

# Introduction: Business Problem

---

The aim of this project is to find a safe and secure location for opening of commercial establishments in Vancouver, Canada. Specifically, this report will be targeted to stakeholders interested in opening any business place like Grocery Store in Vancouver City, Canada.

The first task would be to choose the safest borough by analysing crime data for opening a grocery store and short listing a neighbourhood, where grocery stores are not amongst the most common venues, and yet as close to the city as possible.

We will make use of our data science tools to analyse data and focus on the safest borough and explore its neighborhoods and the 10 most common venues in each neighborhood so that the best neighborhood where grocery store is not amongst the most common venue can be selected.

---

## Data

---

Based on definition of our problem, factors that will influence our decision are:

- finding the safest borough based on crime statistics
- finding the most common venues
- choosing the right neighbourhood within the borough

We will be using the geographical coordinates of Vancouver to plot neighbourhoods in a borough that is safe and in the city's vicinity, and finally cluster our neighborhoods and present our findings.

Following data sources will be needed to extract/generate the required information:

- **Part 1:** [Using a real world data set from Kaggle containing the Vancouver Crimes from 2003 to 2019](#): A dataset consisting of the crime statistics of each Neighbourhood in Vancouver along with type of crime, recorded year, month and hour.
- **Part 2:** [Gathering additional information of the list of officially categorized boroughs in Vancouver from Wikipedia](#): Borough information will be used to map the existing data where each neighbourhood can be assigned with the right borough.
- **Part 3:** [Creating a new consolidated dataset of the Neighborhoods, along with their boroughs, crime data and the respective Neighbourhood's co-ordinates](#): This data will be fetched using OpenCage Geocoder to find the safest borough and explore the neighbourhood by plotting it on maps using Folium and perform exploratory data analysis.
- **Part 4:** [Creating a new consolidated dataset of the Neighborhoods, boroughs, and the most common venues and the respective Neighbourhood along with co-ordinates](#): This data will be fetched using Four Square API to explore the

neighbourhood venues and to apply machine learning algorithm to cluster the neighbourhoods and present the findings by plotting it on maps using Folium.

---

## **Part 1:** Using a real world data set from Kaggle containing the Vancouver Crimes from 2003 to 2019

### Vancouver Crime Report

#### Properties of the Crime Report

- 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>

The top screenshot shows the following code and output:

```
In [11]: vnc_crime_df['Neighbourhood'].value_counts()
Out[11]: 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 [5]: dfToronto = pd.DataFrame(index=Neighbourhoods, columns=["Population_2016", "Income_2016"])
dfToronto.head()
Out[5]:
```

	Population_2016	Income_2016
Aginicourt North	NaN	NaN
Aginicourt South-Malvern West	NaN	NaN

The bottom screenshot shows the following code and output:

```
In [9]: vnc_crime_df = pd.read_csv('https://raw.githubusercontent.com/RamanujaSVL/Coursera_Capstone/master/vancouver_crime_records_2018.')
#Dropping X,Y which represents Lat, Lng data as Coordinates, the data seems to be corrupt
vnc_crime_df.drop(['Unnamed: 0', 'MINUTE', 'HUNDRED_BLOCK', 'X', 'Y'], axis = 1, inplace = True)
#vnc_crime_df.columns
vnc_crime_df.head()
Out[9]:
```

	TYPE	YEAR	MONTH	DAY	HOUR	NEIGHBOURHOOD
0	Break and Enter Commercial	2018	3	2	6	West End
1	Break and Enter Commercial	2018	6	16	18	West End
2	Break and Enter Commercial	2018	12	12	0	West End
3	Break and Enter Commercial	2018	4	9	6	Central Business District
4	Break and Enter Commercial	2018	10	2	18	Central Business District

```
In [10]: vnc_crime_df.columns = ['Type', 'Year', 'Month', 'Day', 'Hour', 'Neighbourhood']
vnc_crime_df.head()
Out[10]:
```

	Type	Year	Month	Day	Hour	Neighbourhood
0	Break and Enter Commercial	2018	3	2	6	West End
1	Break and Enter Commercial	2018	6	16	18	West End
2	Break and Enter Commercial	2018	12	12	0	West End
3	Break and Enter Commercial	2018	4	9	6	Central Business District
4	Break and Enter Commercial	2018	10	2	18	Central Business District

```
In [11]: vnc_crime_df['Neighbourhood'].value_counts()
Out[11]: Central Business District    10857
```

## Part 2: Gathering additional information about the Neighborhood from Wikipedia

As part of data set Borough which the neighborhood was part of was not categorized, so we will create a dictionary of Neighborhood and based on data in the following [Wikipedia page](#).

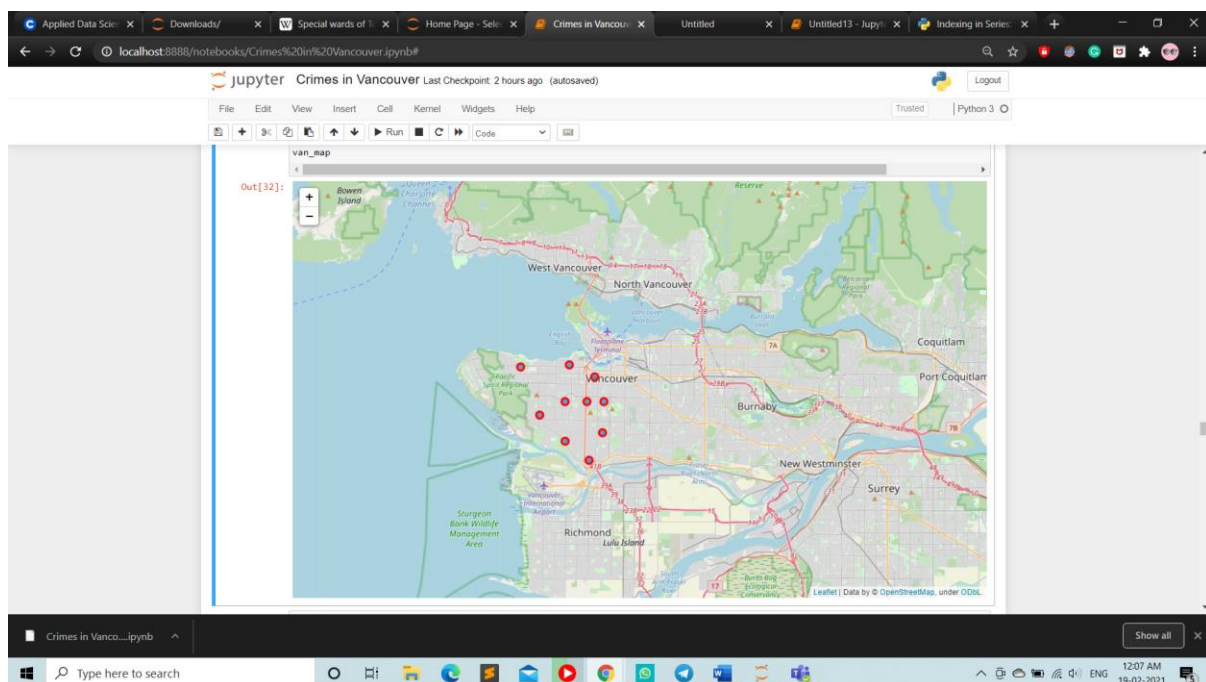
- Methodology

Categorized the methodologysection into two parts:

1. **Exploratory Data Analysis:** Visualise the crime reports in different Vancouver boroughs to identify the safest borough and normalise the neighborhoods of that borough. We will use the resulting data and find 10 most common venues in each neighborhood.
2. **Modelling:** To help stakeholders choose the right neighborhood within a borough we will be clustering similar neighborhoods using K - means clustering which is a form of unsupervised machine learning algorithm that clusters data based on predefined cluster size. We will use K-Means clustering to address this problem so as to group data based on existing venues which will help in the decision making process.

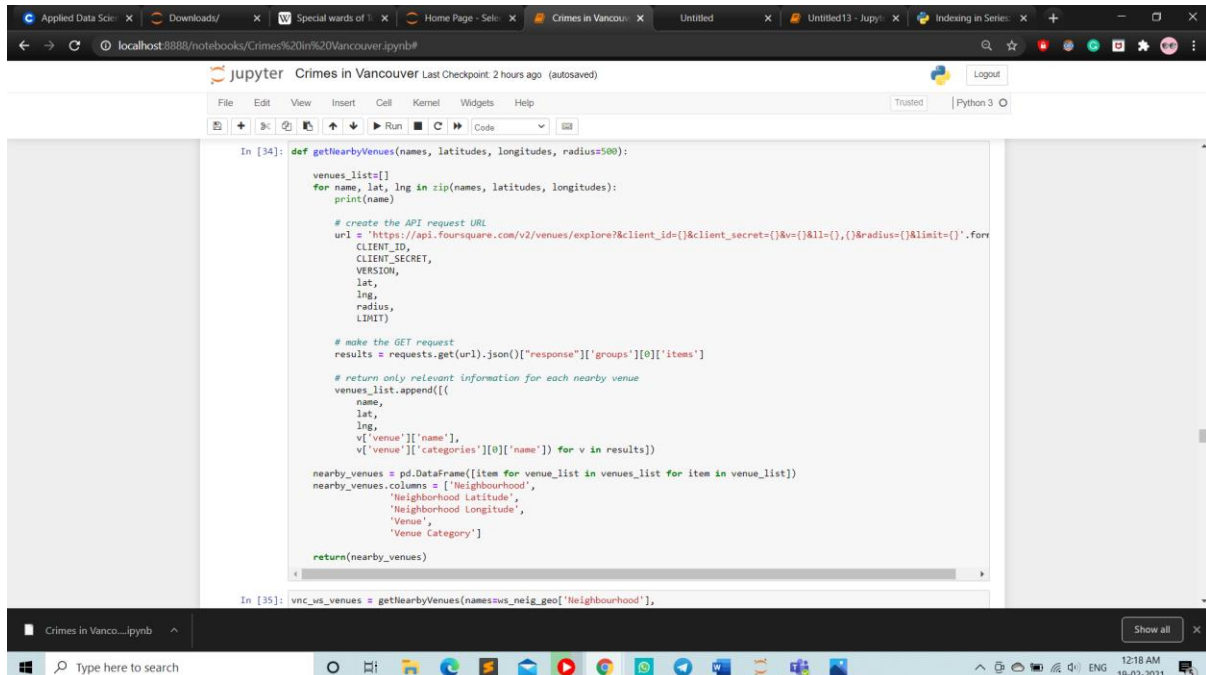
**Part 3:** Creating a new consolidated dataset of the Neighborhoods, along with their boroughs, crime data and the respective Neighbourhood's co-ordinates.:

This data will be fetched using OpenCage Geocoder to find the safest borough and explore the neighbourhood by plotting it on maps using Folium and perform exploratory data analysis.



**Part 4:** Creating a new consolidated dataset of the Neighborhoods, boroughs, and the most common venues and the respective Neighbourhood along with co-ordinates.:

This data will be fetched using Four Square API to explore the neighbourhood venues and to apply machine learning algorithm to cluster the neighbourhoods and present the findings by plotting it on maps using Folium.



```
In [34]: def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&lat={}&lng={}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]["groups"][0]["items"]

        # return only relevant information for each nearby venue
        venues_list.append([
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['categories'][0]['name'] for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighbourhood',
                              'Neighborhood Latitude',
                              'Neighborhood Longitude',
                              'Venue',
                              'Venue Category']

    return(nearby_venues)

In [35]: vnc_us_venues = getNearbyVenues(names=us_neig['Neighbourhood'],
```

## Results and Discussion

The objective of the business problem was to help stakeholders identify one of the safest borough in Vancouver, and an appropriate neighborhood within the borough to set up a commercial establishment especially a Grocery store. This has been achieved by first making use of Vancouver crime data to identify a safe borough with considerable number of neighborhood for any business to be viable. After selecting the borough it was imperative to choose the right neighborhood where grocery shops were not among venues in a close proximity to each other. We achieved this by grouping the neighborhoods into clusters to assist the stakeholders by providing them with relevant data about venues and safety of a given neighborhood.

## Conclusion

We have explored the crime data to understand different types of crimes in all neighborhoods of Vancouver and later categorized them into different boroughs, this helped us group the neighborhoods into boroughs and choose the safest borough

first. Once we confirmed the borough the number of neighborhoods for consideration also comes down, we further shortlist the neighborhoods based on the common venues, to choose a neighborhood which best suits the business problem.