Clustering Neighborhoods in Richmond, Virginia

Introduction to the Problem

We would try to implement the similar problem we have been taught and discussed in the course itself. We would try to find out that how similar or dissimilar two areas of a city are considering some specific features. For our case we are going to consider Richmond, Virginia, It was not easy to fing the Richmond, Virginia dataset but still, we managed to collect it.

Solution

Here I will convert addresses to their corresponding latitude and longitude values. I will use the Foursquare API to explore neighborhoods in Richmond, Virginia. I will use the explore function to get the most common venue categories in each neighborhood, and then use this feature to group the neighborhoods into clusters. I will use the k-means clustering algorithm to complete this task. Finally, I will use the Folium library to visualize the neighborhoods in Richmond, Virginia and their emerging clusters

Way to the Solution

- Download and Explore Dataset
- Explore Neighborhoods in Richmond, Virginia
- Analyze Each Neighborhood
- Cluster Neighborhoods
- Examine Clusters

Installing all the required dependencies

In [33]:

```
# !pip install geocoder
```

Import each and every required library and package

- BeautifulSoup and requests for scraping the data
- Pandas and numpy for making structure and preprocessing of the data
- Geopy for getting the long and lats of the places

- Folium for maps and more information
- Matplotlib for visualization
- Sklearn for KMeans model

In [1]:

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
from geopy.geocoders import Nominatim

import numpy as np
import matplotlib.cm as cm
import matplotlib.colors as colors
import folium
from sklearn.cluster import KMeans
```

Scrapping of the datafrom the wikipedia page https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Richmond,_Virginia (https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Richmond,_Virginia)

After doing the proper inspection of the page I got to know that the the names are stored under ul tags.

In [6]:

```
data = requests.get("https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Richmond,_Virginia")
print('got data')
soup = BeautifulSoup(data, 'html.parser')
neighborhoodList = []
for row in soup.find_all("ul",)[1:6]:
    neighborhoodList.extend(row.text.split('\n'))

kl_df = pd.DataFrame({"Neighborhood": neighborhoodList})
kl_df.head()

```

```
got data
```

Out[6]:

	Neighborhood
0	Arts District
1	Biotech and MCV
2	City Center
3	Court End
4	Gambles Hill

Geolocation coordinates generation of the places

In [10]:

```
geolocator = Nominatim(user_agent="courcera_capston")
new_list = []
def get_latlng(neighborhood):
    global new_list
    location = geolocator.geocode('{}, Richmond, Virginia'.format(neighborhood))
    try:
      loc = (location.latitude, location.longitude)
      new_list.append(neighborhoodList)
      return loc
    except:
      pass
coords = [get_latlng(neighborhood) for neighborhood in kl_df["Neighborhood"].tolist() if get_lat
```

Get the location of the city Richmond, Virginia and combning them to the location data frame.

In [12]:

```
address = 'Richmond, Virginia'
geolocator = Nominatim(user_agent="courcera_capston")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Richmond, Virginia {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate of Richmond, Virginia 37.5385087, -77.43428.
```

In [13]:

```
df_coords = pd.DataFrame(coords, columns=['Latitude', 'Longitude'])
kl_df['Latitude'] = df_coords['Latitude']
kl_df['Longitude'] = df_coords['Longitude']
kl_df.dropna(inplace=True)
print(kl_df.shape)
```

```
(95, 3)
```

Plot the datapoints of the dataframe on the map using folium

In [14]:

```

map_kl = folium.Map(location=[latitude, longitude], zoom_start=11)
for lat, lng, neighborhood in zip(kl_df['Latitude'],  kl_df['Longitude'], kl_df['Neighborhood'])
 label = '{}'.format(neighborhood)
 label = folium.Popup(label, parse_html=True)
 folium.CircleMarker([lat, lng],radius=5,popup=label,color='blue',fill=True,fill_color='#3186cc'
map_kl
```

Out[14]:

Connecting to the foursquare api to get more info about the locations

In [15]:

```
CLIENT_ID = 'JH54IDPYRYILFWBGNXRIB2UXSNYGDGUJVHKPROH44R0TLGII'
CLIENT_SECRET = '1C0YP3ZVJP3ZS3VOQEWAUP4DJM5TBBBHMTIFUTCEAGYZQKBM'
VERSION = '20180605'
radius = 2000
LIMIT = 100
venues = []
for lat, long, neighborhood in zip(kl_df['Latitude'], kl_df['Longitude'], kl_df['Neighborhood'])
  url = "https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&v={}&ll={},{
  results = requests.get(url).json()["response"]['groups'][0]['items']
  for venue in results:
      venues.append((neighborhood,lat,long,venue['venue']['name'],
      venue['venue']['location']['lat'],venue['venue']['location']    ['lng'],venue['venue']['ca
```

In [16]:

```
venues_df = pd.DataFrame(venues)
venues_df.columns = ['Neighborhood', 'Latitude', 'Longitude', 'VenueName', 'VenueLatitude', 'Ver
print(venues_df.shape)
venues_df.head()
```

```
(5981, 7)
```

Out[16]:

	Neighborhood	Latitude	Longitude	VenueName	VenueLatitude	VenueLongitude	VenueCategory
0	Arts District	37.545853	-77.44231	Quirk Hotel	37.546500	-77.444085	Hotel
1	Arts District	37.545853	-77.44231	Perly's	37.543848	-77.441436	Deli / Bodega
2	Arts District	37.545853	-77.44231	Mama Js	37.546469	-77.439696	Southern / Soul Food Restaurant
3	Arts District	37.545853	-77.44231	Salt & Forge	37.545206	-77.440183	Sandwich Place
4	Arts District	37.545853	-77.44231	Saison Market	37.546844	-77.442219	Food & Drink Shop

In [17]:

```
print('There are {} unique categories.'.format(len(venues_df['VenueCategory'].unique())))
venues_df['VenueCategory'].unique()
```

```
       'General Entertainment', 'Music Venue', 'Advertising Agency',
       'Bookstore', 'Sports Bar', 'Mediterranean Restaurant',
       'Asian Restaurant', 'German Restaurant', 'Bar', 'Pub',
       'Italian Restaurant', 'Gym', 'Theater', 'Monument / Landmark',
       'College Gym', 'Tea Room', 'Bistro', 'Art Museum', 'Park',
       'American Restaurant', 'Dance Studio', 'Mexican Restaurant',
       'Food Truck', 'Pizza Place', 'Historic Site',
       'Vegetarian / Vegan Restaurant', 'Trail', 'Caribbean Restaurant',
       'College Theater', 'Breakfast Spot', 'Burger Joint', 'Donut Shop',
       'Thai Restaurant', 'Cuban Restaurant', 'Thrift / Vintage Store',
       'History Museum', 'Clothing Store', 'Hot Dog Joint', 'Salad Place',
       'Neighborhood', 'Museum', 'Bagel Shop', 'River', 'Post Office',
       'Lake', 'Fish & Chips Shop', 'Bakery', 'BBQ Joint',
       'Scenic Lookout', 'Noodle House', 'Speakeasy', 'Diner',
       'Playground', 'Movie Theater', 'Sushi Restaurant',
       'Fried Chicken Joint', 'Dive Bar', 'Pool', 'Smoke Shop',
       'Farmers Market', 'Nightclub', 'Liquor Store',
       'Fast Food Restaurant', 'Ethiopian Restaurant', 'Library',
       'Discount Store', 'Pharmacy', 'Chinese Restaurant',
       'Residential Building (Apartment / Condo)',
```

In [18]:

```python
# One hot encoding of the l
kl_onehot = pd.get_dummies(venues_df[['VenueCategory']], prefix="", prefix_sep="")
# Adding neighborhood column back to dataframe
kl_onehot['Neighborhoods'] = venues_df['Neighborhood']
# Moving neighbourhood column to the first column
fixed_columns = [kl_onehot.columns[-1]] + list(kl_onehot.columns[:-1])
kl_onehot = kl_onehot[fixed_columns]
print(kl_onehot.head())
```

```
   Neighborhoods  ATM  Accessories Store  Adult Boutique  Advertising Agency  \
0  Arts District    0                  0               0                   0
1  Arts District    0                  0               0                   0
2  Arts District    0                  0               0                   0
3  Arts District    0                  0               0                   0
4  Arts District    0                  0               0                   0

   American Restaurant  Antique Shop  Art Gallery  Art Museum  \
0                    0             0            0           0
1                    0             0            0           0
2                    0             0            0           0
3                    0             0            0           0
4                    0             0            0           0

   Arts & Crafts Store  ...  Video Store  Vietnamese Restaurant  \
0                    0  ...            0                      0
1                    0  ...            0                      0
2                    0  ...            0                      0
3                    0  ...            0                      0
4                    0  ...            0                      0

   Volleyball Court  Warehouse Store  Waterfall  Wine Bar  Wine Shop  \
0                 0                0          0         0          0
1                 0                0          0         0          0
2                 0                0          0         0          0
3                 0                0          0         0          0
4                 0                0          0         0          0

   Wings Joint  Women's Store  Yoga Studio
0            0              0            0
1            0              0            0
2            0              0            0
3            0              0            0
```

```
4            0              0            0

[5 rows x 247 columns]
```

In [19]:

```
kl_grouped=kl_onehot.groupby(["Neighborhoods"]).sum().reset_index()
print(kl_grouped.shape)
kl_grouped.head()
```

```
(94, 247)
```

Out[19]:

	Neighborhoods	ATM	Accessories Store	Adult Boutique	Advertising Agency	American Restaurant	Antique Shop	Art Gallery	Art Museum	Arts & Crafts Store	...	Video Store	Vietnamese Restaurant	Vc
0	Ancarrow's Landing	0	0	0	0	1	0	0	0	0	...	0	0	
1	Arts District	0	0	0	1	2	0	3	1	0	...	0	0	
2	Barton Heights	1	0	0	0	1	0	0	0	0	...	1	0	
3	Bellemeade	0	0	0	0	1	0	0	0	0	...	1	0	
4	Bellevue	0	0	0	1	2	0	3	1	0	...	0	0	

5 rows × 247 columns

In [20]:

```
# Creating a dataframe for Shopping Mall data only
kl_mall = kl_grouped[["Neighborhoods","Shopping Mall"]]
```

In [29]:

```

kclusters = 2
kl_clustering = kl_mall.drop(["Neighborhoods"], 1)
# Run k-means clustering algorithm



kmeans = KMeans(n_clusters=kclusters,random_state=0).fit(kl_clustering)
# Checking cluster labels generated for each row in the dataframe


kmeans.labels_[0:10]
```

Out[29]:

```
array([0, 0, 0, 0, 0, 1, 0, 0, 0, 0], dtype=int32)
```

In [30]:

```
# Creating a new dataframe that includes the cluster as well as the top 10 venues for each neigh
kl_merged = kl_mall.copy()


# Add the clustering labels
kl_merged["Cluster Labels"] = kmeans.labels_
kl_merged.rename(columns={"Neighborhoods": "Neighborhood"}, inplace=True)
kl_merged.head(10)
```

Out[30]:

	Neighborhood	Shopping Mall	Cluster Labels
0	Ancarrow's Landing	0	0
1	Arts District	0	0
2	Barton Heights	0	0
3	Bellemeade	0	0
4	Bellevue	0	0
5	Belmont Woods	1	1
6	Biotech and MCV	0	0
7	Blackwell	0	0
8	Brandermill	0	0
9	Brauers	0	0

In [31]:

```python
# Adding latitude and longitude values to the existing dataframe
kl_merged['Latitude'] = kl_df['Latitude']
kl_merged['Longitude'] = kl_df['Longitude']
# Sorting the results by Cluster Labels
kl_merged.sort_values(["Cluster Labels"], inplace=True)
kl_merged
```

Out[31]:

	Neighborhood	Shopping Mall	Cluster Labels	Latitude	Longitude
0	Ancarrow's Landing	0	0	37.545853	-77.442310
68	Pine Camp	0	0	37.516518	-77.455306
67	Peter Paul	0	0	37.552014	-77.536051
66	Oxford	0	0	37.555425	-77.549154
65	Oregon Hill	0	0	37.540329	-77.439526
64	Oakwood	0	0	37.479314	-77.492763
63	Oak Grove	0	0	37.539314	-77.547765
62	Northrop	0	0	37.540329	-77.439526
61	North Highland Park	0	0	37.513465	-77.476409
69	Piney Knolls	0	0	37.468794	-77.463757
58	Navy Hill	0	0	37.506365	-77.454314
56	Mosby	0	0	37.522728	-77.491616

In [32]:

```


# Creating the map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)
# Setting color scheme for the clusters
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]
# Add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(kl_merged['Latitude'], kl_merged['Longitude'], kl_merged['Neig
  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_col
map_clusters
```

Out[32]:

In [20]:

```
print(len(kl_merged.loc[kl_merged['Cluster Labels'] == 0]))
print(len(kl_merged.loc[kl_merged['Cluster Labels'] == 1]))
```

```
14
8
```

In [ ]:

```

```