🚚A Analysis for opening WholeSale Store of Raw Materials for FoodStuffs🚚

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

 # uncomment this line if you haven't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and long

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas data

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

#!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you ha
import folium # map rendering library

print('Libraries imported.')
```

```
Libraries imported.
```

In [3]:
```
data=pd.read_json("G://nyu_2451_34572-geojson.json",orient='index')
data
```

Out[3]:

	0
type	FeatureCollection
totalFeatures	306
features	[{'type': 'Feature', 'id': 'nyu_2451_34572.1',...
crs	{'type': 'name', 'properties': {'name': 'urn:o...
bbox	[-74.2492599487305, 40.5033187866211, -73.7061...

In [4]:
```
neighborhoods_data = data.iloc[2]
```

In [5]:
```
neighborhoods_data[0][0]
```

Out[5]:
```
{'type': 'Feature',
 'id': 'nyu_2451_34572.1',
 'geometry': {'type': 'Point',
  'coordinates': [-73.84720052054902, 40.89470517661]},
 'geometry_name': 'geom',
 'properties': {'name': 'Wakefield',
  'stacked': 1,
  'annoline1': 'Wakefield',
  'annoline2': None,
  'annoline3': None,
  'annoangle': 0.0,
  'borough': 'Bronx',
  'bbox': [-73.84720052054902,
   40.89470517661,
   -73.84720052054902,
   40.89470517661]}}
```

In [6]:
```
column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

# instantiate the dataframe
neighborhoods = pd.DataFrame(columns=column_names)
neighborhoods
```

Out[6]:

Borough	Neighborhood	Latitude	Longitude

In [7]:
```
for data in neighborhoods_data[0]:
    borough = neighborhood_name = data['properties']['borough']
    neighborhood_name = data['properties']['name']

    neighborhood_latlon = data['geometry']['coordinates']
    neighborhood_lat = neighborhood_latlon[1]
    neighborhood_lon = neighborhood_latlon[0]

    neighborhoods = neighborhoods.append({'Borough': borough,
                                          'Neighborhood': neighborhood_name,
                                          'Latitude': neighborhood_lat,
                                          'Longitude': neighborhood_lon}, ignore_
```

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

Out[8]:

	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

In [9]:
```
neighborhoods["Borough"].unique()
```

Out[9]:
```
array(['Bronx', 'Manhattan', 'Brooklyn', 'Queens', 'Staten Island'],
      dtype=object)
```

In [10]:
```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(neighborhoods['Borough'].unique()),
        neighborhoods.shape[0]
    )
)
```

```
The dataframe has 5 boroughs and 306 neighborhoods.
```

latitude and longitude values of New York City.

In [11]:
```
address = 'New York City, NY'
geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of New York City are {}, {}.'.format(latitude,
```

```
The geograpical coordinate of New York City are 40.7127281, -74.0060152.
```

In [12]:
```
map_newyork = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(neighborhoods['Latitude'], neighborhoo
    label = '{}, {}'.format(neighborhood, borough)
    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(map_newyork)

map_newyork
```

Out[12]:

In [13]:
```
Brooklyn_data = neighborhoods[neighborhoods['Borough'] == 'Brooklyn'].reset_index
Brooklyn_data.head()
```

Out[13]:

	Borough	Neighborhood	Latitude	Longitude
0	Brooklyn	Bay Ridge	40.625801	-74.030621
1	Brooklyn	Bensonhurst	40.611009	-73.995180
2	Brooklyn	Sunset Park	40.645103	-74.010316
3	Brooklyn	Greenpoint	40.730201	-73.954241
4	Brooklyn	Gravesend	40.595260	-73.973471

In [14]:
```
address = 'Brooklyn, NY'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Brooklyn are {}, {}.'.format(latitude, longi
```

```
The geograpical coordinate of Brooklyn are 40.6501038, -73.9495823.
```

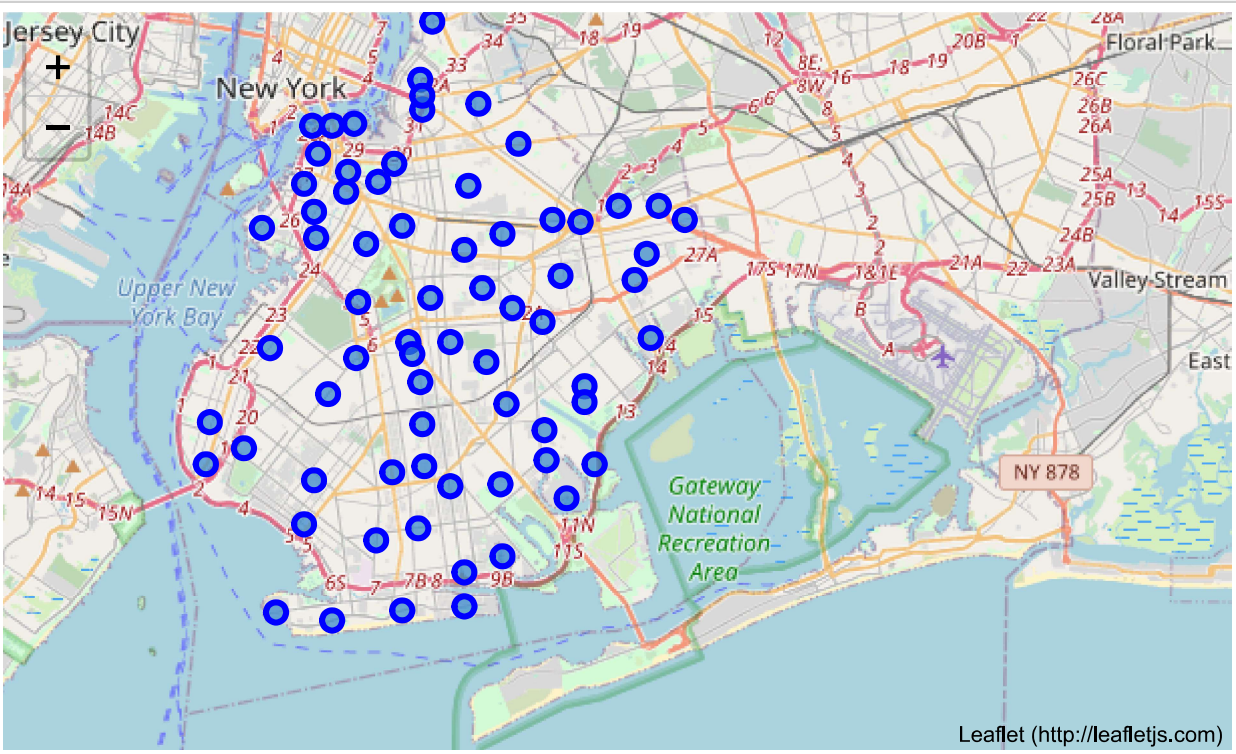
visualize Brooklyn neighborhoods.

In [15]:
```
map_Brooklyn = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(Brooklyn_data['Latitude'], Brooklyn_data['Longitude'],
    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(map_Brooklyn)

map_Brooklyn
```

Out[15]:



In [16]:
```
CLIENT_ID = 'N1UH4XLAXAKK23JT2OYNKYVJKNLKN5IJ4T5JRLVKBWE1RXCE' # your Foursquare
CLIENT_SECRET = 'GFGY513UVIVSVRRRMSYMBPN3X2YGJAY05UAGRWH3SSZ14CFP' # your Foursqu
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: N1UH4XLAXAKK23JT2OYNKYVJKNLKN5IJ4T5JRLVKBWE1RXCE
CLIENT_SECRET:GFGY513UVIVSVRRRMSYMBPN3X2YGJAY05UAGRWH3SSZ14CFP
```

first neighborhood in our dataframe.

In [17]:
```
Brooklyn_data.loc[0, 'Neighborhood']
```

Out[17]:
```
'Bay Ridge'
```

In [18]:
```
Brooklyn_data["Neighborhood"].unique()
```

Out[18]:
```
array(['Bay Ridge', 'Bensonhurst', 'Sunset Park', 'Greenpoint',
       'Gravesend', 'Brighton Beach', 'Sheepshead Bay',
       'Manhattan Terrace', 'Flatbush', 'Crown Heights', 'East Flatbush',
       'Kensington', 'Windsor Terrace', 'Prospect Heights', 'Brownsville',
       'Williamsburg', 'Bushwick', 'Bedford Stuyvesant',
       'Brooklyn Heights', 'Cobble Hill', 'Carroll Gardens', 'Red Hook',
       'Gowanus', 'Fort Greene', 'Park Slope', 'Cypress Hills',
       'East New York', 'Starrett City', 'Canarsie', 'Flatlands',
       'Mill Island', 'Manhattan Beach', 'Coney Island', 'Bath Beach',
       'Borough Park', 'Dyker Heights', 'Gerritsen Beach', 'Marine Park',
       'Clinton Hill', 'Sea Gate', 'Downtown', 'Boerum Hill',
       'Prospect Lefferts Gardens', 'Ocean Hill', 'City Line',
       'Bergen Beach', 'Midwood', 'Prospect Park South', 'Georgetown',
       'East Williamsburg', 'North Side', 'South Side', 'Ocean Parkway',
       'Fort Hamilton', 'Ditmas Park', 'Wingate', 'Rugby',
       'Remsen Village', 'New Lots', 'Paerdegat Basin', 'Mill Basin',
       'Fulton Ferry', 'Vinegar Hill', 'Weeksville', 'Broadway Junction',
       'Dumbo', 'Homecrest', 'Highland Park', 'Madison', 'Erasmus'],
      dtype=object)
```

In [19]:
```
neighborhood_latitude = Brooklyn_data.loc[0, 'Latitude'] # neighborhood latitude
neighborhood_longitude = Brooklyn_data.loc[0, 'Longitude'] # neighborhood longitu

neighborhood_name = Brooklyn_data.loc[0, 'Neighborhood'] # neighborhood name

print('Latitude and longitude values of {} are {}, {}.'.format(neighborhood_name,
                                                               neighborhood_latit
                                                               neighborhood_longi
```

```
Latitude and longitude values of Bay Ridge are 40.625801065010656, -74.03062069
353813.
```

top 100 venues that are in Marble Hill within a radius of 500 meters

In [20]:
```
LIMIT = 100
radius = 500
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
url
```

Out[20]:
```
'https://api.foursquare.com/v2/venues/explore?&client_id=N1UH4XLAXAKK23JT2OYNKY
VJKNLKN5IJ4T5JRLVKBWE1RXCE&client_secret=GFGY513UVIVSVRRRMSYMBPN3X2YGJAY05UAGRW
H3SSZ14CFP&v=20180605&ll=40.625801065010656,-74.03062069353813&radius=500&limit
=100'
```

In [21]:
```
results = requests.get(url).json()
results
```

Out[21]:

```
{'meta': {'code': 200, 'requestId': '5f3c9fcf30e7ef00416d8244'},
 'response': {'suggestedFilters': {'header': 'Tap to show:',
   'filters': [{'name': '$-$$$$', 'key': 'price'},
    {'name': 'Open now', 'key': 'openNow'}]},
  'headerLocation': 'Bay Ridge',
  'headerFullLocation': 'Bay Ridge, Brooklyn',
  'headerLocationGranularity': 'neighborhood',
  'totalResults': 78,
  'suggestedBounds': {'ne': {'lat': 40.63030106951066,
    'lng': -74.02470273356597},
   'sw': {'lat': 40.62130106051065, 'lng': -74.03653865351028}},
  'groups': [{'type': 'Recommended Places',
    'name': 'recommended',
    'items': [{'reasons': {'count': 0,
       'items': [{'summary': 'This spot is popular',
         'type': 'general',
         'reasonName': 'globalInteractionReason'}]},
      'venue': {'id': '4b895827f964a5206c2d32e3',
       'name': 'Pilo Arts Day Spa and Salon',
       'location': {'address': '8413 3rd Ave',
```


extracts the category of the venue

In [22]:
```
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

In [23]:
```
venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venu
nearby_venues =nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues
```

```
C:\Users\mahesh venkat reesu\anaconda3\lib\site-packages\ipykernel_launcher.
py:3: FutureWarning: pandas.io.json.json_normalize is deprecated, use panda
s.json_normalize instead
  This is separate from the ipykernel package so we can avoid doing imports
until
```

In [24]:
```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```

```
78 venues were returned by Foursquare.
```

Explore Neighborhoods in Brooklyn

In [25]:
```
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_
            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']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

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

    return(nearby_venues)
```

Brooklyn_venues

In [26]:
```
Brooklyn_venues = getNearbyVenues(names=Brooklyn_data['Neighborhood'],
                                  latitudes=Brooklyn_data['Latitude'],
                                  longitudes=Brooklyn_data['Longitude']
                                 )
```

```
Bay Ridge
Bensonhurst
Sunset Park
Greenpoint
Gravesend
Brighton Beach
Sheepshead Bay
Manhattan Terrace
Flatbush
Crown Heights
East Flatbush
Kensington
Windsor Terrace
Prospect Heights
Brownsville
Williamsburg
Bushwick
Bedford Stuyvesant
Brooklyn Heights
Cobble Hill
Carroll Gardens
Red Hook
Gowanus
Fort Greene
Park Slope
Cypress Hills
East New York
Starrett City
Canarsie
Flatlands
Mill Island
Manhattan Beach
Coney Island
Bath Beach
Borough Park
Dyker Heights
Gerritsen Beach
Marine Park
Clinton Hill
Sea Gate
Downtown
Boerum Hill
Prospect Lefferts Gardens
Ocean Hill
City Line
Bergen Beach
Midwood
Prospect Park South
Georgetown
East Williamsburg
North Side
```

```
South Side
Ocean Parkway
Fort Hamilton
Ditmas Park
Wingate
Rugby
Remsen Village
New Lots
Paerdegat Basin
Mill Basin
Fulton Ferry
Vinegar Hill
Weeksville
Broadway Junction
Dumbo
Homecrest
Highland Park
Madison
Erasmus
```

In [27]:

```
print(Brooklyn_venues.shape)
Brooklyn_venues.head()
```

```
(2770, 7)
```

Out[27]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Bay Ridge	40.625801	-74.030621	Pilo Arts Day Spa and Salon	40.624748	-74.030591	Spa
1	Bay Ridge	40.625801	-74.030621	Bagel Boy	40.627896	-74.029335	Bagel Shop
2	Bay Ridge	40.625801	-74.030621	Pegasus Cafe	40.623168	-74.031186	Breakfast Spot
3	Bay Ridge	40.625801	-74.030621	Cocoa Grinder	40.623967	-74.030863	Juice Bar
4	Bay Ridge	40.625801	-74.030621	Leo's Casa Calamari	40.624200	-74.030931	Pizza Place

In [131]:
```
count=Brooklyn_venues.groupby('Neighborhood').count()
count
```

Out[131]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Bath Beach	48	48	48	48	48	48
Bay Ridge	78	78	78	78	78	78
Bedford Stuyvesant	29	29	29	29	29	29
Bensonhurst	32	32	32	32	32	32
Bergen Beach	5	5	5	5	5	5
Boerum Hill	88	88	88	88	88	88
Borough Park	26	26	26	26	26	26
Brighton Beach	44	44	44	44	44	44
Broadway Junction	15	15	15	15	15	15

Count of Venues for each neighborhood

In [29]:
```
print('There are {} uniques categories.'.format(len(Brooklyn_venues['Venue Catego

print('Here is the list of different categories:')
list(Brooklyn_venues['Venue Category'].unique())
```

```
There are 286 uniques categories.
Here is the list of different categories:
```

Analyze Each Neighborhood by one hot encoding

In [76]:
```
Brooklyn_onehot = pd.get_dummies(Brooklyn_venues[['Venue Category']], prefix="",

# add neighborhood column back to dataframe
Brooklyn_onehot['Neighborhood'] = Brooklyn_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [Brooklyn_onehot.columns[-1]] + list(Brooklyn_onehot.columns[:-1]
Brooklyn_onehot = Brooklyn_onehot[fixed_columns]

Brooklyn_onehot.head()
```

Out[76]:

	Yoga Studio	Accessories Store	Adult Boutique	American Restaurant	Antique Shop	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Arts & Crafts Store	Er
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	

In [121]:
```
Brooklyn_onehot.shape
```

Out[121]: (2770, 286)

In [122]:
```
Brooklyn_grouped = Brooklyn_onehot.groupby('Neighborhood').mean().reset_index()
Brooklyn_grouped
```

Out[122]:

	Neighborhood	Yoga Studio	Accessories Store	Adult Boutique	American Restaurant	Antique Shop	Arepa Restaurant	Argentir Restau
0	Bath Beach	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000
1	Bay Ridge	0.000000	0.000000	0.00	0.038462	0.000000	0.00	0.000
2	Bedford Stuyvesant	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000
3	Bensonhurst	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000
4	Bergen Beach	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000
5	Boerum Hill	0.022727	0.000000	0.00	0.011364	0.011364	0.00	0.000
6	Borough Park	0.000000	0.000000	0.00	0.038462	0.000000	0.00	0.000
7	Brighton Beach	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000

Removing unnecessary Neighbourhood Venues for WareHouse Owner

In [123]:
```
Brooklyn_grouped.drop(["Yoga Studio","Accessories Store","Adult Boutique","Antiqu
                       ,"Arts & Entertainment","Athletics & Sports","Auto Garage",
                      ,"Baseball Stadium","Basketball Court","Beach","Bike Rental
                      "Board Shop","Boat or Ferry","Boat or Ferry","Bookstore","Bo
                      ,"Bus Station","Bus Stop","Business Service","Butcher","Chil
                      ,"Clothing Store","Concert Hall","Construction & Landscaping
                      ,"Creperie","Cycle Studio","Dance Studio","Deli / Bodega","E
                      ,"Event Service","Event Space","Gaming Cafe","Gaming Cafe","
                       "Gastropub","Gay Bar","General Entertainment","Gift Shop","
                      ,"Gym","Gym / Fitness Center","Gymnastics Gym","Harbor / Mar
                       ,"Hobby Shop","Home Service","Hookah Bar","Indie Movie Thea
                      ,"Jazz Club","Jewelry Store","Laundry Service","Library","Li
                      ,"Martial Arts School","Massage Studio","Mattress Store","Me
                       "Mobile Phone Shop","Monument / Landmark","Motorcycle Shop"
                       "Music Store","Music Venue","Nail Salon","Nightclub","Non-P
                       "Organic Grocery","Other Great Outdoors","Other Repair Shop
                      ,"Paper / Office Supplies Store","Park","Performing Arts Ver
                       ,"Photography Studio","Piercing Parlor","Pilates Studio","P
                      ,"Pool","Pool Hall","Print Shop","Pub","Racetrack","Record S
                      ,"Rock Club","Roller Rink","Roof Deck","Salon / Barbershop",
                       "Sculpture Garden","Shoe Store","Shopping Mall","Skating Ri
                       "Soccer Field","Soup Place","Speakeasy","Sporting Goods Sho
                      ,"Theater","Theme Park Ride / Attraction","Thrift / Vintage
                      ,"Video Game Store","Video Store","Vietnamese Restaurant","W
                       "Wine Shop","Wings Joint","Women's Store"],axis=1,inplace=T
```

In [124]:
```
Brooklyn_grouped
```

Out[124]:

	Neighborhood	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant	BBQ Joint	Bakery	Bangla Resta
0	Bath Beach	0.000000	0.00	0.000000	0.020833	0.000000	0.020833	0.0
1	Bay Ridge	0.038462	0.00	0.000000	0.000000	0.000000	0.000000	0.0
2	Bedford Stuyvesant	0.000000	0.00	0.000000	0.000000	0.034483	0.000000	0.0
3	Bensonhurst	0.000000	0.00	0.000000	0.000000	0.000000	0.031250	0.0
4	Bergen Beach	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.0
5	Boerum Hill	0.011364	0.00	0.000000	0.000000	0.000000	0.034091	0.0
6	Borough Park	0.038462	0.00	0.000000	0.000000	0.000000	0.038462	0.0
7	Brighton Beach	0.000000	0.00	0.000000	0.000000	0.000000	0.022727	0.0

Integrating Different Restaurants and Different Joints

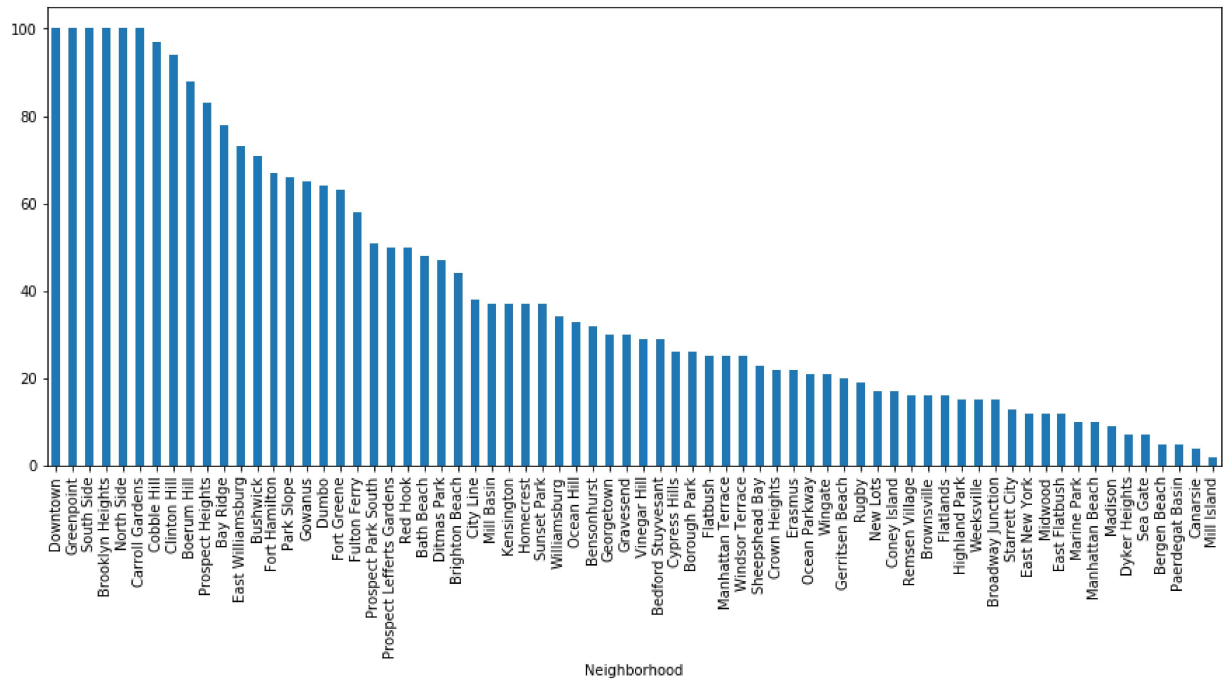
(Assuming Different Resaturants Use the Same Raw Groceries)

This Assumption is made for simplicity and due to not having very large dataset about neighborhoods

In [133]:
```
count.sort_values(by="Venue",ascending=False,axis=0,inplace=True)
```

In [135]:
```
count["Venue"].plot(kind="bar",figsize=(15,6))
```

Out[135]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x20892cbd488>
```



In [80]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

In [99]:
```
num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = Brooklyn_grouped['Neighborhood']

for ind in np.arange(Brooklyn_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(Brookly

neighborhoods_venues_sorted.head()
```

Out[99]:

	Neighborhood	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	Bath Beach	Pizza Place	Donut Shop	Bubble Tea Shop	Sushi Restaurant	Chinese Restaurant	Italian Restaurant	Fast Food Restaurant	R
1	Bay Ridge	Italian Restaurant	Spa	Pizza Place	Greek Restaurant	Bar	American Restaurant	Café	I
2	Bedford Stuyvesant	Coffee Shop	Café	Bar	Pizza Place	Italian Restaurant	New American Restaurant	Juice Bar	
3	Bensonhurst	Chinese Restaurant	Donut Shop	Pizza Place	Ice Cream Shop	Sushi Restaurant	Italian Restaurant	Bakery	
4	Bergen Beach	Vegetarian / Vegan Restaurant	Fast Food Restaurant	Diner	Discount Store	Distillery	Dive Bar	Dog Run	

In [100]:
```
kclusters = 5

Brooklyn_grouped_clustering = Brooklyn_grouped.drop('Neighborhood', 1)

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

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

Out[100]:
```
array([3, 1, 1, 3, 1, 1, 3, 1, 3, 1])
```

In [101]:
```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

Brooklyn_merged = Brooklyn_data

# merge toronto_grouped with toronto_data to add latitude/longitude for each neig
Brooklyn_merged = Brooklyn_merged.join(neighborhoods_venues_sorted.set_index('Nei

Brooklyn_merged.head() # check the last columns!
```

Out[101]:

	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Com V
0	Brooklyn	Bay Ridge	40.625801	-74.030621	1	Italian Restaurant	Spa	Pizza Place	(Resta
1	Brooklyn	Bensonhurst	40.611009	-73.995180	3	Chinese Restaurant	Donut Shop	Pizza Place	Ice C
2	Brooklyn	Sunset Park	40.645103	-74.010316	1	Mexican Restaurant	Pizza Place	Latin American Restaurant	B
3	Brooklyn	Greenpoint	40.730201	-73.954241	1	Bar	Pizza Place	Cocktail Bar	C
4	Brooklyn	Gravesend	40.595260	-73.973471	1	Pizza Place	Italian Restaurant	Bakery	

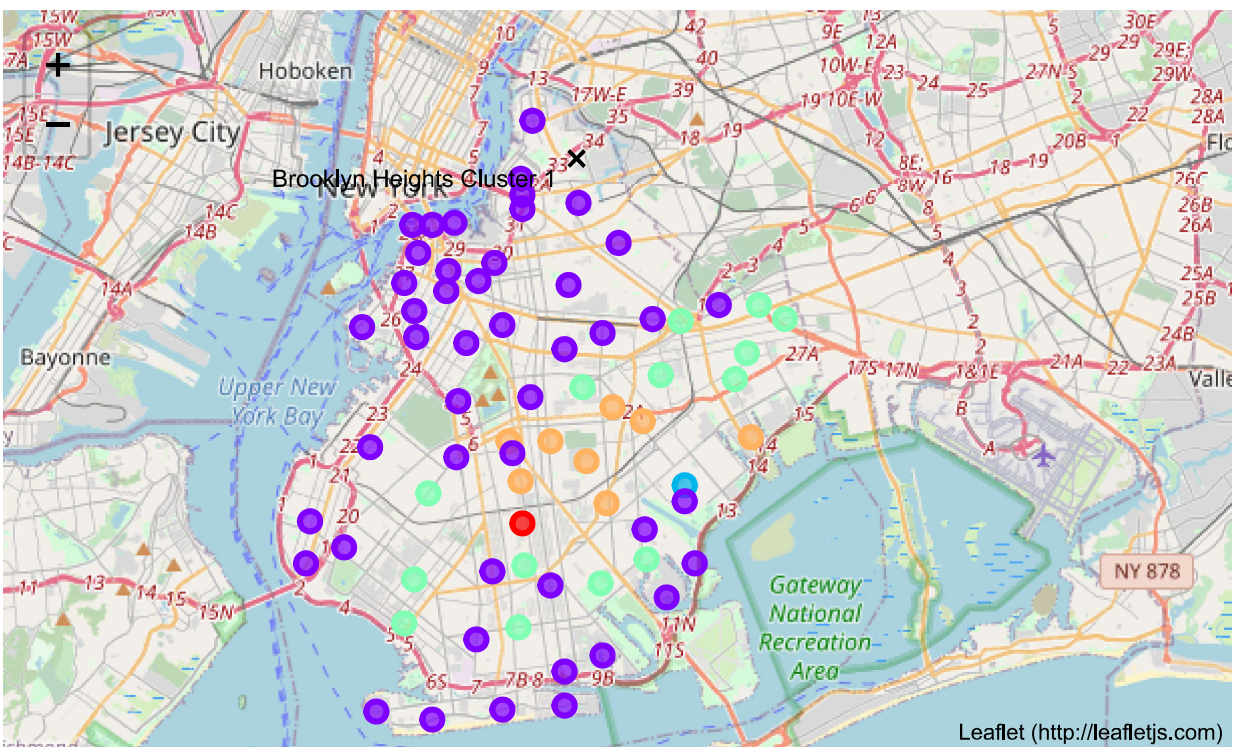
In [102]:
```
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set 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(Brooklyn_merged['Latitude'], Brooklyn_merged['L
    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[102]:



CLUSTER 0:

In [103]:
```
Brooklyn_merged.loc[Brooklyn_merged['Cluster Labels'] == 0, Brooklyn_merged.colum
```

Out[103]:

	Neighborhood	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	8th Cor V
46	Midwood	Pizza Place	Bakery	Convenience Store	Candy Store	Ice Cream Shop	Field	Grocery Store	F Resta

CLUSTER 1:

In [104]:
```
Brooklyn_merged.loc[Brooklyn_merged['Cluster Labels'] == 1, Brooklyn_merged.colum
```

Out[104]:

	Neighborhood	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 Com V
0	Bay Ridge	Italian Restaurant	Spa	Pizza Place	Greek Restaurant	Bar	American Restaurant	
2	Sunset Park	Mexican Restaurant	Pizza Place	Latin American Restaurant	Bakery	Bank	Fried Chicken Joint	Brea
3	Greenpoint	Bar	Pizza Place	Cocktail Bar	Coffee Shop	Mexican Restaurant	French Restaurant	F Resta
4	Gravesend	Pizza Place	Italian Restaurant	Bakery	Bar	Farmers Market	Breakfast Spot	Chi Resta
5	Brighton Beach	Russian Restaurant	Eastern European Restaurant	Restaurant	Sushi Restaurant	Bank	Fast Food Restaurant	Gr

CLUSTER 2:

In [105]:
```
Brooklyn_merged.loc[Brooklyn_merged['Cluster Labels'] == 2, Brooklyn_merged.colum
```

Out[105]:

	Neighborhood	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	8 C
28	Canarsie	Chinese Restaurant	Asian Restaurant	Caribbean Restaurant	Dumpling Restaurant	Farm	Falafel Restaurant	Factory	I Re

CLUSTER 3:

In [106]:
```
Brooklyn_merged.loc[Brooklyn_merged['Cluster Labels'] == 3, Brooklyn_merged.colun
```

Out[106]:

	Neighborhood	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 Mc Comm Ven
1	Bensonhurst	Chinese Restaurant	Donut Shop	Pizza Place	Ice Cream Shop	Sushi Restaurant	Italian Restaurant	Bake
7	Manhattan Terrace	Ice Cream Shop	Pizza Place	Donut Shop	Burger Joint	Bank	Chinese Restaurant	Coffee Sh
14	Brownsville	Chinese Restaurant	Restaurant	Fried Chicken Joint	Farmers Market	Spanish Restaurant	Pizza Place	Burger Jo
25	Cypress Hills	Pizza Place	Fried Chicken Joint	Ice Cream Shop	Chinese Restaurant	Donut Shop	Fast Food Restaurant	Discou Sto
26	East New York	Spanish Restaurant	Pizza Place	Food Truck	Fast Food Restaurant	Caribbean Restaurant	Chinese Restaurant	Fri Chick Jo
33	Bath Beach	Pizza Place	Donut Shop	Bubble Tea Shop	Sushi Restaurant	Chinese Restaurant	Italian Restaurant	Fast Fo Restaura
34	Borough Park	Bank	Pizza Place	Fast Food Restaurant	Café	Hostel	Chinese Restaurant	Easte Europe Restaura
37	Marine Park	Pizza Place	Ice Cream Shop	Chinese Restaurant	Dry Cleaner	Falafel Restaurant	Factory	Ethiopi Restaura
44	City Line	Donut Shop	Fried Chicken Joint	Grocery Store	Bakery	Spanish Restaurant	Food Truck	Ice Crea Sh
55	Wingate	Fried Chicken Joint	Pizza Place	Fish & Chips Shop	Discount Store	Convenience Store	Donut Shop	Juice E
58	New Lots	Fried Chicken Joint	Pizza Place	Chinese Restaurant	Discount Store	Grocery Store	Asian Restaurant	Spani Restaura
60	Mill Basin	Chinese Restaurant	Pizza Place	Japanese Restaurant	Bank	Sushi Restaurant	American Restaurant	Supermark
64	Broadway Junction	Diner	Donut Shop	Fried Chicken Joint	Hotel	Caribbean Restaurant	Burger Joint	Ice Crea Sh
66	Homecrest	Bank	Donut Shop	Mexican Restaurant	Pizza Place	Chinese Restaurant	Grocery Store	Sus Restaura

CLUSTER 4:

In [107]:
```
Brooklyn_merged.loc[Brooklyn_merged['Cluster Labels'] == 4, Brooklyn_merged.colum
```

Out[107]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7 C
8	Flatbush	Coffee Shop	Caribbean Restaurant	Mexican Restaurant	Chinese Restaurant	Middle Eastern Restaurant	Pizza Place	
10	East Flatbush	Department Store	Food & Drink Shop	Chinese Restaurant	Caribbean Restaurant	Supermarket	Fast Food Restaurant	
27	Starrett City	American Restaurant	Supermarket	Donut Shop	Chinese Restaurant	Caribbean Restaurant	Pizza Place	
29	Flatlands	Caribbean Restaurant	Fried Chicken Joint	Fast Food Restaurant	Bar	Discount Store	Dry Cleaner	
47	Prospect Park South	Pizza Place	Caribbean Restaurant	Grocery Store	Fast Food Restaurant	Latin American Restaurant	Donut Shop	Re
56	Rugby	Bank	Grocery Store	Caribbean Restaurant	Pizza Place	Chinese Restaurant	Seafood Restaurant	
57	Remsen Village	Caribbean Restaurant	Fast Food Restaurant	Supermarket	Donut Shop	Café	Breakfast Spot	
69	Erasmus	Caribbean Restaurant	Furniture / Home Store	Bank	Supermarket	Chinese Restaurant	Convenience Store	

RESULT:

Best Cluster is C1(Upper Violet)---Since all top venues according to bar graph are located in upper cluster 1.

Second Best Cluster is C4.

THANK YOU!!!!!!!!!!!!!!!!!!!!

In [ ]:

In [ ]: