COVID 19 – Precautions and Venue Data Analysis of Seattle

Sathya Thiruvengadam

Jun 2020

1. Introduction

1.1Description and Discussion of the Background

The COVID-19 Pandemic, also known as the corona virus pandemic, is an ongoing pandemic of corona virus disease 2019(COVID-19), caused by severe acute respiratory syndrome corona virus 2(SARS-CoV-2). The virus that causes COVID-19 is spreading very easily and sustainably between people. Information from the ongoing COVID-19 pandemic suggest that this virus is spreading more efficiently than influenza, but not as efficiently as measles, which is highly contagious.

The virus is spreading between person-to-person due to close contact, tiny droplets from sneezing, coughing can spread in air and people get infected by touching the contaminated areas and touching their face. As of today, the graph shows the consistent upward arch, while in the other end thing are getting to normal called “new” normal with precautions. But still people find it difficult to get adjusted to new normal, where bringing more risk prone zones for easy contamination.

1.2 Problem

By analyzing the venues of high risky area(where it is highly crowded) and making a legally bonded procedures with extra precautions to handle based on the nature of the venue is an optimal way to control the contamination while we step into the stage of introducing new normal to the people. The idea behind analyzing the venues of Seattle is due to the constant increase in COVID-19 cases and a big city with large crowd, in future this model can be used in any cities for this purpose as states started slowly announcing to reverse the lock down.

1.3 Interest

Government official to consider extra measures to take precautions and monitor the process to minimize the contamination. General public knowing the high-risk prone areas to undertake extra personnel precautions or to the max avoiding the risk prone areas.

2. Data Description

To address this problem and analyze

2.1 Neighborhood data

The data of Seattle Neighborhood is scraped from Wikipedia using BeautifulSoup- python library for web scraping. Wiki table had more column, pulled only district and neighborhood, did not have postal code included- assumed to have unique postal code. Cleaned the data to remove the reference link attached in neighborhood or borough, and to have aligned column.

	Borough	Neighborhood
0	Seattle	North Seattle
1	North Seattle	Broadview
2	North Seattle	Bitter Lake
3	North Seattle	North Beach / Blue Ridge
4	North Seattle	Crown Hill

2.2 Geo-coding

Performing geocoding in python with the help of Geopy and Geopandas libraries to retrieve the **geographical coordinates** of an address from neighborhood data.

```
# used the column address to pull the latitude and longitude for the location, if found None - replace to NaN
import geopy

def main():

    def get_latitude(x):
        if hasattr(x,'latitude') and (x.latitude is not None):
                return x.latitude


    def get_longitude(x):
          if hasattr(x,'longitude') and (x.longitude is not None):
                  return x.longitude


    geolocator = Nominatim(user_agent="sea_d", timeout=5)
    geolocate_column = df_seat['Address'].apply(geolocator.geocode)
    df_seat['latitude'] = geolocate_column.apply(get_latitude)
    df_seat['longitude'] = geolocate_column.apply(get_longitude)



if __name__ == '__main__':
    main()
```

	Neighborhood	level_1	Borough	Address	latitude	longitude
0	North Seattle	0	Seattle	North Seattle, Seattle	47.660773	-122.291497
1	Broadview	0	North Seattle	Broadview, North Seattle	47.722320	-122.360407
2	Bitter Lake	0	North Seattle	Bitter Lake, North Seattle	47.726236	-122.348764
3	North Beach	0	North Seattle	North Beach , North Seattle	47.696210	-122.392362
4	Blue Ridge	0	North Seattle	Blue Ridge, North Seattle	47.701487	-122.375407

2.3 Venues for clustering

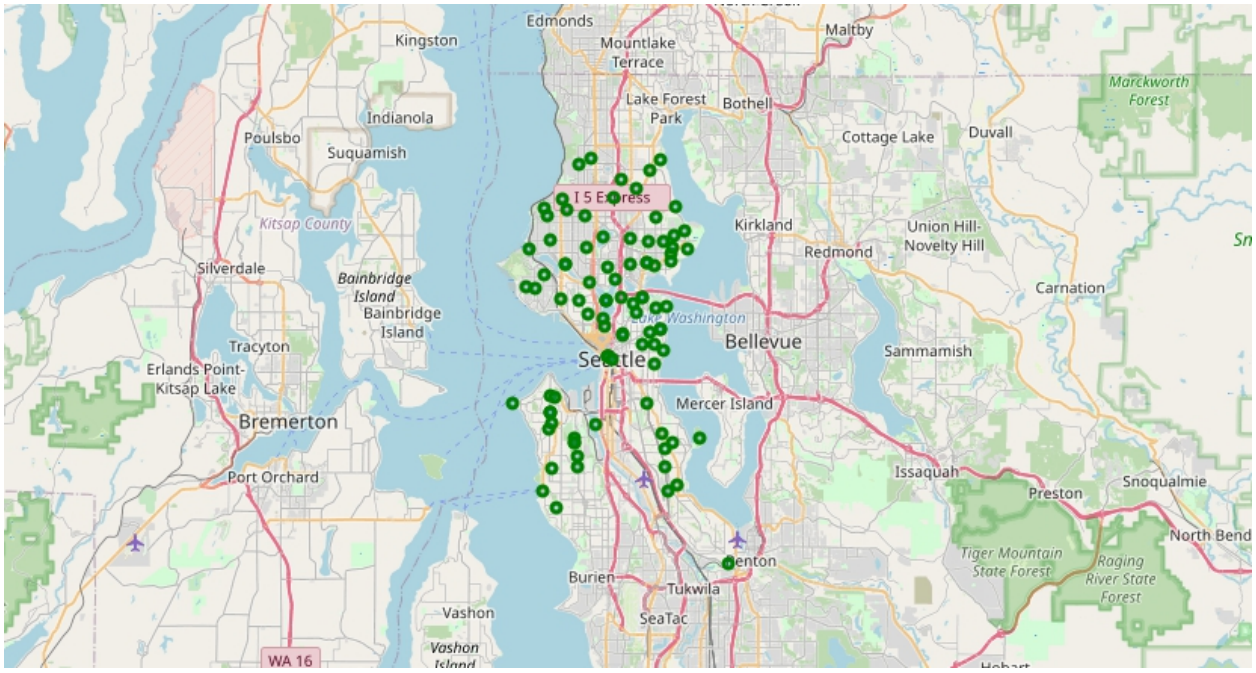
Using the geographical coordinates obtained, the venues are gathered using **FourSquare API** to perform clustering.

3. Methodology:

As mentioned in data section, scraped data from Wikipedia, cleaned and used geocoder to obtain the geographical coordinates of the address listed, the data used below,

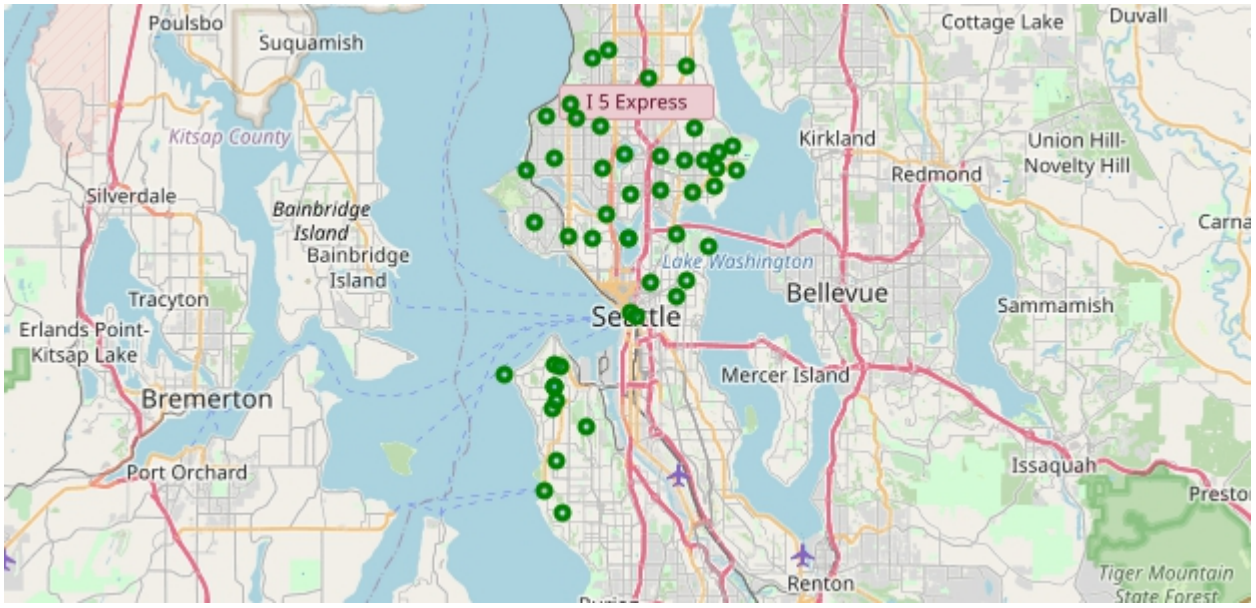
	Neighborhood	level_1	Borough	Address	latitude	longitude
0	North Seattle	0	Seattle	North Seattle, Seattle	47.660773	-122.291497
1	Broadview	0	North Seattle	Broadview, North Seattle	47.722320	-122.360407
2	Bitter Lake	0	North Seattle	Bitter Lake, North Seattle	47.726236	-122.348764
3	North Beach	0	North Seattle	North Beach , North Seattle	47.696210	-122.392362
4	Blue Ridge	0	North Seattle	Blue Ridge, North Seattle	47.701487	-122.375407

I used python **folium** library to visualize geographic details of Seattle and its boroughs and I created a map of Seattle with boroughs superimposed on top. I used latitude and longitude values obtained for each neighborhood to get the visual as below:



Narrowed the borough that contains the term Seattle in it and used folium package to visualize the location map,

```
sea_data = df_seattle[df_seattle['Borough'].str.contains("Seattle")].reset_index(drop=True)
sea_data.head()
```



Then used the Foursquare API to explore the boroughs and segment them, set the boundaries with the radius of 1000 meters and allocated the limit as 50 venues for each

borough from their geographical coordinates' location. Below listed the head of the result includes Venue name, category, latitude, and longitude details combined with neighborhood and boroughs.

```
radius = 1000
LIMIT = 50

venues = []

for lat, long, borough, neighborhood in zip(sea_data['latitude'], sea_data['longitude'], sea_data['Borough'], sea_data['Neighborh
    url = "https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}".format(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        lat,
        long,
        radius,
        LIMIT)

    results = requests.get(url).json()["response"]['groups'][0]['items']

    for venue in results:
        venues.append((
            borough,
            neighborhood,
            lat,
            long,
            venue['venue']['name'],
            venue['venue']['location']['lat'],
            venue['venue']['location']['lng'],
hoods - Cluster Analysis(Seattle).ipynb                    categories'][0]['name']))
```

	Borough	Neighborhood	BoroughLatitude	BoroughLongitude	VenueName	VenueLatitude	VenueLongitude	VenueCategory
0	Seattle	North Seattle	47.660773	-122.291497	Burke-Gilman Brewing Company	47.661308	-122.288067	Brewery
1	Seattle	North Seattle	47.660773	-122.291497	Jak's Grill	47.661072	-122.288073	Steakhouse
2	Seattle	North Seattle	47.660773	-122.291497	Center for Urban Horticulture	47.657978	-122.290237	College Science Building
3	Seattle	North Seattle	47.660773	-122.291497	University Village	47.662487	-122.298531	Shopping Plaza
4	Seattle	North Seattle	47.660773	-122.291497	The North Face	47.662400	-122.298158	Sporting Goods Shop

From the detail count, it shows there are 269 unique categories listed and now each neighborhood is analyzed individually to understand the most common place tend to be crowded within 1000 meters.

The process is taken further by using "one-hot encoding" function of the python panda's library'. One hot encoding converts the categorical variables into a form that could be help ML algorithms for better prediction accuracy.

```
# one hot encoding
sea_onehot = pd.get_dummies(venues_df[['VenueCategory']], prefix="", prefix_sep="")

# add borough and neighborhood column back to dataframe
sea_onehot['Borough'] = venues_df['Borough']
sea_onehot['Neighborhood'] = venues_df['Neighborhood']

# move borough and neighborhood column to the first column
fixed_columns = list(sea_onehot.columns[-3:]) + list(sea_onehot.columns[:-3])
sea_onehot = sea_onehot[fixed_columns]

print(sea_onehot.shape)
sea_onehot.head()
```

	Zoo	Zoo Exhibit	Borough	ATM	Accessories Store	Adult Boutique	Alternative Healer	American Restaurant	Amphitheater	Antique Shop	Arcade	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	R
0	0	0	Seattle	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	Seattle	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	Seattle	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	Seattle	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	Seattle	0	0	0	0	0	0	0	0	0	0	0	0	

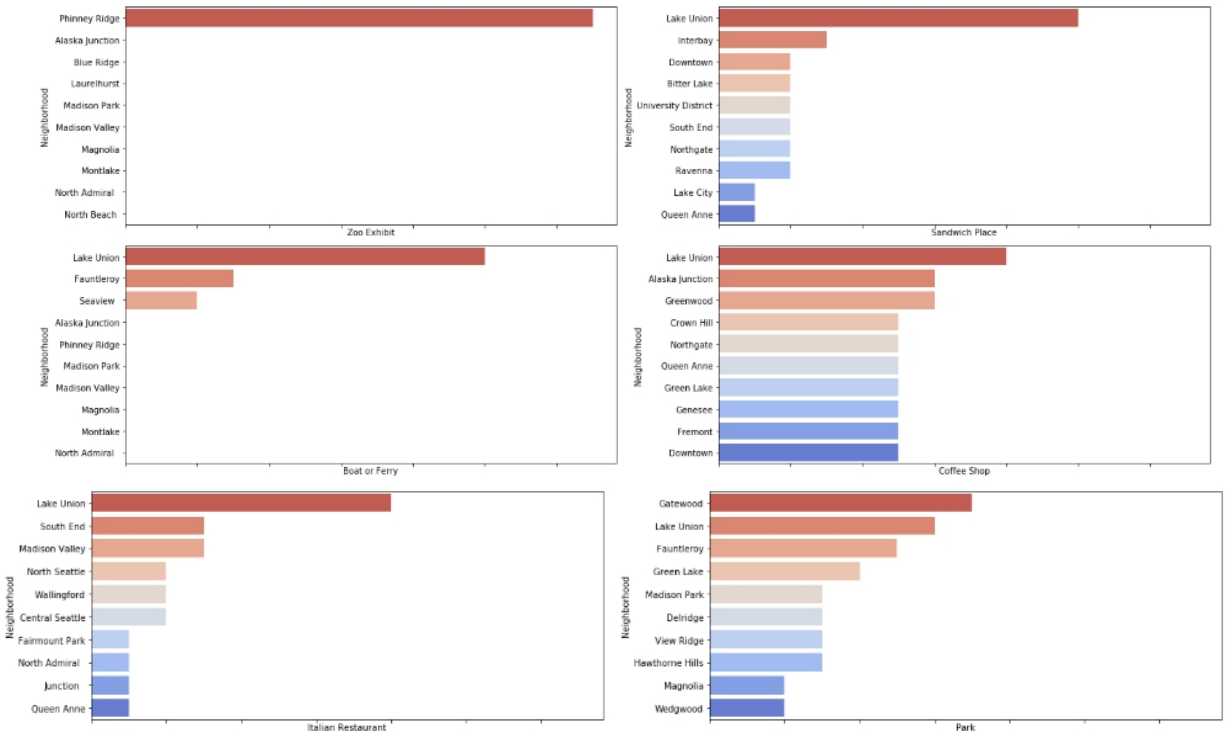
The top 10 venue categories can be found by counting their occurrences, the top venue is listed are Zoo Exhibit, sandwich place, boat or ferry, coffee shop and Italian restaurants.

```
venue_counts_described = venue_counts.describe().transpose()
```

```
venue_top10 = venue_counts_described.sort_values('max', ascending=False)[0:10]
venue_top10
```

	count	mean	std	min	25%	50%	75%	max
Zoo Exhibit	48.0	0.270833	1.876388	0.0	0.0	0.0	0.0	13.0
Sandwich Place	48.0	0.750000	1.577771	0.0	0.0	0.0	1.0	10.0
Boat or Ferry	48.0	0.312500	1.518065	0.0	0.0	0.0	0.0	10.0
Coffee Shop	48.0	2.895833	2.013144	0.0	1.0	3.0	5.0	8.0
Italian Restaurant	48.0	0.750000	1.328893	0.0	0.0	0.0	1.0	8.0
Park	48.0	1.312500	1.613194	0.0	0.0	1.0	2.0	7.0
Pizza Place	48.0	1.937500	1.743148	0.0	1.0	1.0	3.0	6.0
Bus Stop	48.0	0.354167	1.101055	0.0	0.0	0.0	0.0	6.0
Deli / Bodega	48.0	0.291667	0.944375	0.0	0.0	0.0	0.0	6.0
Beach	48.0	0.375000	0.913842	0.0	0.0	0.0	0.0	5.0

Then the top 10 venues are plotted individually by neighborhoods using seaborn python library for easy visual reading to see possibility on where to be crowded for each neighborhood.

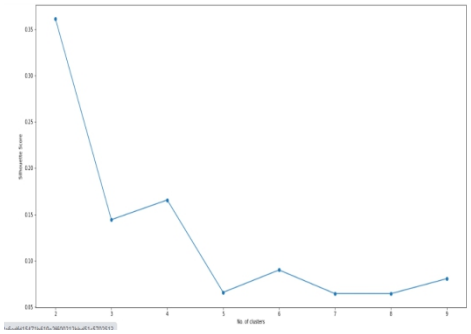


Machine Learning – Clustering (K-Means)

K-Means is an unsupervised ML algorithm which creates cluster of data points based on the similarities among the data point. This algorithm will be used to count neighborhoods for each cluster for variable cluster size.

With initializing the K-Means n to 10, calculate the silhouette_score to find the optimal cluster value,

```
from sklearn.metrics import silhouette_samples, silhouette_score

indices = []
scores = []

sea_grouped_clustering = sea_grouped.drop(["Borough", "Neighborhood"], 1)

for kclusters in range(2, max_range) :

    # Run k-means clustering
    seagc = sea_grouped_clustering
    kmeans = KMeans(n_clusters = kclusters, init = 'k-means++', random_state = 0).fit_predict(seagc)

    # Gets the score for the clustering operation performed
    score = silhouette_score(seagc, kmeans)

    # Appending the index and score to the respective lists
    indices.append(kclusters)
    scores.append(score)
```

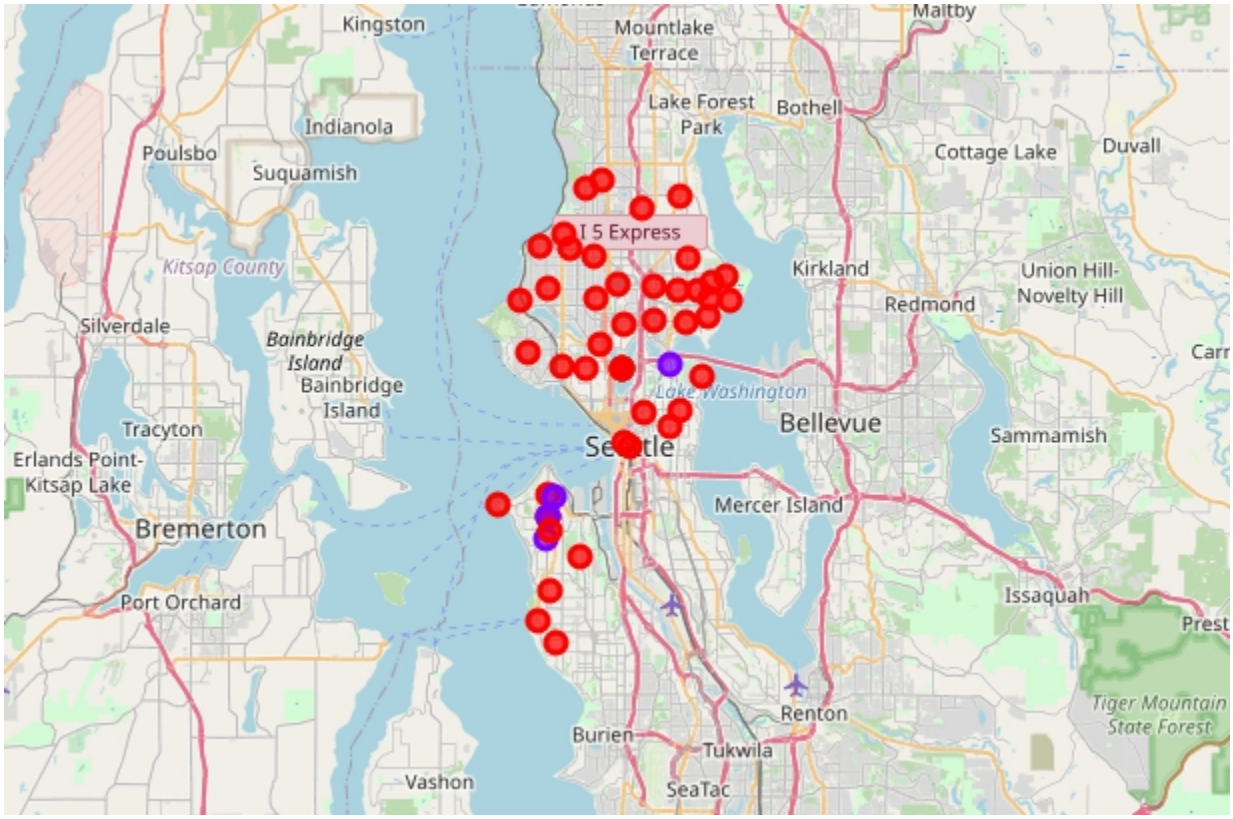


With the optimal value k being 2 , created the KMeans model using the sklearn python packages. Created a dataframe that includes cluster label and the top ten venues listed.

```
# set number of clusters
kclusters = opt

#sea_grouped_clustering = sea_grouped.drop([ "Borough", "Neighborhoods"], 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(sea_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

	Neighborhood	level_1	Borough	Address	latitude	longitude	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
0	North Seattle	0	Seattle	North Seattle, Seattle	47.660773	-122.291497	0	Arts & Crafts Store	Pizza Place	Ice Cream Shop	Lingerie Store	Furniture / Home Store	Italian Restaurant	Cosmetics Shop
1	Broadview	0	North Seattle	Broadview, North Seattle	47.722320	-122.360407	0	Pizza Place	Furniture / Home Store	Trail	Food Truck	Video Store	Beer Bar	Sushi Restaurant
2	Bitter Lake	0	North Seattle	Bitter Lake, North Seattle	47.726236	-122.348764	0	Fast Food Restaurant	Thai Restaurant	Coffee Shop	Sandwich Place	Athletics & Sports	Donut Shop	Automotive Shop
3	North Beach	0	North Seattle	North Beach , North Seattle	47.696210	-122.392362	0	Beach	Trail	Park	Tea Room	Mexican Restaurant	Athletics & Sports	Gluten-free Restaurant
4	Blue Ridge	0	North Seattle	Blue Ridge, North Seattle	47.701487	-122.375407	0	Pizza Place	Coffee Shop	Pet Store	Mexican Restaurant	Pool	Breakfast Spot	Baseball Field

I used folium library again to visualize the seattle neighborhood with the cluster segmentation included on this,



4. Results:

There are two clusters, Below are the cluster information on the venue that suppose to be crowded that can be monitored for extra precautions and safety,

Result from cluster analysis – 0

Most 1st common

Coffee Shop	16
Park	6
Pizza Place	6
Boat or Ferry	4
Café	2
Bar	2
Auto Workshop	1

Most 2nd common

Coffee Shop	7
Pizza Place	4
Sandwich Place	4
Park	3
Bank	2
Beach	2
Bar	2

Neighborhood,

```
-----------------------------------------
Lake Union,  Central Seattle           2
Lake Union, North Seattle              2
Downtown, Central Seattle              1
Genesee, West Seattle                  1
Green Lake, North Seattle              1
South End, Seattle                     1
Wallingford, North Seattle             1
Gatewood, West Seattle                 1
Madison Park,  Central Seattle         1
Queen Anne, Central Seattle            1
Central Seattle, Seattle               1
Ravenna, North Seattle                 1
Magnolia, Central Seattle              1
```

Result from cluster analysis – 1

Most 1st Common Most 2nd Common

Coffee Shop	2	Pizza Place	1
Pizza Place	1	Pub	1
Trail	1	Coffee Shop	1

Neighborhood,

```
Fairmount Park, West Seattle             1
Montlake,  Central Seattle               1
 West Seattle Junction , West Seattle    1
 Alaska Junction, West Seattle           1
```

5. Discussion:

Performed cluster analysis with available data information and the coordinates obtain, segmented into two clusters with most common places being coffee shops, restaurants, boat ferry, park, and beach are most crowded places, Not more surprising that both cluster and the most common place look more alike.

As I mentioned in the data column, the neighborhood information did not contain postal code, assumed that each neighborhood in a borough belong to different postal code, for more detailed and accurate model, the data set can be expanded and verified by obtaining it from specific platform as most current data set.

This model can be expanded further to other states and cities, that is more for discussion in gathering more accurate data.

6. Conclusion:

During this phase of time, it is crucial to act with more precautions and cautions, the graph shows the consistent upward arch while in the other end thing are getting to normal called “new” normal with precautions. But still people find it difficult to get adjusted to new normal, where bringing more risk prone zones for easy contamination.

The main goal of building this model is to obtain a most crowded places in each neighborhood/borough to set a rules based on the venue to maintain social distancing and keep away to get adjusted to new normal and is our responsibility to bend the curve of COVID-19.

References :-

- https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Seattle
- https://developers.google.com/maps/documentation/geocoding/
- https://developer.foursquare.com/
- Applied data science capstone lab notebook created by the staffs.