary_venues = getNearbyVenues(names=Calgary['Neighborhood'],
                                 latitudes=Calgary['Latitude'],
                                 longitudes=Calgary['Longitude']
                                )
```

```
Penbrooke Meadows, Marlborough
Dalhousie, Edgemont, Hamptons, Hidden Valley
Forest Lawn, Dover, Erin Woods
Montgomery, Bowness, Silver Springs, Greenwood
Lynnwood Ridge, Ogden, Foothills Industrial, Great Plains
Rosscarrock, Westgate, Wildwood, Shaganappi, Sunalta
Bridgeland, Greenview, Zoo, YYC
Lakeview, Glendale, Killarney, Glamorgan
Inglewood, Burnsland, Chinatown, East Victoria Park, Saddledome
Hawkwood, Arbour Lake, Citadel, Ranchlands, Royal Oak, Rocky Ridge
Highfield, Burns Industrial
Discovery Ridge, Signal Hill, West Springs, Christie Estates, Patterson, Cougar Ridge
Queensland, Lake Bonavista, Willow Park, Acadia
Martindale, Taradale, Falconridge, Saddle Ridge
Thorncliffe, Tuxedo Park
Sandstone, MacEwan Glen, Beddington, Harvest Hills, Coventry Hills, Panorama Hills
Brentwood, Collingwood, Nose Hill
Tuscany, Scenic Acres
Mount Pleasant, Capitol Hill, Banff Trail
Cranston, Auburn Bay, Mahogany
Kensington, Westmont, Parkdale, University
Northeast Calgary
City Centre, Calgary Tower
Symons Valley
Connaught, West Victoria Park
Northwest Calgary
Elbow Park, Britannia, Parkhill, Mission
Southeast Calgary
South Calgary (Altadore / Bankview / Richmond)
Oak Ridge, Haysboro, Kingsland, Kelvin Grove, Windsor Park
Braeside, Cedarbrae, Woodbine
Midnapore, Sundance
Rundle, Whitehorn, Monterey Park
Millrise, Somerset, Bridlewood, Evergreen
Douglas Glen, McKenzie Lake, Copperfield, East Shepard
```

Out[65]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Dalhousie, Edgemont, Hamptons, Hidden Valley	51.12606	-114.143158	Petro-Canada	51.128068	-114.138057	Gas Station
1	Dalhousie, Edgemont, Hamptons, Hidden Valley	51.12606	-114.143158	Edgemont City	51.126473	-114.138997	Asian Restaurant
2	Dalhousie, Edgemont, Hamptons, Hidden Valley	51.12606	-114.143158	Friends Cappuccino Bar & Bake Shop	51.126370	-114.138676	Café
3	Dalhousie, Edgemont, Hamptons, Hidden Valley	51.12606	-114.143158	Mac's	51.128309	-114.137902	Convenience Store
4	Forest Lawn, Dover, Erin Woods	51.03180	-113.978600	Bonasera Pizza And Sports Bar	51.029893	-113.982543	Bar

Out[76]:

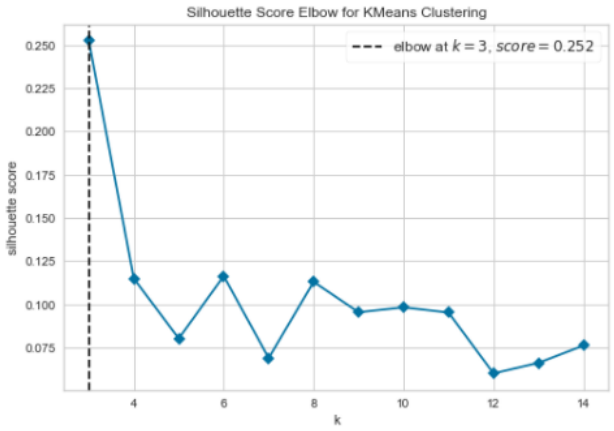
	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Braeside, Cedarbrae, Woodbine	Ice Cream Shop	Hockey Rink	Gym	Convenience Store	Pharmacy	Gas Station	Pizza Place	Coffee Shop	Pub	Yoga Studio
1	Brentwood, Collingwood, Nose Hill	Dog Run	Electronics Store	Yoga Studio	Food Truck	Dim Sum Restaurant	Diner	Donut Shop	Dry Cleaner	Falafel Restaurant	Fast Food Restaurant
2	Bridgeland, Greenview, Zoo, YYC	Fast Food Restaurant	Noodle House	Steakhouse	Pizza Place	Falafel Restaurant	Restaurant	Sandwich Place	Chinese Restaurant	Seafood Restaurant	Brewery
3	City Centre, Calgary Tower	Coffee Shop	Mediterranean Restaurant	Bakery	Sushi Restaurant	Sporting Goods Shop	Camera Store	Pizza Place	Pub	Restaurant	Sandwich Place
4	Connaught, West Victoria Park	Brewery	Yoga Studio	Camera Store	Gourmet Shop	Italian Restaurant	Donut Shop	Middle Eastern Restaurant	Moroccan Restaurant	Coffee Shop	Pizza Place

6. The neighborhood venues were clustered.

```
Using Silhouette Score to determine optimum number of clusters
```

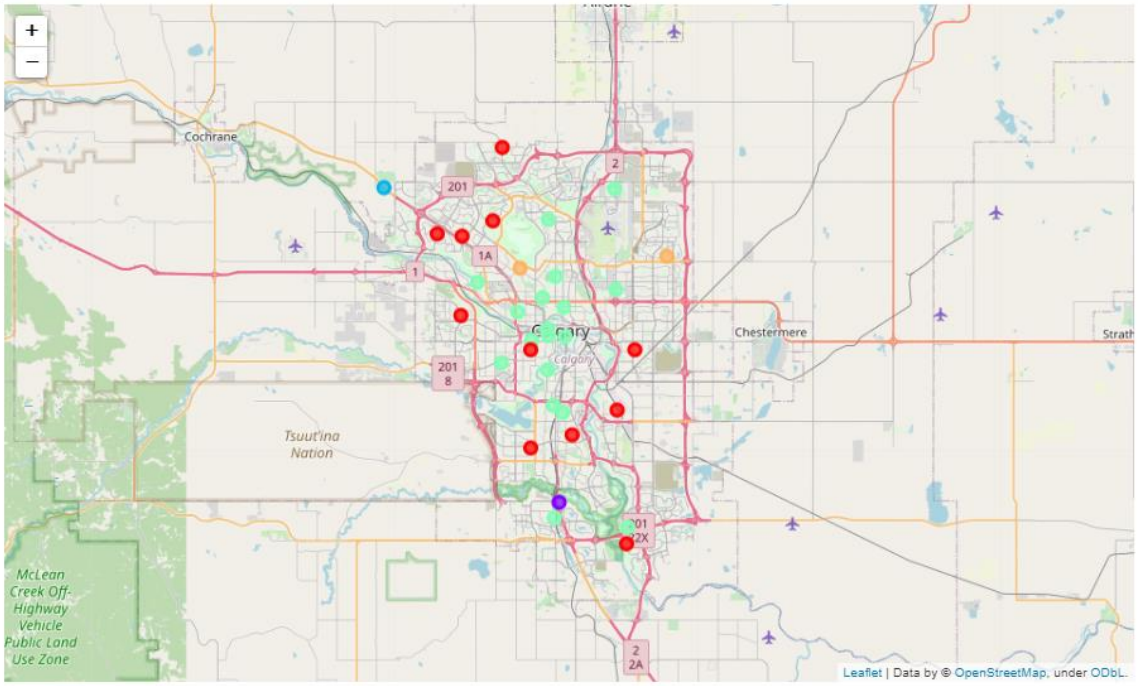
In [78]:

```
kclusters = (3,15)

calgary_grouped_clustering = calgary_grouped.drop('Neighborhood', 1)

# run k-means clustering
model = KMeans(random_state=0)
visualizer = KElbowVisualizer(model, k=kclusters, metric='silhouette', timings=False) # random state = 0

visualizer.fit(calgary_grouped_clustering)
visualizer.show()
```



7. Map showing cluster locations in Calgary.

Out[88]:



8. The clusters were examined.

Cluster 4

In [99]:
```
Cluster_4 = calgary_merged.loc[calgary_merged['Cluster Labels'] == 3, calgary_merged.columns[[2] + list(range(5, calgary_merg
Cluster_4
```

Out[99]:

	Neighborhood	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
3	Montgomery, Bowness, Silver Springs, Greenwood	3	Food Court	Bank	Coffee Shop	Scenic Lookout	Yoga Studio	Diner	Dog Run	Donut Shop	Dry Cleaner	Electronics Store
5	Rosscarrock, Westgate, Wildwood, Shaganappi, S...	3	Pub	Sandwich Place	Indian Restaurant	Health & Beauty Service	Mexican Restaurant	Mediterranean Restaurant	Fried Chicken Joint	Breakfast Spot	Bookstore	Sports Bar
6	Bridgeland, Greenview, Zoo, YYC	3	Fast Food Restaurant	Noodle House	Steakhouse	Pizza Place	Falafel Restaurant	Restaurant	Sandwich Place	Chinese Restaurant	Seafood Restaurant	Brewery
7	Lakeview, Glendale, Killarney, Glamorgan	3	Wine Shop	Coffee Shop	Grocery Store	Food Court	Dim Sum Restaurant	Diner	Dog Run	Donut Shop	Dry Cleaner	Electronics Store
8	Inglewood, Burnsland, Chinatown, East Victoria...	3	Deli / Bodega	Performing Arts Venue	Steakhouse	Ice Cream Shop	Museum	Coffee Shop	Cocktail Bar	Liquor Store	Pub	Restaurant
10	Highfield, Burns Industrial	3	American Restaurant	Skating Rink	Food Court	Dim Sum Restaurant	Diner	Dog Run	Donut Shop	Dry Cleaner	Electronics Store	Falafel Restaurant
14	Thorncliffe, Tuxedo Park	3	Vietnamese Restaurant	Convenience Store	Bar	Fast Food Restaurant	Massage Studio	Yoga Studio	Food Truck	Diner	Dog Run	Donut Shop
15	Sandstone, MacEwan Glen, Beddington, Harvest H...	3	Bank	Grocery Store	Italian Restaurant	Pharmacy	Pizza Place	Coffee Shop	Liquor Store	Yoga Studio	Flea Market	Diner

In [102]:
```
Venues = Cluster_4[['Neighborhood','1st Most Common Venue']].groupby('1st Most Common Venue').count().sort_values(by='Neighbo
Venues.rename(columns={'1st Most Common Venue':'Venue'}, inplace=True)
Venues
```

Out[102]:

	Venue	Neighborhood
0	Coffee Shop	3
1	Hotel	2
2	American Restaurant	1
3	Bank	1
4	Brewery	1
5	Burger Joint	1
6	Curling Ice	1
7	Deli / Bodega	1
8	Fast Food Restaurant	1
9	Food Court	1
10	Park	1
11	Pub	1
12	Sporting Goods Shop	1
13	Vietnamese Restaurant	1
14	Wine Shop	1

Results and Discussion

From the analysis the following deductions were made:

1)

- Peak Pedestrian Hrs = 9am - 2pm

- Biking peak hours range from 2pm to 8pm

From the above, to cater and satisfy more customers e-scooters should be readily available at these peak hours, so more people can use it.

2)

Peace bridge has both high pedestrian traffic and bike traffic and should be avoided due the city of Calgary's regulations.

3)

Top days are Saturdays and Wednesdays. Wednesdays are not too busy with pedestrians and bikes; their traffic volume is about average. Saturdays are one of the busiest days, utilizing the optimum time between 9am and 2pm will cater for more customers.

4)

The best location to cluster scooters for optimum customer satisfaction is Cluster_4 neighborhoods. From the analysis these areas have several activities going on such as Coffee shops, Restaurants, parks, Burger spots and pubs.

Conclusion

The K means clustering wasn't satisfactory for this data, i would like to suggest that other clustering methods be tried such as 'Hierarchical Clustering' or DBSCAN clustering. A better cluster will be able to provide more accurate locations to cluster the e-scooters.

The purpose of this project was to acquire useful information using Foursquare API and other location data to help improve an e-scooter business. From our analysis, we were able to understand Calgary’s traffic pattern (Peak hours, peak days, peak locations etc.), we also explored top spots/venues in the city.

References

1. (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_T) – To retrieve postal codes, borough, Latitude and Longitude data.
2. (https://data.calgary.ca/Transportation-Transit/Pedestrian-Traffic-Count/nz3v-2tau) – To retrieve pedestrian traffic count.
3. (https://data.calgary.ca/Transportation-Transit/Bike-Traffic-Count/pubj-un4t) – To retrieve bike traffic count.
4. (https://developer.foursquare.com) – To retrieve the top venues or hotspots in Calgary.
5. The Battle of Neighborhoods: Starting a Coffee Shop Business | by Diardano Raihan | Towards Data Science