

The Battle of the Neighborhoods (Jeff) - Crimes in Vancouver

1. A description of the problem and a discussion of the background

We will analyse and discuss the safest place in Vancouver where, It would be safe for opening a new bussiness or even to live, by analysing the rate crime, and also those kinds of crimes, we will plot the map also to get know about if there is some park our beach around the place

2. A description of the data and how it will be used to solve the problem

To do our analyse we will use a open data set from Kaggle data contains Vancouver crimes rate in 2018 Considering the following information:

TYPE - Crime type, @ YEAR - Recorded year, @ MONTH - Recorded month, @ DAY - Recorded day, @ HOUR - Recorded hour, @ MINUTE - Recorded minute, @ HUNDRED_BLOCK - Recorded block, @ NEIGHBOURHOOD - Recorded neighborhood, @ X- GPS longitude, @ Y - GPS latitude

Data set URL: <https://www.kaggle.com/agilesifaka/vancouver-crime-report/version/2>

```
In [1]: #install the packages
!pip install opencage
```

```

!pip install folium
!pip install geopy

# import the libraries
import pandas as pd
import numpy as np
from opencage.geocoder import OpenCageGeocode
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.style.use('ggplot')
import matplotlib.cm as cm
import matplotlib.colors as colors
import folium
import requests
from pandas.io.json import json_normalize
from sklearn.cluster import KMeans

#python magic function
%matplotlib inline

print('all libraries imported and packages installed')

```

Collecting opencage

Downloading opencage-1.2.1-py3-none-any.whl (6.1 kB)

Requirement already satisfied: Requests>=2.2.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from opencage) (2.23.0)

Collecting backoff>=1.10.0

Downloading backoff-1.10.0-py2.py3-none-any.whl (31 kB)

Requirement already satisfied: six>=1.4.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from opencage) (1.15.0)

Requirement already satisfied: pyopenssl>=0.15.1 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from opencage) (19.1.0)

Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from Requests>=2.2.0->opencage) (1.25.9)

Requirement already satisfied: certifi>=2017.4.17 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from Requests>=2.2.0->opencage) (2020.4.5.2)

Requirement already satisfied: chardet<4,>=3.0.2 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from Requests>=2.2.0->opencage) (3.0.2)

Requirement already satisfied: idna<3,>=2.5 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from Requests>=2.2.0->opencage) (2.10)

```

ebook/lib/python3.6/site-packages (from requests>=2.2.0->opencage) (3.
0.4)
Requirement already satisfied: idna<3,>=2.5 in /srv/conda/envs/noteboo
k/lib/python3.6/site-packages (from Requests>=2.2.0->opencage) (2.9)
Requirement already satisfied: cryptography>=2.8 in /srv/conda/envs/not
ebook/lib/python3.6/site-packages (from pyopenssl>=0.15.1->opencage)
(2.9.2)
Requirement already satisfied: cffi!=1.11.3,>=1.8 in /srv/conda/envs/no
tebook/lib/python3.6/site-packages (from cryptography>=2.8->pyopenssl>=
0.15.1->opencage) (1.14.0)
Requirement already satisfied: pycparser in /srv/conda/envs/notebook/li
b/python3.6/site-packages (from cffi!=1.11.3,>=1.8->cryptography>=2.8->
pyopenssl>=0.15.1->opencage) (2.20)
Installing collected packages: backoff, opencage
Successfully installed backoff-1.10.0 opencage-1.2.1
Collecting folium
  Downloading folium-0.11.0-py2.py3-none-any.whl (93 kB)
    |████████████████████████████████████████| 93 kB 2.2 MB/s eta 0:00:011
Requirement already satisfied: numpy in /srv/conda/envs/notebook/lib/py
thon3.6/site-packages (from folium) (1.18.5)
Requirement already satisfied: requests in /srv/conda/envs/notebook/li
b/python3.6/site-packages (from folium) (2.23.0)
Requirement already satisfied: Jinja2>=2.9 in /srv/conda/envs/notebook/
lib/python3.6/site-packages (from folium) (2.11.2)
Collecting branca>=0.3.0
  Downloading branca-0.4.1-py3-none-any.whl (24 kB)
Requirement already satisfied: certifi>=2017.4.17 in /srv/conda/envs/no
tebook/lib/python3.6/site-packages (from requests->folium) (2020.4.5.2)
Requirement already satisfied: chardet<4,>=3.0.2 in /srv/conda/envs/not
ebook/lib/python3.6/site-packages (from requests->folium) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /srv/conda/envs/noteboo
k/lib/python3.6/site-packages (from requests->folium) (2.9)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1
in /srv/conda/envs/notebook/lib/python3.6/site-packages (from requests-
>folium) (1.25.9)
Requirement already satisfied: MarkupSafe>=0.23 in /srv/conda/envs/note
book/lib/python3.6/site-packages (from Jinja2>=2.9->folium) (1.1.1)
Installing collected packages: branca, folium
Successfully installed branca-0.4.1 folium-0.11.0
Collecting geopy
```

```
Collecting geopy
  Downloading geopy-1.22.0-py2.py3-none-any.whl (113 kB)
    |████████████████████████████████████████| 113 kB 4.3 MB/s eta 0:00:01
Collecting geographiclib<2,>=1.49
  Downloading geographiclib-1.50-py3-none-any.whl (38 kB)
Installing collected packages: geographiclib, geopy
Successfully installed geographiclib-1.50 geopy-1.22.0
all libraries imported and packages installed
```

```
In [2]: Van_crimes_df = pd.read_csv('https://raw.githubusercontent.com/RamanujaSVL/Coursera_Capstone/master/vancouver_crime_records_2018.csv')
```

```
In [3]: Van_crimes_df.head()
```

Out[3]:

	Unnamed: 0	TYPE	YEAR	MONTH	DAY	HOUR	MINUTE	HUNDRED_BLOCK	NEIGHBOURHOOD
0	11	Break and Enter Commercial	2018	3	2	6	17	10XX ALBERNI ST	W
1	38	Break and Enter Commercial	2018	6	16	18	0	10XX ALBERNI ST	W
2	97	Break and Enter Commercial	2018	12	12	0	0	10XX BEACH AVE	W
3	110	Break and Enter Commercial	2018	4	9	6	0	10XX BEACH AVE	Central B
4	152	Break and Enter Commercial	2018	10	2	18	31	10XX BEACH AVE	Central B

```
In [4]: Van_crimes_df.drop(['Unnamed: 0', 'MONTH', 'DAY', 'HOUR', 'MINUTE', 'HUNDRED_BLOCK', 'X', 'Y'], axis=1, inplace=True)
```

```
In [5]: Van_crimes_df
```

```
Out[5]:
```

		TYPE	YEAR	NEIGHBOURHOOD
0		Break and Enter Commercial	2018	West End
1		Break and Enter Commercial	2018	West End
2		Break and Enter Commercial	2018	West End
3		Break and Enter Commercial	2018	Central Business District
4		Break and Enter Commercial	2018	Central Business District
...	
38072	Vehicle Collision or Pedestrian Struck (with I...		2018	Marpole
38073	Vehicle Collision or Pedestrian Struck (with I...		2018	Marpole
38074	Vehicle Collision or Pedestrian Struck (with I...		2018	Mount Pleasant
38075	Vehicle Collision or Pedestrian Struck (with I...		2018	Riley Park
38076	Vehicle Collision or Pedestrian Struck (with I...		2018	Mount Pleasant

38077 rows × 3 columns

```
In [6]: Van_crimes_df['NEIGHBOURHOOD'].value_counts()
```

```
Out[6]: Central Business District    10857
West End                            3031
Mount Pleasant                      2396
Strathcona                          1987
Kitsilano                           1802
Fairview                           1795
Renfrew-Collingwood                 1762
Grandview-Woodland                  1761
Kensington-Cedar Cottage            1391
Hastings-Sunrise                    1270
Sunset                              967
Riley Park                          866
Marpole                             828
```

```

Victoria-Fraserview      600
Killarney                565
Oakridge                 499
Dunbar-Southlands       474
Kerrisdale               417
Shaughnessy              414
West Point Grey          372
Arbutus Ridge            311
South Cambie              292
Stanley Park             154
Musqueam                  17
Name: NEIGHBOURHOOD, dtype: int64

```

```

In [6]: Van_crimes_df = pd.pivot_table(Van_crimes_df,
                                         values=['YEAR'],
                                         index=['NEIGHBOURHOOD'],
                                         columns=['TYPE'],
                                         aggfunc=len,
                                         fill_value=0,
                                         margins=True)

Van_crimes_df

```

Out[6]:

TYPE	YEAR							Veh Col or Pec Stru (wit Fat
	Break and Enter Commercial	Break and Enter Residential/Other	Mischief	Other Theft	Theft from Vehicle	Theft of Bicycle	Theft of Vehicle	
NEIGHBOURHOOD								
Arbutus Ridge	12	78	49	18	111	12	12	
Central Business District	551	124	1812	2034	5301	640	165	
Dunbar- Southlands	8	106	81	31	199	16	9	

YEAR									Veh Col or Pec Stri (wit Fat
TYPE	Break and Enter Commercial	Break and Enter Residential/Other	Mischief	Other Theft	Theft from Vehicle	Theft of Bicycle	Theft of Vehicle		
NEIGHBOURHOOD									
Fairview	138	73	233	297	692	245	55		
Grandview- Woodland	148	162	304	215	634	110	123		
Hastings-Sunrise	48	117	195	107	607	52	74		
Kensington-Cedar Cottage	62	145	255	148	541	69	71		
Kerrisdale	24	97	49	9	172	13	11		
Killarney	34	72	90	31	240	19	33		
Kitsilano	106	165	320	154	755	189	51		
Marpole	44	125	134	75	290	34	39		
Mount Pleasant	205	124	353	493	822	232	67		
Musqueam	0	4	3	0	4	2	2		
Oakridge	19	123	64	63	164	18	18		
Renfrew- Collingwood	91	156	243	472	569	37	92		
Riley Park	35	122	140	53	378	52	39		
Shaughnessy	12	120	41	0	187	10	11		
South Cambie	22	42	41	38	111	19	8		
Stanley Park	6	2	8	0	109	14	3		
Strathcona	160	124	527	81	821	108	76		
Sunset	37	93	175	105	382	18	63		

TYPE	YEAR							Veh Col or Pec Stru (wit Fat
	Break and Enter Commercial	Break and Enter Residential/Other	Mischief	Other Theft	Theft from Vehicle	Theft of Bicycle	Theft of Vehicle	
NEIGHBOURHOOD								
Victoria- Fraserview	15	80	94	57	239	15	36	
West End	230	72	460	455	1461	203	77	
West Point Grey	18	71	50	11	157	32	11	
All	2025	2397	5721	4947	14946	2159	1146	

In [9]: `Van_crimes_df.describe()`

Out[9]:

TYPE	YEAR						
	Break and Enter Commercial	Break and Enter Residential/Other	Mischief	Other Theft	Theft from Vehicle	Theft of Bicycle	The Vel
count	25.000000	25.000000	25.000000	25.000000	25.000000	25.000000	25.0
mean	162.000000	191.760000	457.680000	395.760000	1195.680000	172.720000	91.0
std	405.344915	461.397179	1153.040608	1032.710966	3043.72373	435.689159	223.5
min	0.000000	2.000000	3.000000	0.000000	4.000000	2.000000	2.0
25%	18.000000	73.000000	50.000000	31.000000	172.000000	16.000000	11.0
50%	37.000000	117.000000	140.000000	75.000000	378.000000	34.000000	39.0

YEAR							
TYPE	Break and Enter Commercial	Break and Enter Residential/Other	Mischief	Other Theft	Theft from Vehicle	Theft of Bicycle	Theft of Vehicle
75%	138.000000	124.000000	304.000000	215.000000	692.000000	110.000000	74.000000
max	2025.000000	2397.000000	5721.000000	4947.000000	14946.000000	2159.000000	1146.000000

```
In [7]: Van_crimes_df.reset_index(inplace=True)
Van_crimes_df.columns = Van_crimes_df.columns.map(''.join)
Van_crimes_df.rename(columns={'YEARAll': 'Total'}, inplace=True)
```

```
In [8]: Van_crimes_df.head()
```

Out[8]:

	NEIGHBOURHOOD	YEARBreak and Enter Commercial	YEARBreak and Enter Residential/Other	YEARMischief	YEAROther Theft	YEARTheft from Vehicle	YEATheft of Bicycle	YEATheft of Vehicle
0	Arbutus Ridge	12	78	49	18	111		
1	Central Business District	551	124	1812	2034	5301		
2	Dunbar-Southlands	8	106	81	31	199		
3	Fairview	138	73	233	297	692		
4	Grandview-Woodland	148	162	304	215	634		

```
In [9]: Van_crimes_df.sort_values(['Total'], ascending=False, axis=0, inplace=True)
```

```
Van_crimes_df_top5 = Van_crimes_df.iloc[1:6]
Van_crimes_df_top5
```

Out[9]:

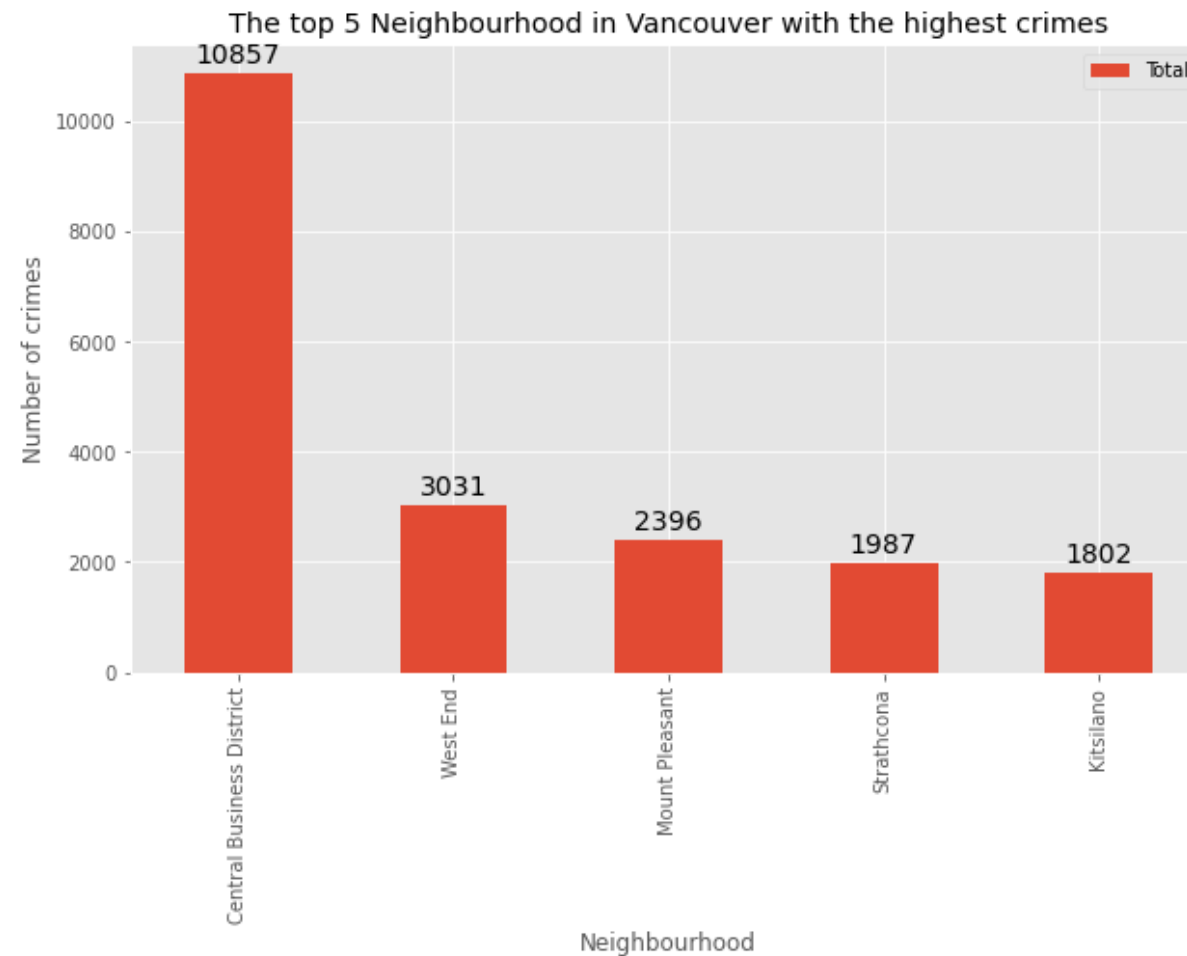
	NEIGHBOURHOOD	YEARBreak and Enter Commercial	YEARBreak and Enter Residential/Other	YEARMischief	YEAROther Theft	YEARTheft from Vehicle	YE of
1	Central Business District	551	124	1812	2034	5301	
22	West End	230	72	460	455	1461	
11	Mount Pleasant	205	124	353	493	822	
19	Strathcona	160	124	527	81	821	
9	Kitsilano	106	165	320	154	755	

```
In [10]: per_neighbourhood = Van_crimes_df_top5[['NEIGHBOURHOOD', 'Total']]
per_neighbourhood.set_index('NEIGHBOURHOOD', inplace=True)

ax = per_neighbourhood.plot(kind='bar',
                             figsize=(10,6))
ax.set_ylabel('Number of crimes')
ax.set_xlabel('Neighbourhood')
ax.set_title('The top 5 Neighbourhood in Vancouver with the highest crimes')

for p in ax.patches:
    ax.annotate(np.round(p.get_height(), decimals=2),
                (p.get_x()+p.get_width()/2., p.get_height()),
                ha='center',
                va='center',
                xytext=(0, 10),
                textcoords='offset points',
                fontsize = 14,
                )
```

```
plt.show()
```



```
In [11]: Van_crimes_df_low5 = Van_crimes_df.tail(5)
Van_crimes_df_low5
```

```
Out[11]:
```

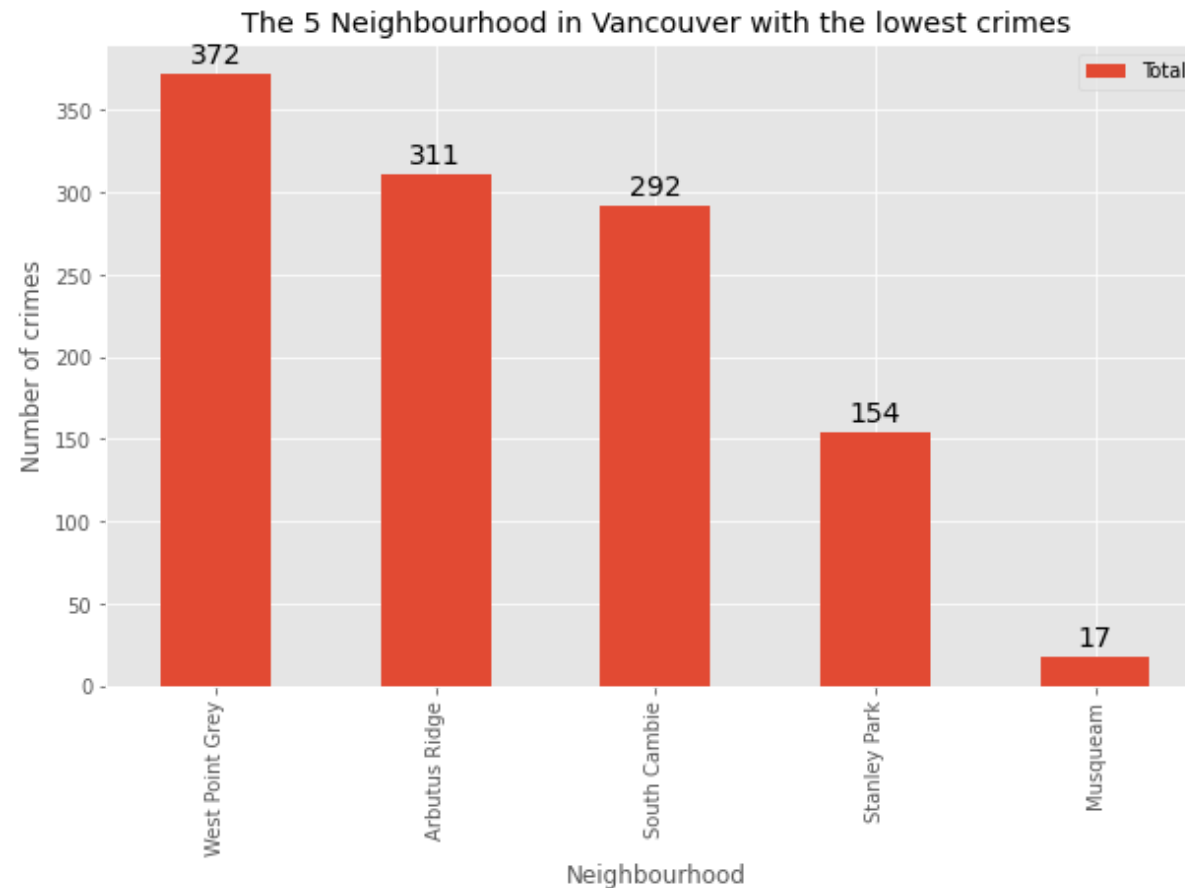
	NEIGHBOURHOOD	YEARBreak and Enter Commercial	YEARBreak and Enter Residential/Other	YEARMischief	YEAROther Theft	YEARTheft from Vehicle	YE of
23	West Point Grey	18	71	50	11	157	
0	Arbutus Ridge	12	78	49	18	111	
17	South Cambie	22	42	41	38	111	
18	Stanley Park	6	2	8	0	109	
12	Musqueam	0	4	3	0	4	

```
In [12]: per_neighbourhood = Van_crimes_df_low5[['NEIGHBOURHOOD', 'Total']]
per_neighbourhood.set_index('NEIGHBOURHOOD', inplace=True)

ax = per_neighbourhood.plot(kind='bar',
                             figsize=(10,6))
ax.set_ylabel('Number of crimes')
ax.set_xlabel('Neighbourhood')
ax.set_title('The 5 Neighbourhood in Vancouver with the lowest crimes')

for p in ax.patches:
    ax.annotate(np.round(p.get_height(), decimals=2),
                (p.get_x()+p.get_width()/2., p.get_height()),
                ha='center',
                va='center',
                xytext=(0, 10),
                textcoords='offset points',
                fontsize = 14,
                )

plt.show()
```



```
In [13]: Latitude = []
Longitude = []
Neighbourhood = Van_crimes_df['NEIGHBOURHOOD'].unique()

key = '830323b5ca694362904814ff0a11b803'
geocoder = OpenCageGeocode(key)

for i in range(len(Neighbourhood)):
    address = '{} , Vancouver, BC, Canada'.format(Neighbourhood[i])
    location = geocoder.geocode(address)
    Latitude.append(location[0]['geometry']['lat'])
```

```

Longitude.append(location[0]['geometry']['lng'])

print(Latitude, Longitude)

[49.2807848, 49.24966, 49.2841308, 49.2633296, 49.279554, 49.2694099, 4
9.2641128, 49.2420242, 49.2705588, 49.2476321, 49.2775935, 49.2195929,
49.2474381, 49.2092233, 49.2184156, 49.2242738, 49.2308288, 49.2534601,
49.2346728, 49.2518626, 49.2644843, 49.2409677, 49.2466847, 49.3019112,
49.2346005] [-123.124856, -123.11934, -123.1317949, -123.0965885, -123.
0899788, -123.155267, -123.1268352, -123.0576794, -123.0679417, -123.08
42067, -123.0439199, -123.0902386, -123.1029664, -123.1361495, -123.073
2871, -123.0462504, -123.1311342, -123.1850439, -123.1553893, -123.1380
226, -123.1854326, -123.1670008, -123.120915, -123.1415405, -123.18339
7]

```

```

In [14]: neighbourhood_dict = {'Neighbourhood': Neighbourhood, 'Latitude': Latitude, 'Longitude': Longitude}
neighbourhood_geo = pd.DataFrame(data=neighbourhood_dict, columns=['Neighbourhood', 'Latitude', 'Longitude'], index=None)
neighbourhood_geo

```

Out[14]:

	Neighbourhood	Latitude	Longitude
0	All	49.280785	-123.124856
1	Central Business District	49.249660	-123.119340
2	West End	49.284131	-123.131795
3	Mount Pleasant	49.263330	-123.096588
4	Strathcona	49.279554	-123.089979
5	Kitsilano	49.269410	-123.155267
6	Fairview	49.264113	-123.126835
7	Renfrew-Collingwood	49.242024	-123.057679
8	Grandview-Woodland	49.270559	-123.067942
9	Kensington-Cedar Cottage	49.247632	-123.084207

	Neighbourhood	Latitude	Longitude
10	Hastings-Sunrise	49.277594	-123.043920
11	Sunset	49.219593	-123.090239
12	Riley Park	49.247438	-123.102966
13	Marpole	49.209223	-123.136150
14	Victoria-Fraserview	49.218416	-123.073287
15	Killarney	49.224274	-123.046250
16	Oakridge	49.230829	-123.131134
17	Dunbar-Southlands	49.253460	-123.185044
18	Kerrisdale	49.234673	-123.155389
19	Shaughnessy	49.251863	-123.138023
20	West Point Grey	49.264484	-123.185433
21	Arbutus Ridge	49.240968	-123.167001
22	South Cambie	49.246685	-123.120915
23	Stanley Park	49.301911	-123.141541
24	Musqueam	49.234600	-123.183397

```
In [15]: address = 'Vancouver, BC, Canada'

location = geocoder.geocode(address)
latitude = location[0]['geometry']['lat']
longitude = location[0]['geometry']['lng']

print('The geographical coordinate of Vancouver, Canada are {}, {}'.format(latitude, longitude))
```

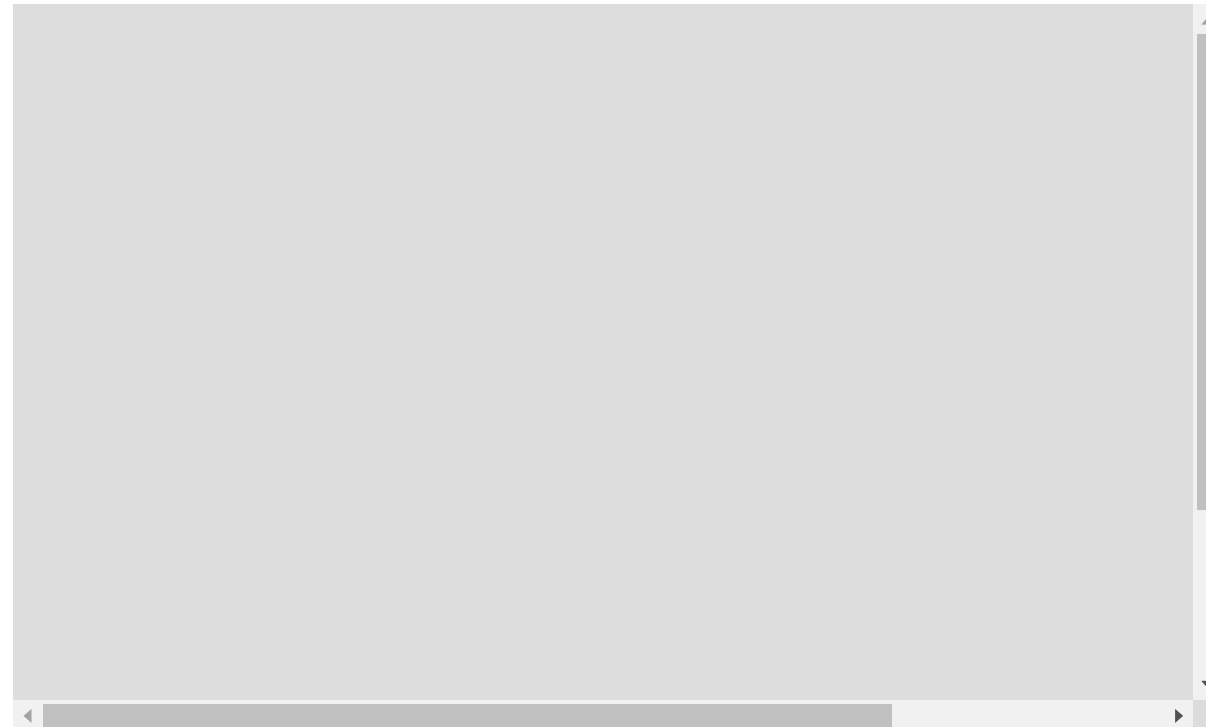
The geographical coordinate of Vancouver, Canada are 49.2608724, -123.1139529.

```
In [16]: Van_map = folium.Map(location=[latitude, longitude], zoom_start=12)
```

```
for lat, lng, Neighbourhood in zip(neighbourhood_geo['Latitude'], neighbourhood_geo['Longitude'], neighbourhood_geo['Neighbourhood']):
    label = '{}'.format(Neighbourhood)
    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(Van_map)
```

Van_map

Out[16]:




```
In [17]: kclusters = 5
Van_clustered = Van_crimes_df.drop('NEIGHBOURHOOD', 1)
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(Van_clustered)
kmeans.labels_[0:10]
neighbourhood_geo.insert(0, 'Cluster Labels', kmeans.labels_)
```

```
In [18]: vancouver_merged = neighbourhood_geo
vancouver_merged.head()
```

Out[18]:

	Cluster Labels	Neighbourhood	Latitude	Longitude
0	1	All	49.280785	-123.124856
1	2	Central Business District	49.249660	-123.119340
2	3	West End	49.284131	-123.131795
3	4	Mount Pleasant	49.263330	-123.096588
4	4	Strathcona	49.279554	-123.089979

```
In [19]: map_clusters = folium.Map(location=[latitude, longitude], zoom_start=12)

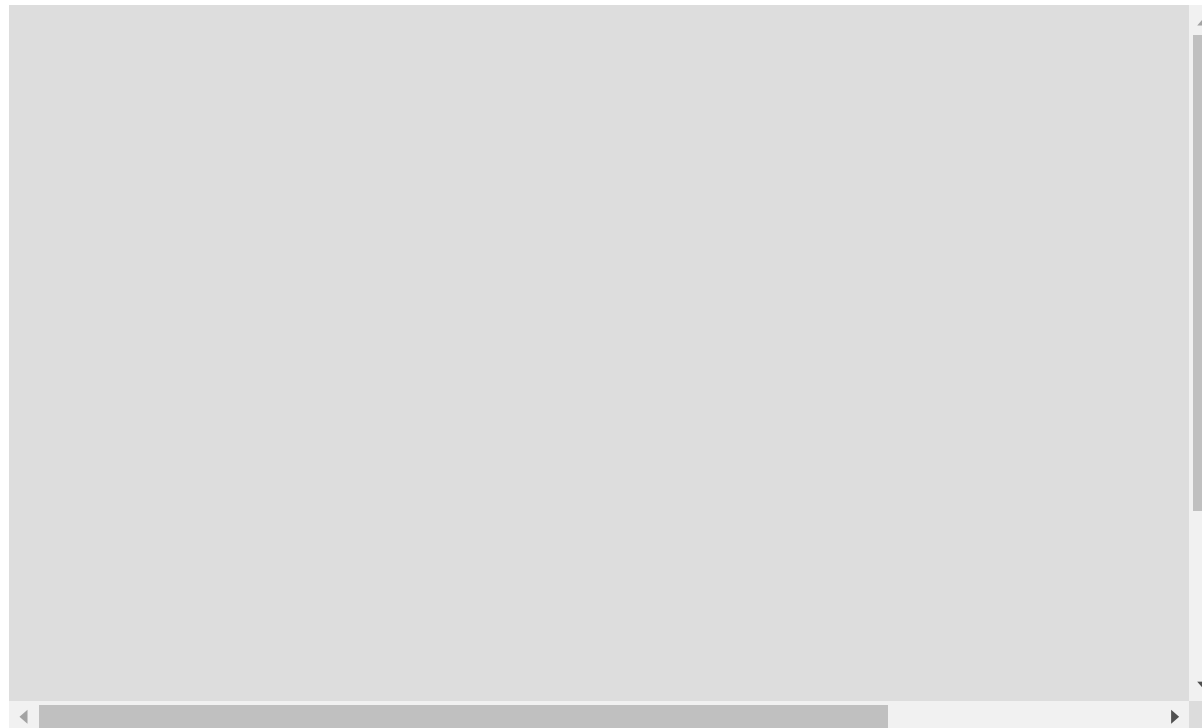
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0,1,len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

markers_colors = []
for lat, lon, poi, cluster in zip(vancouver_merged['Latitude'], vancouver_merged['Longitude'], vancouver_merged['Neighbourhood'], vancouver_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
```

```
color=rainbow[cluster-1],  
fill=True,  
fill_color=rainbow[cluster-1],  
fill_opacity=0.7).add_to(map_clusters)
```

map_clusters

Out[19]:



Results

Attending the goal of this project, we have been analysing the safest neighbourhood in Vancouver, our stakeholders are specially investor to open a new business or a family looking for a safe place to move in.

Conclusion

We get through some analyses comparing which neighbourhood are the safest and which are the most dangerous, we also check by the kind of crime.

In []: