

Introduction

The primary objective of this project was to process and perform thorough analysis on data obtained from multiple sources. This will be done in order to make educated and data-driven conclusions on a specific aspect of the Greater Toronto Area.

The business problem chosen was to find the best location in the GTA to open a bicycle store based on a few specific criteria. The measures that would be observed to determine the desirability of a location are as follows; **the number of recorded bike thefts in a neighborhood, the number of already existing bike stores in a neighborhood, and the average price of bikes being stolen in a neighborhood.**

The reasoning for choosing the aforementioned criteria as the primary factors when selecting a location stems from our team's rationale, which is as follows; the *more* thefts in an area is better as people will require new bikes more often, the *lower* number of already existing stores is better as that would entail less business competition, and a *higher* average bike cost is better as it is an indication that people would be willing to pay a higher price when purchasing a new bicycle.

Explanation of Data and Sources

The sources being utilized to obtain the necessary data for this project comes from the **Foursquare API** and the **city of Toronto website**.

The Foursquare API would provide the necessary data regarding already existing bicycle stores. More specifically, the Foursquare 'Places' API returns location data in an easily accessible and highly scalable manner. This API provides thorough geolocation data, allowing users to draw rich details about venues within a range of locations. The API requires the creation of a developer account for use, which afterwards a user may access an endpoint to communicate with. The user passes numerous parameters to the endpoint in order to specify the data desired, such as coordinates, radius, venue category id, and so on.

The venue category id chosen to retrieve data from was **17119**. This venue id pertains to the **Retail > Sporting Goods Retail > Bicycle Store** category. The base latitude and longitude passed was near the center of the city of Toronto, with a 15000 meter radius. This radius would cover a majority of Toronto, whilst not crossing into the Mississauga, Vaughan, Markham, or Pickering regions. Shown below are images displaying the selection of a radius and the code used to make a request to the API endpoint.

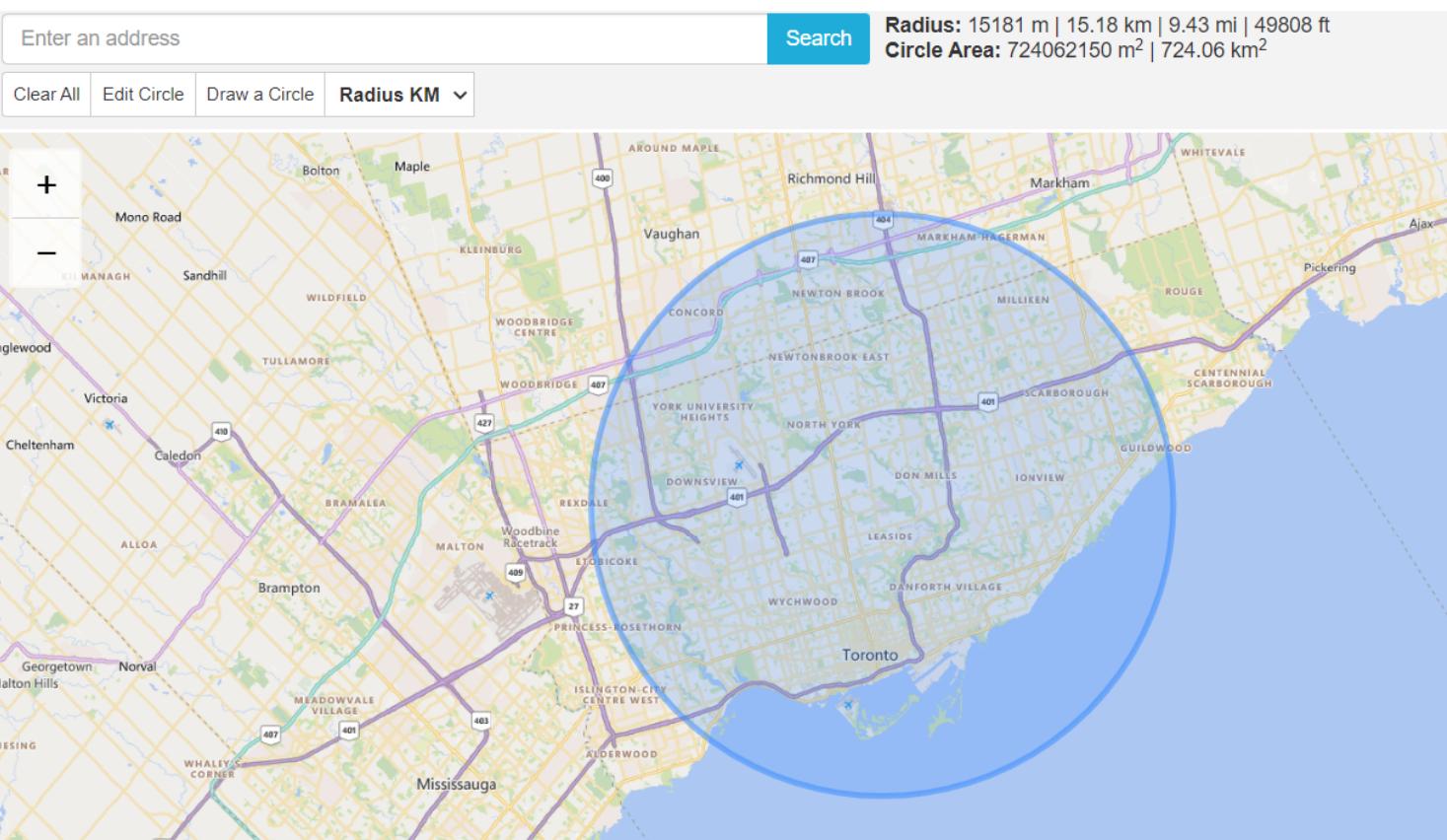


Figure 1

```
from textwrap import indent
import requests
import json

url = "https://api.foursquare.com/v3/places/search?ll=43.69%2C-79.33&radius=15000&categories=17119&limit=50"

headers = {
    "Accept": "application/json",
    "Authorization": "fsq37aiTLwAGMwc0LBG0ob2Sb22qNqfeOz01G7w624eeVfc="
}

response = requests.request("GET", url, headers=headers)

#print(response.text)
data = json.loads(response.text)

with open('FoursquareData.json', 'w', encoding='utf-8') as f:
    json.dump(data, f, ensure_ascii=False, indent=4)
```

Figure 2

Due to the limitations of the API endpoint, the maximum number of bicycle stores that could be returned was 50. As a result, this imposes a constraint on the data our team will be working with, as the number of points will be limited by the API call.

The city of Toronto website contains an **open data portal** which provides access to a large collection of data catalogs. The city of Toronto website provided the necessary data regarding bicycle theft, which included details such as the neighborhood the bike was stolen in, the cost of the bike, date information, and so on. This data was provided as a **CSV file**, titled **bicycle_thefts.csv** in the project folder.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	Id	Event
	Bike_Colo	Bike_Make	Bike_Modl	Bike_Spee	Bike_Type	City	Cost_of_B_Division	Hood_ID	Location_Neighbour	OBJECTID	ObjectID2	Occurrence_0	Occurrence_1	Occurrence_2	Occurrence_3	Occurrence_4	Occurrence_5	Occurrence_6	Occurrence_7	Occurrence_8	Premises	Primary_O_Report_Dx	Report_Dx	Report_Hc	Report_Mi	Report_Status	.Id	Event				
1	Bike	GI	ESCAPE	2	7 OT	Toronto	700 D22	15 Streets, Rc Kingsway S	17744	1	2017-10-0	3 Tuesday	276	14 October	2017 Outside	THEFT UN	2017-10-0	3 Tuesday	276	18 October	2017	STOLEN	1	GI	2017 RECOVERE	2 G						
2	BLK	GI	UNKNOWN MAKE	1	TO	Toronto	1100 D22	15 Single Hon Kingsway S	17759	2	2017-11-0	8 Wednesday	312	3 November	2017 House	THEFT UN	2017-11-0	8 Wednesday	312	22 November	2017	RECOVERE	2	GI	2017 RECOVERE	2 G						
3	BLK	OT	CROSSTRA	24	MT	Toronto	904 D22	15 Ttc Subwa Kingsway S	17906	3	2018-09-1	14 Friday	257	9 September	2018 Transit	THEFT UN	2018-09-1	17 Monday	260	16 September	2018	STOLEN	3	GI	2018 STOLEN	3 G						
4	BLK	RED	GI	6	MT	Toronto	600 D22	15 Ttc Subwa Kingsway S	17962	4	2015-05-0	7 Thursday	127	18 May	2015 Transit	THEFT UN	2015-05-1	14 Thursday	134	14 May	2015	STOLEN	4	GI	2015 STOLEN	4 G						
5	BLK	GR	GT	10	TO	Toronto	400 D22	15 Ttc Subwa Kingsway S	17963	5	2015-05-1	16 Saturday	136	12 May	2015 Transit	THEFT UN	2015-05-1	16 Saturday	136	15 May	2015	STOLEN	5	GI	2015 STOLEN	5 G						
6	TRQ	OTHER	UNK	18	RC	Toronto	100 D22	15 Parking Lo Kingsway S	18003	6	2015-09-0	1 Tuesday	244	0 September	2015 Outside	THEFT UN	2015-10-2	29 Thursday	302	18 October	2015	STOLEN	6	GI	2015 STOLEN	6 G						
7	WHD	NO	SASQUATC	18	MT	Toronto	0 D22	15 Ttc Subwa Kingsway S	18013	7	2016-02-1	18 Thursday	49	9 February	2016 Transit	THEFT UN	2016-02-1	18 Thursday	49	23 February	2016	STOLEN	7	GI	2016 STOLEN	7 G						
8	BLK	GARY FISHER	7	OT	Toronto	800 D22	15 Ttc Bus Kingsway S	19993	8	2019-05-0	1 Wednesday	121	13 May	2019 Transit	THEFT UN	2019-05-0	6 Monday	126	11 May	2019	STOLEN	8	GI	2019 STOLEN	8 G							
9	BLK	UK	UK	20	RG	Toronto	700 D22	15 Go Bus Kingsway S	21209	9	2017-09-0	6 Wednesday	249	8 September	2017 Transit	THEFT UN	2017-09-0	6 Wednesday	249	21 September	2017	STOLEN	9	GI	2017 STOLEN	9 G						
10	ONG	UK	HARDROCI	24	MT	Toronto	550 D22	15 Streets, Rc Kingsway S	21999	10	2014-05-0	7 Wednesday	127	18 May	2014 Outside	THEFT UN	2014-05-0	7 Wednesday	127	19 May	2014	STOLEN	10	GI	2014 STOLEN	10 G						
11	BLU	OTHE	GI	16	RC	Toronto	600 D22	15 Single Hon Kingsway S	22015	11	2014-06-2	21 Saturday	172	12 June	2014 House	B&E	2014-06-2	23 Monday	174	10 June	2014	STOLEN	11	GI	2014 STOLEN	11 G						
12	BLU	BLK	GI	50	RC	Toronto	829 D22	15 Open Area Kingsway S	22026	12	2014-07-2	24 Thursday	205	9 July	2014 Outside	THEFT UN	2014-07-2	28 Monday	209	11 July	2014	STOLEN	12	GI	2014 STOLEN	12 G						
13	BLK	GI	GI	50	RC	Toronto	2213	15 Parking Lo Kingsway S	22133	13	2016-10-1	14 Friday	288	13 October	2016 Outside	THEFT UN	2016-10-1	17 Monday	291	11 October	2016	STOLEN	13	GI	2016 STOLEN	13 G						
14	BLK	SPECIALIZI	HARDROCI	27	MT	Toronto	690 D22	15 Private Prc Kingsway S	22134	14	2016-11-0	2 Wednesday	307	1 November	2016 Other	B&E/WIN'	2016-11-0	2 Wednesday	307	9 November	2016	STOLEN	14	GI	2016 STOLEN	14 G						
15	BLU	TREK	7000	18	MT	Toronto	1000 D22	15 Private Prc Kingsway S	22135	15	2016-11-0	2 Wednesday	307	1 November	2016 Other	B&E/WIN'	2016-11-0	2 Wednesday	307	9 November	2016	STOLEN	15	GI	2016 STOLEN	15 G						
16	BLKWHI	OTHER	GI	18	MT	Toronto	1000 D22	15 Ttc Subwa Kingsway S	22136	16	2016-11-1	18 Friday	323	8 November	2016 Transit	THEFT UN	2016-11-1	18 Friday	323	22 November	2016	RECOVERE	16	GI	2016 RECOVERE	16 G						
17	MRN	RA	24	MT	Toronto	1100 D22	15 Single Hon Kingsway S	22817	17	2020-08-0	3 Monday	216	20 August	2020 House	B&E/WIN'	2020-08-0	3 Monday	216	20 August	2020	STOLEN	17	GI	2020 STOLEN	17 G							
18	BLKBLLU	SPECIALIZED	21	MT	Toronto	500 D22	15 Single Hon Kingsway S	22818	18	2020-08-0	3 Monday	216	20 August	2020 House	B&E/WIN'	2020-08-0	3 Monday	216	20 August	2020	STOLEN	18	GI	2020 STOLEN	18 G							
19	BLKBLLU	SPECIALIZED	21	MT	Toronto	500 D22	15 Single Hon Kingsway S	22819	19	2018-07-1	13 Friday	194	19 July	2018 House	B&E/WIN'	2018-07-1	14 Saturday	195	9 July	2018	STOLEN	19	GI	2018 STOLEN	19 G							
20	GRY	TREK	18	MT	Toronto	1200 D22	15 Streets, Rc Kingsway S	22826	20	2020-08-2	25 Tuesday	238	14 August	2020 Outside	THEFT UN	2020-08-2	25 Tuesday	238	14 August	2020	STOLEN	20	GI	2020 STOLEN	20 G							
21	BLK	UK	70	MT	Toronto	115 D22	15 Streets, Rc Kingsway S	22826	21	2017-09-0	6 Wednesday	249	8 September	2017 Transit	THEFT UN	2017-09-0	6 Wednesday	249	21 September	2017	STOLEN	21	GI	2017 STOLEN	21 G							
22	BLK	UK	20	RG	Toronto	700 D22	15 Go Bus Kingsway S	24225	21	2017-09-0	6 Wednesday	249	8 September	2017 Transit	THEFT UN	2017-09-0	6 Wednesday	249	21 September	2017	STOLEN	22	GI	2017 STOLEN	22 G							
23	GRY	RA	18	MT	Toronto	200 D22	15 Ttc Subwa Kingsway S	24347	22	2018-07-2	27 Friday	208	10 July	2018 Transit	THEFT UN	2018-07-2	28 Saturday	209	20 July	2018	STOLEN	23	GI	2018 STOLEN	23 G							
24	SIL	OT	SPECIALIZI	24	OT	Toronto	600 D22	15 Other Non Kingsway S	24479	23	2015-08-0	8 Saturday	220	14 August	2015 Other	THEFT UN	2015-08-0	9 Sunday	221	20 August	2015	STOLEN	24	GI	2015 STOLEN	24 G						
25	GRY	OT	2013	8	RG	Toronto	600 D22	15 Schools Di Kingsway S	24515	24	2016-05-0	5 Thursday	126	10 May	2016 Education	THEFT UN	2016-05-1	10 Tuesday	131	18 May	2016	STOLEN	25	GI	2016 STOLEN	25 G						
26	GRN	RALEIGH	3	RG	Toronto	400 D22	15 Ttc Subwa Kingsway S	24531	25	2016-07-2	21 Thursday	203	8 July	2016 Transit	THEFT UN	2016-07-2	25 Monday	207	21 July	2016	STOLEN	26	GI	2016 STOLEN	26 G							
27	YEL	GF	3	MT	Toronto	600 D22	16 Apartment Stonegate	60	26	2017-01-1	10 Tuesday	10	11 January	2017 Apartment	THEFT UN	2017-01-2	23 Monday	23	13 January	2017	STOLEN	27	GI	2017 STOLEN	27 G							
28	BLK	SPECIALIZI	DIVERGE	18	OT	Toronto	1600 D22	16 Other Con Stonegate	114	27	2017-02-2	23 Thursday	54	18 February	2017 Commercial	THEFT UN	2017-02-2	27 Monday	58	20 February	2017	STOLEN	28	GI	2017 STOLEN	28 G						
29	GRY	OT	WE THE PE	1	BM	Toronto	250 D22	16 Streets, Rc Stonegate	1438	28	2017-09-1	16 Saturday	259	23 September	2017 Outside	THEFT UN	2017-09-1	17 Sunday	260	18 September	2017	STOLEN	29	GI	2017 STOLEN	29 G						
30	GRY	RA	REDUX 1	20	RG	Toronto	600 D22	16 Single Hon Stonegate	1529	29	2017-09-2	26 Tuesday	269	3 September	2017 House	THEFT UN	2017-09-2	26 Tuesday	269	12 September	2017	STOLEN	30	GI	2017 STOLEN	30 G						
31	BLKWHI	VILANO	21	OT	Toronto	886 D22	16 Single Hon Stonegate	1664	30	2017-10-1	13 Friday	286	0 October	2017 House	THEFT UN	2017-10-1	13 Friday	286	13 October	2017	STOLEN	31	GI	2017 STOLEN	31 G							
32	BLU	CC	18	MT	Toronto	400 D22	16 Single Hon Stonegate	1960	31	2017-11-1	11 Saturday	315	22 November	2017 House	THEFT UN	2017-12-2	28 Thursday	362	11 December	2017	STOLEN	32	GI	2017 STOLEN	32 G							
33	GRY	OT	S TYPE	30	MT	Toronto	4500 D22	16 Private Prc Stonegate	3498	32	2018-09-1	17 Monday	260	8 September	2018 Other	THEFT UN	2018-09-1	17 Monday	260	23 September	2018	STOLEN	33	GI	2018 STOLEN	33 G						
34	GRY	SPECIALIZI	ROCK JUM	12	MT	Toronto	D22	16 Single Hon Stonegate	4104	33	2019-03-2	21 Thursday	80	17 March	2019 House	B&E	2019-03-2	23 Saturday	82	21 March	2019	STOLEN	34	GI	2019 STOLEN	34 G						
35	BRZ	GIGANT	SLAMMER	1	BM	Toronto	380 D22	16 Schools Di Stonegate	4252	34	2019-05-0	8 Wednesday	128	9 May	2019 Education	THEFT UN	2019-05-0	8 Wednesday	128	19 May	2019	STOLEN	35	GI	2019 STOLEN	35 G						
36	BLK	UNKNOWN MAKE	10	RG	Toronto	1000 D22	16 Apartment Stonegate	4554	35	2019-06-2	20 Thursday	171	3 June	2019 Apartment	THEFT FRC	2019-06-2	20 Thursday	171	15 June	2019	STOLEN	36	GI	2019 STOLEN	36 G							
37	RED	SPECIALIZI	LANGSTER	1	OT	Toronto	903 D22	16 Streets, Rc Stonegate	4686	36	2019-07-0	4 Thursday	185	13 July	2019 Outside	THEFT UN	2019-07-0	4 Thursday	185	17 July	2019	STOLEN	37	GI	2019 STOLEN	37 G						
38	TRQ	OT	10	TO	Toronto	0 D22	16 Private Prc Stonegate	5319	37	2019-09-0	1 Sunday	244	20 September	2019 Other	THEFT UN	2019-09-1	13 Friday	256	23 September	2019	STOLEN	38	GI	2019 STOLEN	38 G							
39	TRQ	OT	10	OT	Toronto	0 D22	16 Private Prc Stonegate	5320	38	2019-09-0	1 Sunday	244	20 September	2019 Other	THEFT UN	2019-09-1	13 Friday	256	23 September	2019	STOLEN	40	GI	2019 STOLEN	40 G							
40	BLK	OT	21	OT	Toronto	879 D22	16 Private Prc Stonegate	6165	39	2020-05-1	11 Monday	132	23 May	2020 Other	THEFT UN	2020-05-1	12 Tuesday	133	8 May	2020	STOLEN	41	GI	2020 STOLEN	41 G							
41	BLU	CA	SL4 (M)	18	MT	Toronto	632 D22	16 Bar / Rest Stonegate	6738	40	2020-07-2	24 Friday	206	22 July	2020 Commercial	THEFT UN	2020-07-2	26 Sunday	208	8 July	2020	STOLEN	42	GI	2020 STOLEN	42 G						
42	BLK	OT	21	MT	Toronto	459 D22	16 Private Prc Stonegate	6936	41	2020-08-1	10 Monday	223	23 August	2020 Other	THEFT UN	2020-08-1	11 Tuesday	224	13 August	2020	STOLEN	43	GI	2020 STOLEN	43 G							
43	BLK	GI	CYPRESS	32	MT	Toronto	300 D22	16 Single Hon Stonegate	7120	42	2020-08-2	29 Saturday	242	13 August	2020 House	THEFT UN	2020-08-2	29 Saturday	242	16 August	2020	STOLEN	44	GI	2020 STOLEN	44 G						
44	GRYLGR	NORTHR0 CTM	21	MT	Toronto	400 D22	16 Parking Lo Stonegate	7620	43	2020-09-2	29 Tuesday	273	20 September	2020 Outside	THEFT UN	2020-11-1	12 Thursday	317	10 November	2020	STOLEN	45										

```
    "main": {
        "latitude": 43.678605,
        "longitude": -79.298545
    },
    "roof": {
        "latitude": 43.678605,
        "longitude": -79.298545
    }
},
"location": {
    "address": "615 Kingston Rd",
    "country": "CA",
    "cross_street": "at Main St",
    "formatted_address": "615 Kingston Rd (at Main St), Toronto ON M4E 1R3",
    "locality": "Toronto",
    "neighborhood": [
        "East York"
    ],
    "postcode": "M4E 1R3",
    "region": "ON"
},
"name": "Cycle Solutions",
"related_places": {},
"timezone": "America/Toronto"
```

Figure 4

As seen above, each bicycle shop has an associated latitude and longitude, which will be used to obtain further neighborhood information about each location. Whilst each bicycle shop has a neighborhood name, they do not have an associated neighborhood id, which is essential to merge the shop data with the bicycle theft data by location. Opposed to doing this manually, our team decided to use already existing geojson data to help group bicycle shops into neighborhoods using their longitudes and latitudes.

Our team found a geojson file online, titled **Neighborhoods.json**, which already had the Greater Toronto Area partitioned into polygonal regions with defined geometry and coordinate data. Furthermore, this file also had ids associated with each geographical neighborhood, which is exactly what was required to group our bicycle shops into neighborhoods.

Below is a visual representation of Neighborhoods.json in its geojson form.

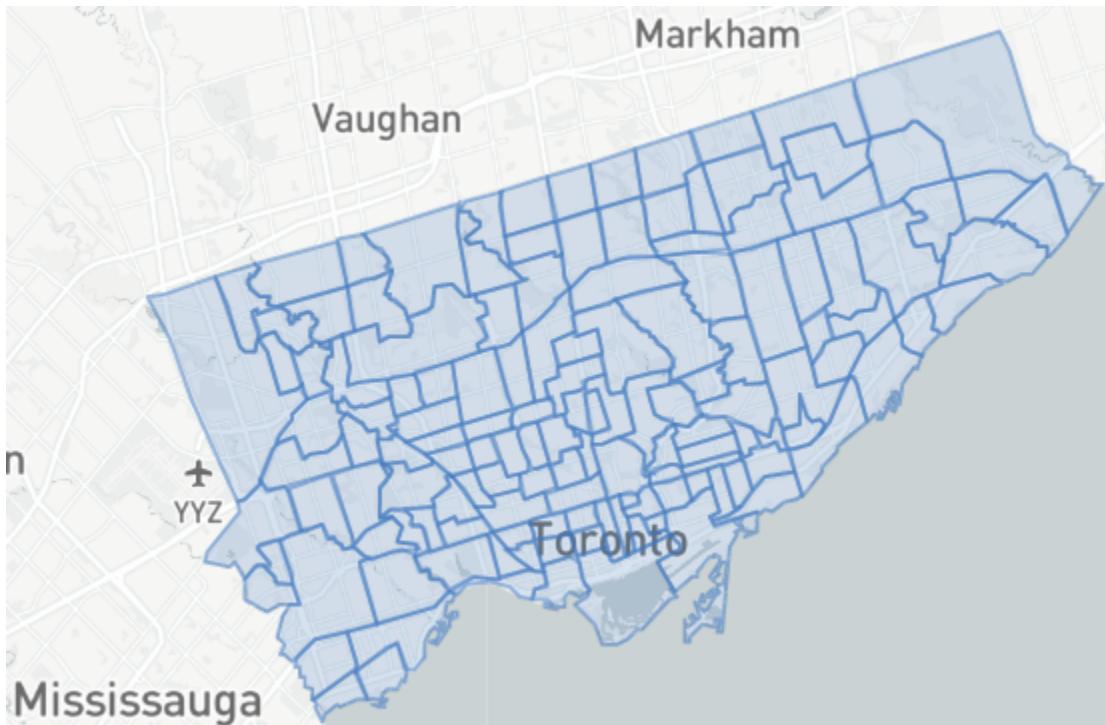


Figure 5

Leveraging the data provided by Neighborhoods.json, the coordinates of each bicycle shop were partitioned into neighborhoods. The process was performed in the **fqapi_to_csv.py** file as shown below.

Figure 6

```
import pandas as pd
import json
from shapely.geometry import shape, Point

bicycle_shop_coordinates_du = open("bicycle_shop_coordinates.csv", "w")

points_with_hood_id = []

data = json.load(open("FoursquareData.json", "r"))
for i in data["results"]:
    lat = i["geocodes"]["main"]["latitude"]
    lon = i["geocodes"]["main"]["longitude"]
    points_with_hood_id.append( { "point": Point(lon, lat), "hood_id": None, "lon": lon, "lat": lat } )

with open("Neighbourhoods.json", "r") as f:
    geojs = json.load(f)

for feature in geojs["features"]:
    polygon = shape(feature["geometry"])
    for point in points_with_hood_id:
        print(point["point"])
        if polygon.contains(point["point"]):
            point["hood_id"] = feature["properties"]["AREA_SHORT_CODE"]

for p in points_with_hood_id:
    bicycle_shop_coordinates_du.write(f"{p['lat']},{p['lon']},{p['hood_id']}") You, last week • find hood_
    bicycle_shop_coordinates_du.write("\n")
```

After this partitioning took place, the resulting data was saved to a CSV titled **bicycle_shop_coordinates.csv**, which can be seen below.

	A	B	C
1	latitude	longitude	hood_id
2	43.67861	-79.2985	63
3	43.65945	-79.4384	93
4	43.70011	-79.4552	108
5	43.67967	-79.3904	98
6	43.65921	-79.4393	93
7	43.66598	-79.408	95
8	43.6654	-79.4913	89
9	43.71248	-79.36	55
10	43.71636	-79.4001	99
11	43.77294	-79.2541	127
12	43.67908	-79.3434	69
13	43.66616	-79.3179	65
14	43.66646	-79.3426	70
15	43.66982	-79.3004	63
16	43.70914	-79.3627	55
17	43.65825	-79.352	70
18	43.65368	-79.3539	77
19	43.66065	-79.3706	73
20	43.65486	-79.3686	73
21	43.73541	-79.345	42
22	43.69995	-79.3972	100
23	43.71526	-79.3999	99
24	43.66051	-79.3999	79
25	43.66293	-79.4036	79
26	43.65559	-79.3988	78

Figure 7

The complete file consisted of 50 indexes, pertaining to the 50 bicycle shops retrieved from the Foursquare API. As shown, there are now neighborhood ids associated with each shop.

Now that all the necessary data had an associated neighborhood id, our team could then proceed to merge the data. All the merging and processing of the remaining data took place in the **data_process.py** file. As such, all the following screenshots were taken from that file until stated otherwise.

As one of the criteria for assessing the desirability of a neighborhood was the number of already existing bike stores, it was necessary to retrieve the number of stores already present in each neighborhood (this is why we required bike stores to have an associated neighborhood id). As stated before, the *lower* number of already existing stores is better as that would entail less business competition.

Shown below is the process of calculating the number of instances of an already existing bike store per neighborhood id.

```
from operator import index
import pandas as pd

df = pd.read_csv (r"bicycle_shop_coordinates.csv")

predata = {
    'hood_id': [],
    'number of bike shops': []
}

#Add stores to list while counting the number of instances a store appears in the same hood
hood_store_count = {}
for hood in df['hood_id']:
    if hood not in hood_store_count:
        hood_store_count[hood] = 1
    else:
        hood_store_count[hood] += 1

#Only keep unique instances of a hood
df = df.drop_duplicates(subset='hood_id')
df.reset_index(drop=True, inplace=True)

for key, value in hood_store_count.items():
    predata['hood_id'].append(key)
    predata['number of bike shops'].append(value)

df_revised = pd.DataFrame(predata)

#Add geodata to revised df
df_revised['latitude'] = df['latitude']
df_revised['longitude'] = df['longitude']
```

Figure 8

As numerous stores may have been in a single neighborhood, duplicate instances of a neighborhood id were removed by grouping them together, allowing the number of stores per neighborhood to be easily calculated. Grouping the stores into neighborhoods resulted in a reduction of 50 indexes in the data frame to 31, indicating that 19 of the 50 stores shared a common neighborhood id. After finding the number of existing bike stores per neighborhood, it was time to move onto the next criteria: the number of recorded bike thefts per neighborhood. As stated before, the *more* thefts in an area is better as people will require new bikes more often. Shown below is the process for obtaining the number of bike thefts per neighborhood id.

```

theft_data = pd.read_csv(r"bicycle_thefts.csv")
predata['bike_thefts'] = [0]*len(predata['hood_id'])

#Get number of bike thefts per hood
index= 0
for hood_id in theft_data[ 'Hood_ID']:
    try:
        theft_data['Hood_ID'][index] = int(hood_id)
    except Exception:
        theft_data.drop(df.index[index])
        index += 1
        continue

    if int(hood_id) in predata['hood_id']:
        i = predata['hood_id'].index(int(hood_id))
        predata['bike_thefts'][i] += 1

df_revised['bike_thefts'] = predata['bike_thefts']

```

Figure 9

Upon obtaining the number of recorded bike thefts per neighborhood, the code for satisfying the final criteria was developed; the average price of a purchased bicycle per neighborhood. As stated before, a *higher* average bike cost is better as it is an indication that people would be willing to pay a higher price when purchasing a new bicycle.

Shown below is the process utilized to obtain the average price of a bicycle per neighborhood id.

```

#Clean the data
theft_data = theft_data[theft_data.Cost_of_Bike != 0]
theft_data = theft_data[theft_data.Cost_of_Bike != '']
theft_data = theft_data[theft_data.Hood_ID != 'NSA']

#Get average bike cost per hood
bike_costs = (theft_data.groupby(['Hood_ID', 'Cost_of_Bike'], as_index=False).mean().groupby('Hood_ID')['Cost_of_Bike'].mean()).to_frame()

#Add bike costs to revised dataframe
avg_bike_cost = []
for hood_id in df_revised["hood_id"]:
    avg_bike_cost.append(list(bike_costs[str(hood_id)])[0])

df_revised["average_bicycle_cost"] = avg_bike_cost
predata['average_bicycle_cost'] = df_revised['average_bicycle_cost'].tolist()

```

Figure 10

As all of the required data was collected and merged into a single dataframe, a method of quantifying a neighborhood's desirability was necessary. A simple formula for the scoring metric was devised, as shown below.

$$SCORE_{desirability} = \frac{\text{Number of thefts}_{neighborhood n} \times \text{Average Cost of Bicycle}_{neighborhood n}}{\text{Number of Already Existing Stores}_{neighborhood n}}$$

The scoring metric created was a rough measure of a neighborhood's **monetary profitability**. Keeping this in mind, the devised scoring metric would have to be coherent to provide the highest scores to what is seen as the most profitable neighborhoods according to the established criteria. As the number of bike thefts and the average cost of a bicycle were seen as positive drivers, those factors were multiplied. As the number of already existing bicycle stores was seen as a negative driver, that was a dividing factor in the scoring metric.

Using the scoring metric above, a score was assigned to each neighborhood to signify its desirability with respect to each of the three criteria. These scores would be vital for plotting and analyzing the data collected. After assigning scores, each neighborhood id was also matched with its respective name retrieved from the bicycle_thefts.csv file. These names would be used for labels when visualizing the data. These scores were added to the finalized dataframe, and all the merged data was exported as a CSV to plot and map the results of the data processing, in a file titled **processed_data.csv**. The process for these steps are shown below.

Figure 11

```

predata['score'] = [0]*len(predata['hood_id'])
#Produce scores for each neighborhood
for i in range(0, len(predata['hood_id'])):
    score = predata['bike_thefts'][i]*predata['average_bicycle_cost'][i] / predata['number of bike shops'][i]
    predata['score'][i] = round(score ,2)

df_revised['score'] = predata['score']

#Obtain neighborhood name information
predata['neighborhood'] = []
hood_names = theft_data['NeighbourhoodName'].tolist()
hoodlist = theft_data['Hood_ID'].tolist()
for hood in predata['hood_id']:
    common_index = hoodlist.index(str(hood))
    predata['neighborhood'].append(hood_names[common_index])
df_revised['neighborhood'] = predata['neighborhood']

#Export to CSV
print(df_revised)
df_revised.to_csv("processed_data.csv")

```

Results

The results of the data processing and analysis will be represented in two formats. Contained within the **heatmap.py** file, the first will be a **heat map visualization** of the results to more easily signify the desirability of a specific neighborhood. The heat map will contain a marker indicating the location of a neighborhood, and the weight of each marker will be directly proportional to the score of a neighborhood. Furthermore, the heat map exhibits dynamic granularity, showing finer detail the more zoomed in it is. The following was used to develop the heat map and export it as an html, titled **heatmap.html**.

```
import pandas as pd
import folium
from folium.plugins import HeatMap, MarkerCluster

df = pd.read_csv (r"processed_data.csv")

#-----HEATMAP-----

map = folium.Map(location=[43.728136, -79.384666], zoom_start=11)
cluster = MarkerCluster().add_to(map)

coordinates = [list(items) for items in zip(df['latitude'], df['longitude'])]

index = 0
for i in range(len(coordinates)):
    folium.Marker(coordinates[i], popup= f"""
        Neighborhood: {df['neighborhood'][index]}\n
        Hood ID: {df['hood_id'][index]}\n
        Score: {df['score'][index]}
    """).add_to(cluster)

    index += 1

max_score = df['score'].max()

lat = df['latitude']
long = df['longitude']
weight = df['score']

data = []
for i in range(0, len(weight)):
    data.append([lat[i], long[i], weight[i]])

HeatMap(data).add_to(map)

map.save("heatmap.html")
```

Figure 12

The following is presented when viewing heatmap.html



Figure 13

The number 31 in the above image represents the 31 neighborhoods being examined for desirability. Upon zooming in, the granularity of the heat map increases. The following images show the granularity of the heat map increasing every level zoomed in.

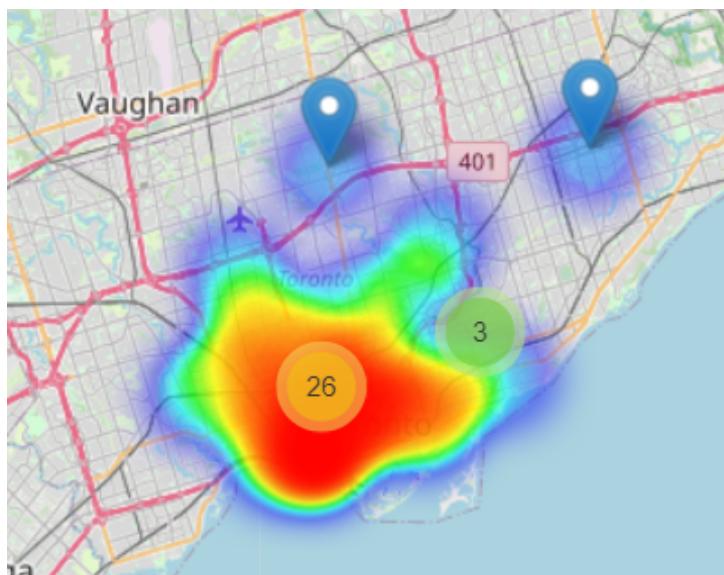


Figure 14

As shown above, every level zoomed in reveals more clusters of neighborhoods. The heat surrounding clusters of neighborhoods signifies the desirability of said cluster of neighborhoods based on the scores assigned to them. The higher the heat, the higher the score associated with those neighborhoods. As such, the most desirable neighborhoods will contain the most heat in the surrounding area.

Also evident from the image, zooming in results in some neighborhood markers being revealed as they are more distant from other clusters.

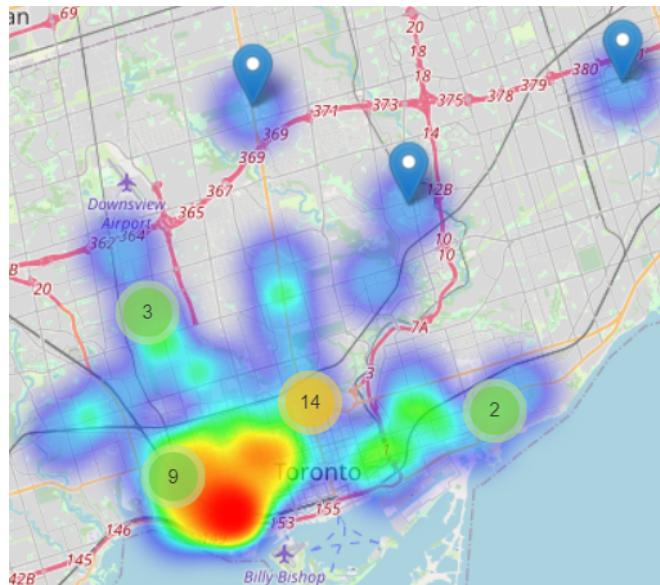


Figure 15

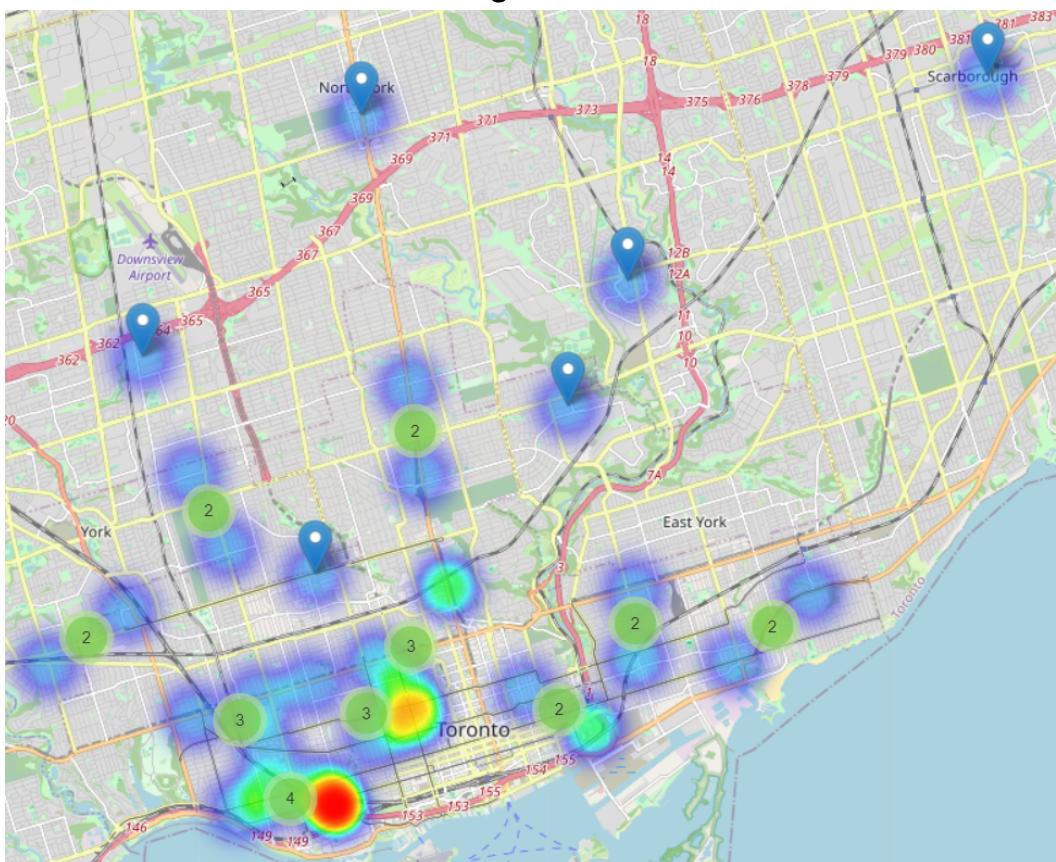


Figure 16

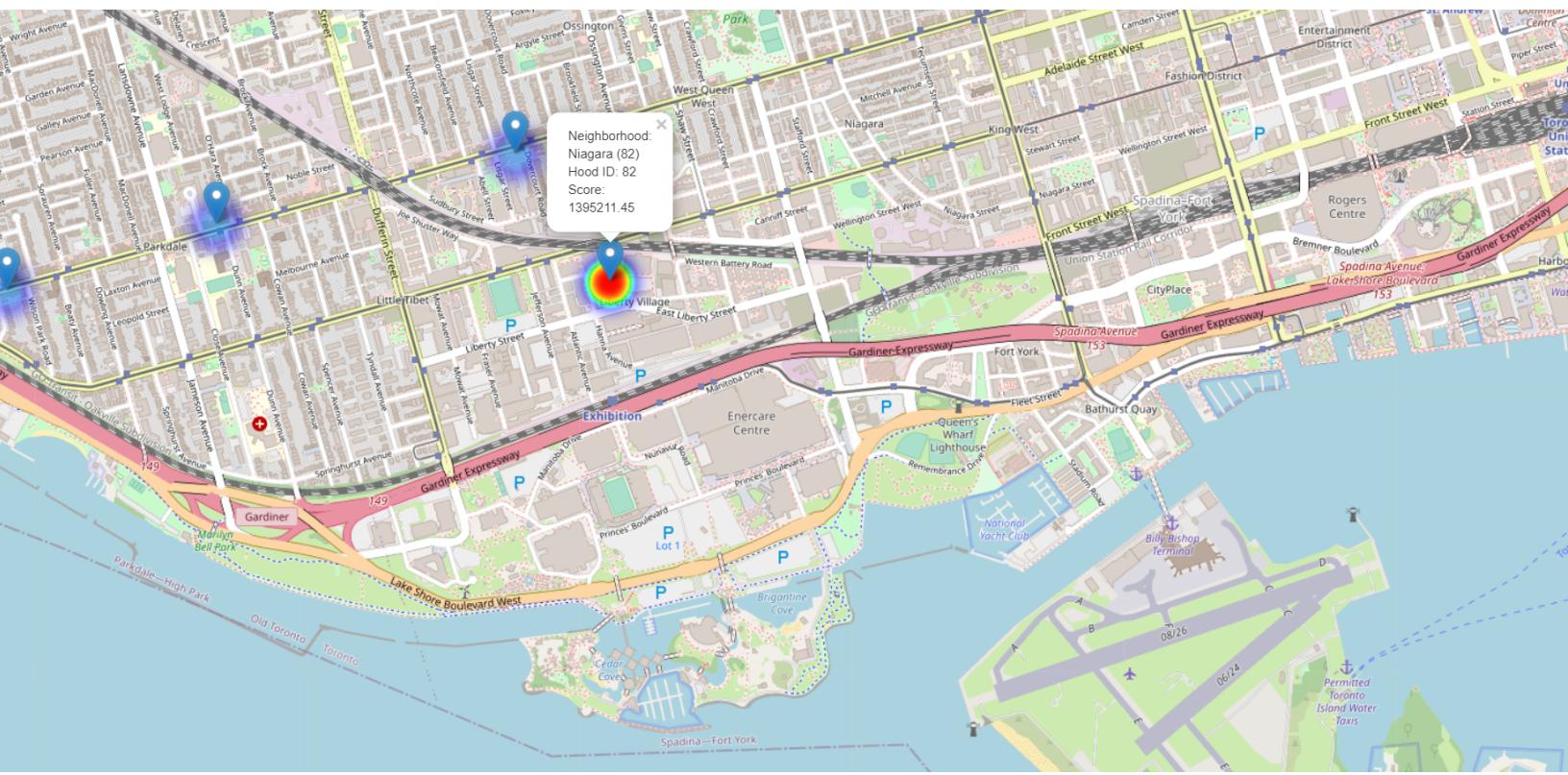


Figure 17

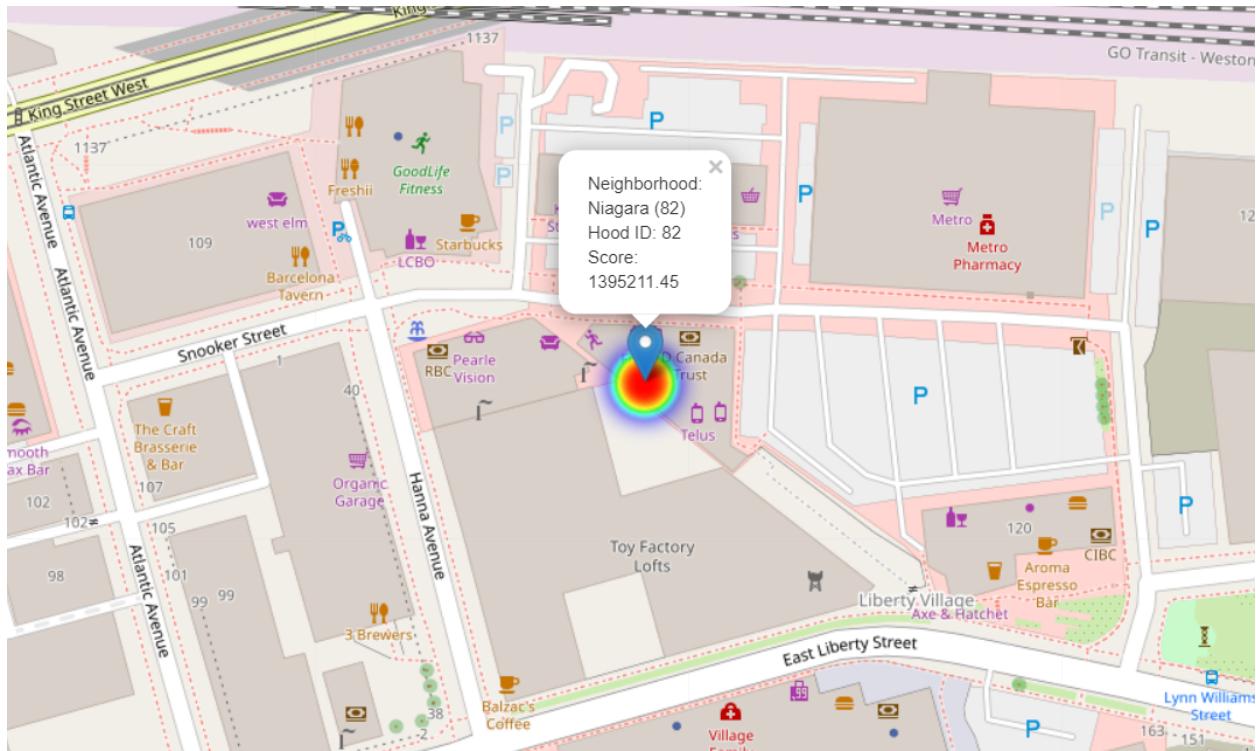


Figure 18

The second method of analyzing the collected data, contained within **kmeans_clustering.ipynb**, utilizes **k-means clustering** to plot the data and cluster them into meaningful groups. The general idea behind k-means clustering is to group the data points into “k” groups based on similarities in data, more specifically, the euclidean distances between the data points and predefined centroids for each cluster. All the points assigned to a cluster are collected and have the mean value amongst them calculated, and the euclidean distance from these means determine which cluster a point belongs to. This form of clustering is a more simple method of clustering, but is an efficient way to easily segment and group data in order to extract useful information from a dataset.

When initially plotting the collected data, our team decided the best scales to use for each axis would be the **Hood IDs** and the **Scores**. This would allow neighborhoods, represented by their hood id, to be segmented by their level of desirability, i.e. their respective scores. In order to view the data before clustering, the following code was used to plot neighborhoods against their desirability.

```
import pandas as pd
from matplotlib import pyplot as plt

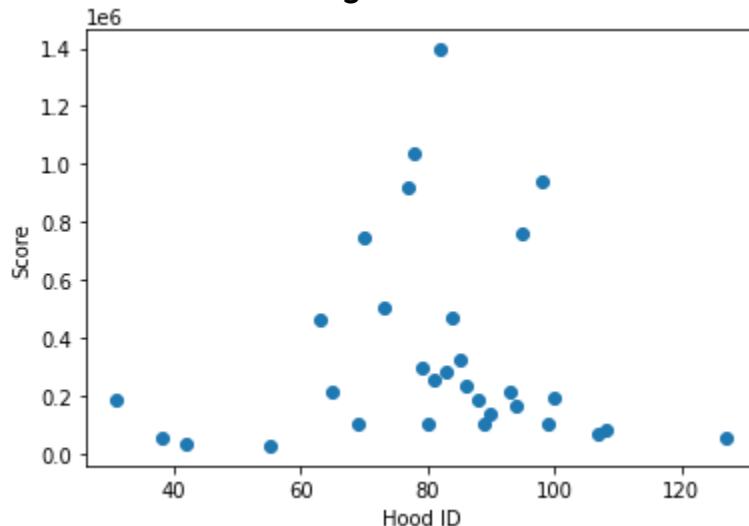
cluster_data = pd.read_csv("processed_data.csv")

plt.xlabel("Hood ID")
plt.ylabel("Score")
plt.scatter(cluster_data[ "hood_id"], cluster_data[ "score"])
```

Figure 19

Before clustering, the following graph represents hood ids vs their score.

Figure 20



From examining the above plot, our team believed the points could be clustered into 3 groups, each cluster representing three general levels of desirability. The following code was used to segment the dataset of hood ids vs their scores into 3 distinct clusters using the k-means clustering algorithm and export it to a CSV.

```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt

cluster_data = pd.read_csv("processed_data.csv")

km = KMeans(n_clusters=3)

y_predicted = km.fit_predict(cluster_data[["hood_id", "score"]])

cluster_data["cluster"] = y_predicted

df1 = cluster_data[cluster_data.cluster == 0]
df2 = cluster_data[cluster_data.cluster == 1]
df3 = cluster_data[cluster_data.cluster == 2]

plt.scatter(df1["hood_id"], df1["score"], color="red")
plt.scatter(df2["hood_id"], df2["score"], color="green")
plt.scatter(df3["hood_id"], df3["score"], color="blue")

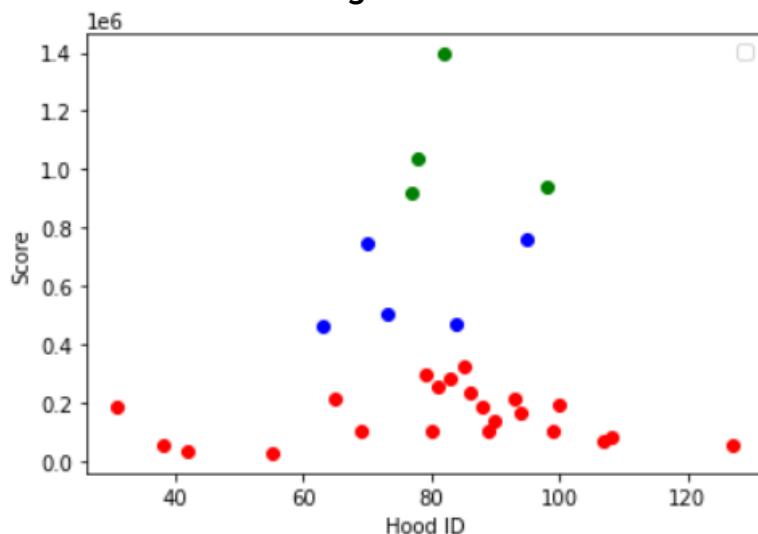
plt.xlabel("Hood ID")
plt.ylabel("Score")
plt.legend()

df = pd.DataFrame(cluster_data[["hood_id", "neighborhood", "score", "cluster"]])
print(df)
df.to_csv("segmented_neighborhoods.csv")
```

Figure 21

The following plot is the result of segmenting the data into 3 clusters.

Figure 22



The following is the list of neighborhoods and the cluster they have been segmented into, captured from the **segmented_neighborhoods.csv** file. **The green data points are a part of cluster 1, the blue data points are a part of cluster 2, and the red data points are a part of cluster 0.**

A	B	C	D	E
1	hood_id	neighborhood	score	cluster
2	0	63 The Beaches (63)	465570.93	2
3	1	93 Dovercourt-Wallace Emerson-Junction (93)	212023.86	0
4	2	108 Briar Hill-Belgravia (108)	81731.82	0
5	3	98 Rosedale-Moore Park (98)	944045.29	1
6	4	95 Annex (95)	761499.1	2
7	5	89 Runnymede-Bloor West Village (89)	105552.68	0
8	6	55 Thorncliffe Park (55)	28604.05	0
9	7	99 Mount Pleasant East (99)	105881.15	0
10	8	127 Bendale (127)	58773	0
11	9	69 Blake-Jones (69)	101675.1	0
12	10	65 Greenwood-Coxwell (65)	217697.56	0
13	11	70 South Riverdale (70)	750123.69	2
14	12	77 Waterfront Communities-The Island (77)	919067.42	1
15	13	73 Moss Park (73)	503606.5	2
16	14	42 Banbury-Don Mills (42)	37013.85	0
17	15	100 Yonge-Eglinton (100)	192190.91	0
18	16	79 University (79)	294202.3	0
19	17	78 Kensington-Chinatown (78)	1037767.84	1
20	18	81 Trinity-Bellwoods (81)	253224.99	0
21	19	94 Wychwood (94)	163849.23	0
22	20	80 Palmerston-Little Italy (80)	107514.37	0
23	21	82 Niagara (82)	1395211.45	1
24	22	84 Little Portugal (84)	472313.02	2
25	23	107 Oakwood Village (107)	72653.93	0
26	24	83 Dufferin Grove (83)	282152.62	0
27	25	86 Roncesvalles (86)	234449.13	0
28	26	38 Lansing-Westgate (38)	55167.43	0
29	27	88 High Park North (88)	188693.86	0
30	28	85 South Parkdale (85)	327063.59	0
31	29	90 Junction Area (90)	137578.55	0
32	30	31 Yorkdale-Glen Park (31)	187134.94	0

Figure 23

Discussion and Conclusion

This section of the report will discuss the findings of the **Results** section, first assessing the results of the heat map visualization of the data, followed by analyzing the results of segmenting and clustering the plot of hood id vs score.

As shown in **Figure 15**, the most desirable locations to open a bicycle store according to our team's criteria reside in the south-western corner of the downtown Toronto area. This is evident by the most concentration of heat in that area, which is directly proportional to the score exhibited by a neighborhood. There is a cluster of 9 neighborhoods in that region which are potential candidates for the most desirable location. Zooming in further, the heat map dynamically adjusts its level of granularity and segments the neighborhoods into smaller clusters., as shown in **Figure 16**. This reveals more specific locations of neighborhoods that would be the most desirable for our team to open a bicycle store. One of those regions contains a cluster of 4 candidates, and the other contains a cluster of 3 candidates, as shown below in more detail.

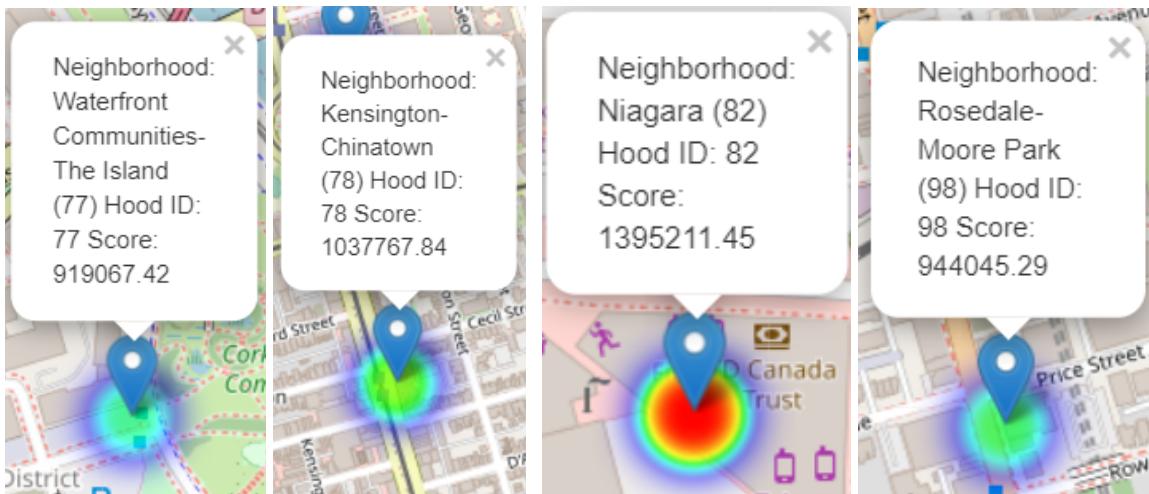


Figure 24

Upon zooming in another 3 levels, the clusters are eventually broken down into individual neighborhood markers, which indicate the desirability of each specific neighborhood id. As shown in **Figure 18**, the most desirable neighborhood to open a bicycle store in, according to our team's scoring criteria, is the **Niagara** neighborhood with neighborhood id **82** and a score of **1395211.45**. As shown in **Figure 17**, this neighborhood resides near the Toronto **downtown waterfront area**, not too distant from **Centre Island** and the **National Yacht Club**.

Based on our team's personal experience, this "most desirable" has a high number of bicycle riders in the area, particularly due to the concentration of bike stalls in the vicinity. Furthermore, due to the location being near the waterfront area, the weather conditions are typically optimal for bike riding. This shows that the results obtained from the heat map visualization align with real life trends, reflecting the success of the data analysis performed.

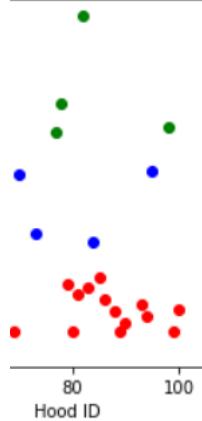
Observing the next few most desirable neighborhoods, our team began to notice a trend among the best candidates.



As shown in the images above, the most desirable neighborhoods have a neighborhood id within the high 70s to 100 range, and neighborhoods lower than that, such as the 50s and 60s ranges, do not present nearly as high of a score. This observation is further reinforced in **Figure 16**, showing how the neighborhoods with the highest scores are within the same general region.

Now it is time to analyze the data resulting from the k-means clustering performed on the plot of neighborhood ids vs their respective scores. As shown in **Figure 22**, the data was segmented into 3 clusters, represented by 3 different colors. From observing the clustered data, it is evident that the least desirable neighborhoods belong to the red cluster (cluster 0), the more “middle of the road” neighborhoods belong to the blue cluster (cluster 2), and the most desirable neighborhoods belong to the green cluster (cluster 1). As our team will obviously want to pick the neighborhoods with the highest scores, our primary focus falls upon the green cluster.

When further inspecting the green cluster (cluster 1), the correlation between the two representations of our results is immediately apparent. The neighborhood ids with the highest desirabilities fall between the high 70s and 100, directly matching our observations from the heat map. Shown below is a segment of the cluster data outlining this.



When taking a look at the CSV output of this clustered data, as shown in **Figure 23**, the neighborhoods belonging to the green cluster (cluster 1) can be grouped to easily view the most desirable neighborhoods as shown in the image below.

B	C	D	E
hood_id	neighborhood	score	cluster
98	Rosedale-Moore Park (98)	944045.29	1
77	Waterfront Communities-The Island (77)	919067.42	1
78	Kensington-Chinatown (78)	1037767.84	1
82	Niagara (82)	1395211.45	1

Once again, these results align perfectly with the observations made when inspecting the heat map visualization of the data, showing consistency amongst the process. From the clustered data, it can also be determined that the best neighborhood to open a bicycle store according to our team's criteria, is the **Niagara** neighborhood with neighborhood id **82**.

In conclusion, this project has been a resounding success. Our team was able to successfully satisfy the original business problem set out, which was to find the best neighborhood to open a bicycle store based on the following criteria; the *more* thefts in an area is better as people will require new bikes more often, the *lower* number of already existing stores is better as that would entail less business competition, and a *higher* average bike cost is better as it is an indication that people would be willing to pay a higher price when purchasing a new bicycle.

Our team was able to sufficiently process the data and put it into formats to easily interpret and perform further analysis on. After collecting all the necessary data, our team was able to represent our findings in multiple formats with a high degree of effectiveness and accuracy. Furthermore, the conclusions resulting from this project have not only been consistent and coherent, but have also aligned with real life trends and patterns, showing the validity of our team's process and the success of this project.