IBM Applied Data Science Capstone

The Battle of Neighborhoods: John Data is looking for the best neighborhood to live in Vancouver

By: Gabriel Gomez

1. Introduction

1.1. General

- Vancouver is one of the most populated cities in province of British Columbia
- The population at Vancouver is about 631,486 as per the 2016 census, this makes it the 8th city with most population in Canada
- There are approximately 79,989 businesses, while most businesses employ less than 20 employees, most employees work in larger businesses
- The University of British Columbia is a global center of research, consistently ranked among the top 20 public universities in the world.

1.2. Problem Description

John Data is going to start a Master program at the University of British Columbia at Vancouver and he's looking for an apartment to rent.

He tried to look for the nearest apartment to the school, but they are out of his budget, so he wants to look for other options. John already checked the price range of the apartments and the best neighborhoods are Kitsilano, Fairview, West Point Grey, and Arbutus-Ridge the only thing that keeps him from deciding is the venues near each neighborhood in which he could work part-time.

Each neighborhood will be analyzed to select the one with the most venues where he could work at and create a list of venues

2. Data Description

2.1. City of Vancouver Open Data Portal

The data of the neighborhoods and their location will be taken directly from this portal. The CSV file will be used to get the latitude and longitude of the specified neighborhoods

Data Link: https://opendata.vancouver.ca/explore/dataset/local-area-boundary/table/?dataChart=eyJxdWVyaWVzIjpbeyJjb25maWciOnsiZGF0YXNldCI6ImxvY2FsLWFyZWEtYm91bmRhcnkiLCJvcHRpb25zIjp7ImxvY2F0aW9uIjoiMTIsNDkuMjQ3NCwtMTIzLjEyNDAyIn19LCJjaGFydHMiOlt7ImFsaWduTW9udGgiOnRydWUsInR5cGUiOiJjb2x1bW4iLCJmdW5jIjoiQ09VTlQiLCJzY2llbnRpZmljRGlzcGxheSI6dHJ1ZSwiY29sb3IiOiIjMDI3OUIxIn1dLCJ4QXhpcyI6Im5hbWUiLCJtYXhwb2ludHMiOjUwLCJzb3J0IjoiIn1dLCJ0aW1lc2NhbGUiOiIiLCJkaXNwbGF5TGVnZW5kIjp0cnVlLCJhbGlnbk1vbnRoIjp0cnVlfQ%3D%3D&location=12,49.2474,-123.12402

CSV File Location: https://opendata.vancouver.ca/explore/dataset/local-area-boundary/download/?format=csv&timezone=America/Denver&lang=en&use_labels_for_header=true&csv_separator=%3B

The information obtained from this DB is:

1. MapID: The identifier of the neighborhood
2. Name: Official name of the Neighborhood
3. Geom: Spatial representation of Local Area
4. geo_point_2d: Latitude and Longitude of the neighborhood

2.2. Foursquare

The data about the venues located in each of the neighborhoods selected by John is needed. Foursquare will be used in order to get the locational information. This data provider has information about the venues (such as names, locations, menus and more) per location and will be helpful to get the data that we need.

The data retrieved from Foursquare contained information of venues within a radius of 500 meters of the longitude and latitude of each neighborhood. The information obtained per venue as follows:

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

3. Methodology

3.1. Data Exploration

The first step is to get the data from the neighborhoods in Vancouver. This data was located at the City of Vancouver Open Data Portal (https://opendata.vancouver.ca/pages/home/). This portal has many data sources but the one that we need is the Local area boundary data, which can be extracted via API or exporting the file. To retrieve the data, the data was exported but using the CSV download link and use pandas in order read the CSV

```
In [2]: url = "https://opendata.vancouver.ca/explore/dataset/local-area-boundary/download/?format=csv&timezone=America/Denver&lang=en&use_labels_for_header=true&csv_separator=%3B"

In [3]: s = req.get(url).content

In [4]: c = pd.read_csv(io.StringIO(s.decode(encoding='cp1252')),error_bad_lines=False,sep=';')

In [5]: c
```

Out[5]:

	MAPID	Name	Geom	geo_point_2d
0	AR	Arbutus-Ridge	{"type": "Polygon", "coordinates": [[[-123.152...	49.2468049108,-123.161669238
1	CBD	Downtown	{"type": "Polygon", "coordinates": [[[-123.112...	49.2807470711,-123.116567008
2	FAIR	Fairview	{"type": "Polygon", "coordinates": [[[-123.145...	49.2645404871,-123.131048865
3	GW	Grandview-Woodland	{"type": "Polygon", "coordinates": [[[-123.077...	49.2764396102,-123.066728221
4	HS	Hastings-Sunrise	{"type": "Polygon", "coordinates": [[[-123.056...	49.277934053,-123.040269923

Since the latitude and longitude are required in order to locate the neighborhoods described before, the column geo_point_2d was splitted into “lat” and “lon” using the comma to separate both coordinates.

```
In [6]: c[['lat','lon']] = c.geo_point_2d.str.split(",",expand=True)

In [7]: c.drop(columns=['geo_point_2d','Geom'])
```

Out[7]:

	MAPID	Name	lat	lon
0	AR	Arbutus-Ridge	49.2468049108	-123.161669238
1	CBD	Downtown	49.2807470711	-123.116567008
2	FAIR	Fairview	49.2645404871	-123.131048865
3	GW	Grandview-Woodland	49.2764396102	-123.066728221
4	HS	Hastings-Sunrise	49.277934053	-123.040269923

After getting the data frame, variables were created to specify each of the neighborhood’s coordinates

In [8]:

```
#Kitsilano
Kitlat = (c[c['Name'].str.contains("Kitsilano")].reset_index(drop=True)).loc[0]["lat"]
Kitlon = (c[c['Name'].str.contains("Kitsilano")].reset_index(drop=True)).loc[0]["lon"]


#Fairview
Fairlat = (c[c['Name'].str.contains("Fairview")].reset_index(drop=True)).loc[0]["lat"]
Fairlon = (c[c['Name'].str.contains("Fairview")].reset_index(drop=True)).loc[0]["lon"]


#West Point Grey
Westlat = (c[c['Name'].str.contains("West Point Grey")].reset_index(drop=True)).loc[0]["lat"]
Westlon = (c[c['Name'].str.contains("West Point Grey")].reset_index(drop=True)).loc[0]["lon"]


#Arbutus-Ridge
Arblat = (c[c['Name'].str.contains("Arbutus-Ridge")].reset_index(drop=True)).loc[0]["lat"]
Arblon = (c[c['Name'].str.contains("Arbutus-Ridge")].reset_index(drop=True)).loc[0]["lon"]

print("Kitsilano:","(",Kitlat , Kitlon,")")
print("Fairview:","(",Fairlat , Fairlon,")")
print("West Point Grey:","(",Westlat , Westlon,")")
print("Arbutus-Ridge:","(",Arblat , Arblon,")")
```

```
Kitsilano: ( 49.2675398494 -123.16329474 )
Fairview: ( 49.2645404871 -123.131048865 )
West Point Grey: ( 49.2684012111 -123.203467483 )
Arbutus-Ridge: ( 49.2468049108 -123.161669238 )
```

3.2. Data Geocoding

Once the coordinates of the neighborhoods to analyze were retrieved, the Foursquare API was used in order to get the quantity of venues near each neighborhood (500m radius) and their names

In [11]:
```
rad = 500

#Kitsilano
Kiturl = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    Kitlat,
    Kitlon,
    rad,
    LIMIT)

#Fairview
Fairurl = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    Fairlat,
    Fairlon,
    rad,
    LIMIT)


#West Point Grey
Westurl = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    Westlat,
    Westlon,
    rad,
    LIMIT)


#Arbutus-Ridge
Arburl = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    Arblat,
    Arblon,
    rad,
    LIMIT)

print(Kiturl)
print(Fairurl)
print(Westurl)
print(Arburl)
```

In [12]:
```
Kit_ven = (req.get(Kiturl).json())['response']['groups'][0]['items']
Fair_ven = (req.get(Fairurl).json())['response']['groups'][0]['items']
West_ven = (req.get(Westurl).json())['response']['groups'][0]['items']
Arb_ven = (req.get(Arburl).json())['response']['groups'][0]['items']


Kitnearby_ven = json_normalize(Kit_ven)
Fairnearby_ven = json_normalize(Fair_ven)
Westnearby_ven = json_normalize(West_ven)
Arbnearby_ven = json_normalize(Arb_ven)

Kitnearby_ven
```

```
In [13]: filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']

         #Kitsilano
         Kitnearby_ven = Kitnearby_ven.loc[:, filtered_columns]
         Kitnearby_ven['venue.categories'] = Kitnearby_ven.apply(get_category_type, axis=1)
         Kitnearby_ven.columns = [col.split(".")[-1] for col in Kitnearby_ven.columns]

         #Fairview
         Fairnearby_ven = Fairnearby_ven.loc[:, filtered_columns]
         Fairnearby_ven['venue.categories'] = Fairnearby_ven.apply(get_category_type, axis=1)
         Fairnearby_ven.columns = [col.split(".")[-1] for col in Fairnearby_ven.columns]

         #West Point Grey
         Westnearby_ven = Westnearby_ven.loc[:, filtered_columns]
         Westnearby_ven['venue.categories'] = Westnearby_ven.apply(get_category_type, axis=1)
         Westnearby_ven.columns = [col.split(".")[-1] for col in Westnearby_ven.columns]

         #Arbutus-Ridge
         Arbnearby_ven = Arbnearby_ven.loc[:, filtered_columns]
         Arbnearby_ven['venue.categories'] = Arbnearby_ven.apply(get_category_type, axis=1)
         Arbnearby_ven.columns = [col.split(".")[-1] for col in Arbnearby_ven.columns]
```

4. Results

4.1. Neighborhood with most venues

With the result divided per neighborhood, the last step to get the quantity of venues near each neighborhood is to count the results

```
In [14]: print("Kitsilano has", Kitnearby_ven['name'].count(),"venues")
         print("Fairview has", Fairnearby_ven['name'].count(),"venues")
         print("West Point Grey has", Westnearby_ven['name'].count(),"venues")
         print("Arbutus-Ridge has", Arbnearby_ven['name'].count(),"venues")

         Kitsilano has 49 venues
         Fairview has 26 venues
         West Point Grey has 6 venues
         Arbutus-Ridge has 2 venues
```

The result of our analysis is that Kitsilano has the most venues near, which makes it the ideal neighborhood for John to rent an apartment that might be near his future workplace.

4.2. List of venues

Now that the best neighborhood is clear, the list of venues near is:

	name	categories	lat	lng
0	The Only Cafe	Café	49.2682	-123.16554
1	Cafe Lokal	Coffee Shop	49.26817	-123.16471
2	Guanaco Salvadoran Cuisine food truck	Food Truck	49.26825	-123.16175
3	Terra Breads	Bakery	49.26814	-123.15928
4	Raisu	Japanese Restaurant	49.26824	-123.15843
5	Dark Table	Restaurant	49.26832	-123.16466
6	Thomas Haas Patisserie	Dessert Shop	49.26397	-123.16326
7	The Naam Restaurant	Vegetarian / Vegan Restaurant	49.2683	-123.16705
8	Au Comptoir	French Restaurant	49.2682	-123.15704
9	Nat's New York Pizzeria	Pizza Place	49.26403	-123.166
10	Maria's Taverna	Greek Restaurant	49.26809	-123.15808
11	Market Meats	Deli / Bodega	49.2681	-123.15821
12	Semperviva Yoga	Yoga Studio	49.26397	-123.16494
13	Burgoo Bistro	Mac & Cheese Joint	49.26821	-123.15681
14	Chewie’s Biscuit Co	Southern / Soul Food Restaurant	49.26825	-123.16879
15	Darby's Public House	Pub	49.26834	-123.16838
16	Sunshine Diner	Diner	49.26404	-123.16552
17	Dairy Queen	Ice Cream Shop	49.26423	-123.16489
18	Linh Cafe	Breakfast Spot	49.26824	-123.16906
19	Whole Foods Market	Grocery Store	49.26827	-123.15705
20	White Spot	Burger Joint	49.26395	-123.16327
21	Feastro	Food Truck	49.26813	-123.15671
22	Drexoll Games	Toy / Game Store	49.26842	-123.16993
23	COBS Bread	Bakery	49.26817	-123.15795
24	Mr. Red Cafe	Vietnamese Restaurant	49.26406	-123.16589
25	Menchie's	Frozen Yogurt Shop	49.26395	-123.16399
26	Broadway International Wine Shop	Wine Shop	49.26406	-123.16746
27	Kitsilano Wine Cellar	Wine Shop	49.26829	-123.15656
28	Iki Japanese Restaurant	Japanese Restaurant	49.26404	-123.16751
29	Starbucks	Coffee Shop	49.26443	-123.16717
30	Shoppers Drug Mart	Pharmacy	49.26797	-123.15781
31	Peaceful Restaurant 和平饭店	Chinese Restaurant	49.26822	-123.15958
32	Rowan's Roof	American Restaurant	49.268	-123.15838
33	Browns Socialhouse Kitsilano	Gastropub	49.2682	-123.15748
34	Starbucks	Coffee Shop	49.26877	-123.15849
35	Starbucks	Coffee Shop	49.26799	-123.15672
36	Darby's Cold Beer and Wine	Liquor Store	49.26819	-123.1683
37	RBC Royal Bank	Bank	49.26837	-123.15965
38	Safeway Canada	Supermarket	49.26853	-123.1584
39	RBC Royal Bank	Bank	49.2643	-123.16806
40	Hi Nippon Japanese Restaurant	Japanese Restaurant	49.2682	-123.15693
41	Uncle Fatih's Pizza	Pizza Place	49.26412	-123.16776
42	New Apple Farm Market	Grocery Store	49.26836	-123.15831
43	Nusa Coffee	Coffee Shop	49.26914	-123.1661
44	Buen Café	Coffee Shop	49.26813	-123.15857
45	Bus Stop 50063 (2,22,32)	Bus Stop	49.26866	-123.16814
46	Mistral French Bistro	French Restaurant	49.26402	-123.16438
47	Big Johnny's Pizza House	Pizza Place	49.26841	-123.16896
48	Rogers	Electronics Store	49.26375	-123.16621

5. Discussion

- The results on this analysis are based only on the venues near each neighborhood, they might or might not be hiring, but with more venues the probability of finding a part-time job increases
- A further investigation about the best place to rent was attempted but since an open database with the price and location of apartments in rent couldn't be found, only the 4 specified neighborhoods were used.

6. Conclusion

- Pandas was a great help in order to retrieve the information of the Vancouver neighborhoods, since there were some issues with the API.
- Data analysis techniques used in this project were of great help and will be helpful in the solution of future business problems.
- Thanks to this project I was able to pinpoint the areas that I still lack in and will keep practicing to improve.
- Although the problem in this project was created just for the sake of this assignment, investigating about the university and the area made me consider the master degree in Vancouver, and if I ever decide to go there, I will most likely improve this investigation to help me find a place to live in.