DSC 680 Project 3

June 5, 2021

1 Caleb Corpuz

2 DSC 680 Project 3

2.0.1 Things to do

- figure out haversine distance(distance between two points on a sphere given their lat/lon)
  - there is a haversine library
- Filter make filtes or indicators for common categories
  - Example: if restaurant in categories, the bus_df[‘restaurant’] =1

[ ]:
```
# geopy documentation : https://geopy.readthedocs.io/en/stable/#
# haversine documentation: https://pypi.org/project/haversine/
```

[30]:
```
import json
import numpy as np
import pandas as pd
from geopy.geocoders import Nominatim
import matplotlib.pyplot as plt
import descartes
import geopandas as gpd
from shapely.geometry import Point, Polygon
from haversine import haversine, Unit

%matplotlib inline
```

[2]:
```
data_file_bus = open("/Users/ccorpuz/Desktop/yelp_dataset/
 ↪yelp_academic_dataset_business.json")
data_bus = []

for line in data_file_bus:
    data_bus.append(json.loads(line))

bus_df = pd.DataFrame(data_bus)
data_file_bus.close()
```

[3]:
```
bus_df = bus_df.dropna()
```

[4]:
```
bus_df.columns
```

[4]:
```
Index(['business_id', 'name', 'address', 'city', 'state', 'postal_code',
       'latitude', 'longitude', 'stars', 'review_count', 'is_open',
       'attributes', 'categories', 'hours'],
      dtype='object')
```

[5]:
```
## Isolate the businesses that are restaurants
```

[6]:
```
# convert the category field into a list. Currently in string format

def make_cat_list(data):
    x = data.split(",")
    y = []
    for i in x:
        cat = i.strip()
        y.append(cat)
    return y

bus_df['cat_list'] = bus_df['categories'].apply(lambda x: make_cat_list(x))
```

[7]:
```
# Create an indicator for restaurant

def is_restaurant(data):
    if "Restaurants" in data:
        return 1
    else:
        return 0

bus_df['restaurant_ind'] = bus_df['cat_list'].apply(lambda x: is_restaurant(x))
```

[8]:
```
bus_df.head(3)
```

[8]:
```
              business_id                      name              address  \
0  6iYb2HFDywm3zjuRg0shjw      Oskar Blues Taproom          921 Pearl St
1  tCbdrRPZA0oiIYSmHG3J0w  Flying Elephants at PDX  7000 NE Airport Way
2  bvN78flM8NLprQ1a1y5dRg           The Reclaimory   4720 Hawthorne Ave

       city state postal_code   latitude   longitude  stars  review_count  \
0   Boulder    CO       80302  40.017544 -105.283348    4.0            86
1  Portland    OR       97218  45.588906 -122.593331    4.0           126
2  Portland    OR       97214  45.511907 -122.613693    4.5            13

   is_open                                         attributes  \
0        1  {'RestaurantsTableService': 'True', 'WiFi': 'u…
1        1  {'RestaurantsTakeOut': 'True', 'RestaurantsAtt…
2        1  {'BusinessAcceptsCreditCards': 'True', 'Restau…
```

```
                                            categories  \
0  Gastropubs, Food, Beer Gardens, Restaurants, B…
1  Salad, Soup, Sandwiches, Delis, Restaurants, C…
2  Antiques, Fashion, Used, Vintage & Consignment…

                                               hours  \
0  {'Monday': '11:0-23:0', 'Tuesday': '11:0-23:0'…
1  {'Monday': '5:0-18:0', 'Tuesday': '5:0-17:0', …
2  {'Thursday': '11:0-18:0', 'Friday': '11:0-18:0…

                                           cat_list  restaurant_ind
0  [Gastropubs, Food, Beer Gardens, Restaurants, …               1
1  [Salad, Soup, Sandwiches, Delis, Restaurants, …               1
2  [Antiques, Fashion, Used, Vintage & Consignmen…               0
```

[9]:
```
rest_df = bus_df[bus_df['restaurant_ind'] == 1]
rest_df = rest_df[rest_df['stars'] >= 4.0] # only include businesses with good␣
 ↪reviews
rest_df = rest_df.reset_index(drop = True)
```

[10]:
```
rest_df.head(3)
```

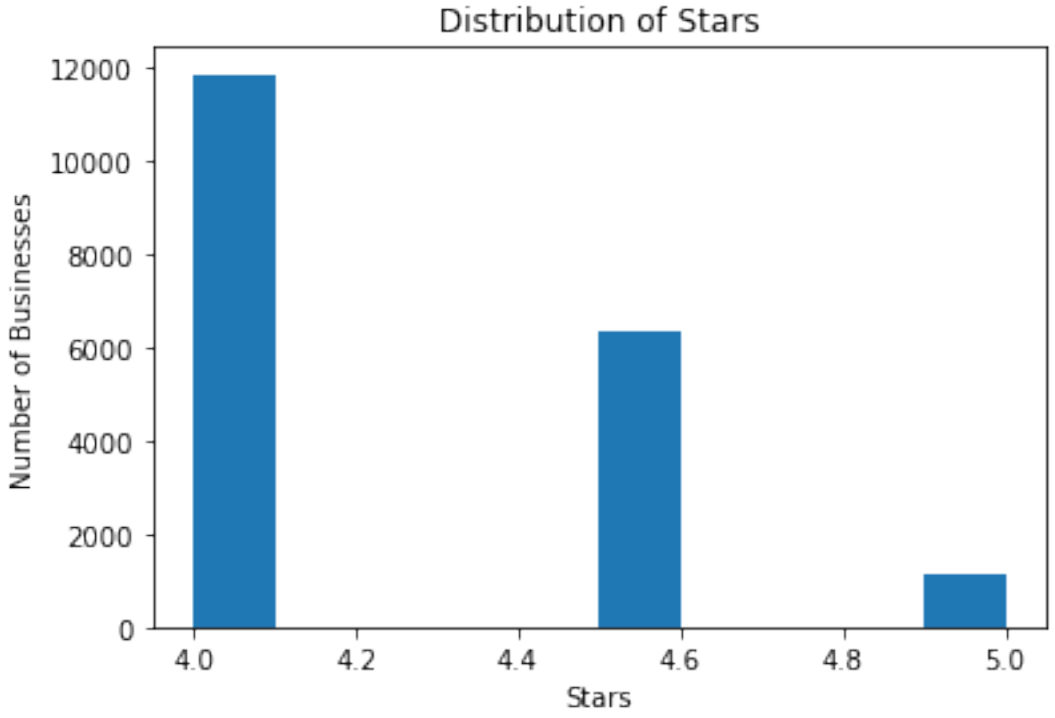
[10]:
```
              business_id                      name              address  \
0  6iYb2HFDywm3zjuRg0shjw      Oskar Blues Taproom          921 Pearl St
1  tCbdrRPZA0oiIYSmHG3J0w  Flying Elephants at PDX  7000 NE Airport Way
2  HPA_qyMEddpAEtFof02ixg      Mr G's Pizza & Subs         474 Lowell St

       city state postal_code   latitude   longitude  stars  review_count  \
0   Boulder    CO       80302  40.017544 -105.283348    4.0            86
1  Portland    OR       97218  45.588906 -122.593331    4.0           126
2   Peabody    MA       01960  42.541155  -70.973438    4.0            39

   is_open                                         attributes  \
0        1  {'RestaurantsTableService': 'True', 'WiFi': 'u…
1        1  {'RestaurantsTakeOut': 'True', 'RestaurantsAtt…
2        1  {'RestaurantsGoodForGroups': 'True', 'HasTV': …

                                            categories  \
0  Gastropubs, Food, Beer Gardens, Restaurants, B…
1  Salad, Soup, Sandwiches, Delis, Restaurants, C…
2                          Food, Pizza, Restaurants

                                               hours  \
0  {'Monday': '11:0-23:0', 'Tuesday': '11:0-23:0'…
1  {'Monday': '5:0-18:0', 'Tuesday': '5:0-17:0', …
2  {'Monday': '11:0-21:0', 'Tuesday': '11:0-21:0'…
```

```
                                                  cat_list  restaurant_ind
0  [Gastropubs, Food, Beer Gardens, Restaurants, …                1
1  [Salad, Soup, Sandwiches, Delis, Restaurants, …                1
2                          [Food, Pizza, Restaurants]                1
```

2.1 EDA

2.1.1 Distribution of Stars

[11]:
```
plt.hist(rest_df['stars'])
plt.title("Distribution of Stars")
plt.xlabel("Stars")
plt.ylabel("Number of Businesses")
plt.show()
```



2.1.2 States

[12]:
```
print('States and Number of Businesses:')
rest_df['state'].value_counts()
```

```
States and Number of Businesses:
```

[12]:
```
OR     3670
MA     3569
FL     3099
TX     2435
BC     2334
GA     2040
OH     1532
CO      341
WA      281
ABE       1
NH        1
Name: state, dtype: int64
```

2.1.3 Common Cities

[13]:
```
print('Top 10 Most Common Cities')
rest_df['city'].value_counts().head(10)
```

```
Top 10 Most Common Cities
```

[13]:
```
Portland     3060
Austin       2267
Vancouver    1714
Orlando      1518
Atlanta      1405
Columbus      974
Boston        952
Cambridge     287
Boulder       262
Kissimmee     226
Name: city, dtype: int64
```

2.1.4 Common Categories

[14]:
```
cat = []
distinct_cat = []

for i in rest_df['categories']:
    x = i.split(",")
    for category in x:
        k = category.strip() # have to use this because some categories have a␣
 ↪whitespace in the beginning
        cat.append(k)

for i in cat:
    if i not in distinct_cat:
        distinct_cat.append(i)
```

```
    else:
        continue

cat_pd = pd.DataFrame(cat, columns=["Category"])

print("The top 10 most frequent categories for restaurants are:")
cat_pd.value_counts().head(10)
```

```
The top 10 most frequent categories for restaurants are:
```

[14]:
```
Category
Restaurants               19303
Food                       8140
Nightlife                  3524
Bars                       3392
Sandwiches                 3032
Breakfast & Brunch         2504
American (New)             2273
Coffee & Tea               2129
American (Traditional)     1960
Pizza                      1748
dtype: int64
```

2.1.5 Attributes

- Perform same analysis on attributes that we did on categories

[15]:
```
key_list = []

for i in rest_df['attributes']:
    keys = i.keys()
    for k in keys:
        key_list.append(k)

key_pd = pd.DataFrame(key_list, columns = ['Keys'])

print("The top 10 most common attributes are:")
key_pd['Keys'].value_counts().head(10)
```

```
The top 10 most common attributes are:
```

[15]:
```
RestaurantsTakeOut            18425
RestaurantsDelivery           18056
BusinessParking               17976
OutdoorSeating                17129
RestaurantsReservations       16604
HasTV                         16552
RestaurantsPriceRange2        16365
```

```
BusinessAcceptsCreditCards    16248
Ambience                      16166
Alcohol                       16023
Name: Keys, dtype: int64
```

[16]:
```
# this can be used to retrieve an address given coordinates

# from functools import partial
# reverse = partial(geolocator.reverse, language="es")
# print(reverse("49.2779085, -123.10894170979472"))
```

2.2 Create the user program

[ ]:
```
# 800 Griffiths Way Vancouver BC is rogers arena in Vancoucer BC
# use this as an example because BC is one of the places where we have a lot of␣
 ↪restaurants
```

[17]:
```
user_address = input("Hello, please enter an address: ")
```

```
Hello, please enter an address: 800 Griffiths Way Vancouver BC
```

[18]:
```
geolocator = Nominatim(user_agent = "user_geo")
user_coord = geolocator.geocode(user_address)
print(user_coord.address)
print("Coordinates:")
print(user_coord.latitude, user_coord.longitude)
user_location  = (user_coord.latitude, user_coord.longitude)
```

```
Rogers Arena, 800, Griffiths Way, Gastown, Downtown, Vancouver, District of
North Vancouver, Metro Vancouver Regional District, British Columbia, V6B 1V4,
Canada
Coordinates:
49.2779085 -123.10894170979472
```

[56]:
```
distances = []

for i in range(len(rest_df)):
        restaurant_location = (rest_df['latitude'][i], rest_df['longitude'][i])
        restaurant_distance = haversine(user_location, restaurant_location,␣
 ↪unit=Unit.MILES)
        restaurant_data = {'business_id' : rest_df['business_id'][i], 'name':␣
 ↪rest_df['name'][i],
                           'stars' :rest_df['stars'][i], 'distance_to_user':␣
 ↪round(restaurant_distance, 3),
                           'address': (rest_df['address'][i] + ", " +␣
 ↪rest_df['city'][i] + ", " +
```

```
                          rest_df['state'][i] + " " +␣
↪rest_df['postal_code'][i]),
                          'lat' : rest_df['latitude'][i], 'lon':␣
↪rest_df['longitude'][i]}
        distances.append(restaurant_data)

distances.append({'business_id': 'userlocation', 'name': 'You are here',␣
↪'stars': 5.0,
                'distance_to_user': 0, 'address': user_address, 'lat':␣
↪user_coord.latitude,
                'lon': user_coord.longitude})

output_pd = pd.DataFrame(distances)

def location_indicator(data):
    if data == 'userlocation':
        return 'userlocation'
    else:
        return 'recommendation'

output_pd['location_ind'] = output_pd['business_id'].apply(lambda x:␣
↪location_indicator(x))
```

[57]:
```
top_ten = output_pd.sort_values(by = 'distance_to_user').head(11)
```

[58]:
```
top_ten
```

[58]:

	business_id	name	stars	\
19303	userlocation	You are here	5.0	
10958	z2tHNBxoQ-_CoczbhzLNrQ	TAKO	4.0	
12668	dMMu2LgKX_Cg-SwlRV-PBw	Odo Sushi	4.0	
5557	sUxHZhLvTrLE4eoewdv-ug	Pizzeria Ludica	4.0	
14738	7Y9dgVNnW6gxeEkdK-gzgg	Papparoti	4.5	
17503	Ebu8VPDeehcRbxzaDN5fDg	Charisma Cafe & Dessert House	4.0	
6159	LjdbthVdtLYKSi7iVAFl0g	Jam Cafe on Beatty	4.5	
11677	_4R46MNkwx9MeOyt0YfNxA	Chambar	4.0	
4570	XAml4XXnE1a6RZuGmxpbiA	The Dirty Apron Cooking School	4.5	
574	cLGh_q9jWTpp53tsj29S-w	The Dirty Apron Delicatessen	4.5	
1841	DBnvmIwc6H0BC7JeZlxbbw	Kim Son	4.0	

	distance_to_user	address	\
19303	0.000	800 Griffiths Way Vancouver BC	
10958	0.075	601 Expo Boulevard, Vancouver, BC V6B 0J5	
12668	0.135	82 keefer Place, Vancouver, BC V6B 6C1	
5557	0.149	189 Keefer Place, Vancouver, BC V6B 6L4	
14738	0.150	193 Keefer Place, Unit 103, Vancouver, BC V6B 6C1	
17503	0.152	181 Keefer Place, Unit 101, Vancouver, BC V6B 6C1	

```
6159              0.158           556 Beatty Street, Vancouver, BC V6B 2L3
11677             0.161           568 Beatty Street, Vancouver, BC V6B 2L3
4570              0.180           540 Beatty Street, Vancouver, BC V6B 2L3
574               0.185               540 Beatty St, Vancouver, BC V6B 2L3
1841              0.192  88 W Pender Street, Unit 2019, Vancouver, BC V…

             lat         lon    location_ind
19303  49.277909 -123.108942    userlocation
10958  49.278883 -123.108234  recommendation
12668  49.279475 -123.107134  recommendation
5557   49.280043 -123.108487  recommendation
14738  49.280069 -123.108635  recommendation
17503  49.280076 -123.108333  recommendation
6159   49.280155 -123.109542  recommendation
11677  49.280146 -123.109925  recommendation
4570   49.280506 -123.109143  recommendation
574    49.280578 -123.109191  recommendation
1841   49.280290 -123.106736  recommendation
```

[70]:
```
from bokeh.io import output_notebook, show, output_file
from bokeh.plotting import figure, ColumnDataSource
from bokeh.tile_providers import get_provider, Vendors
from bokeh.palettes import PRGn, RdYlGn
from bokeh.transform import CategoricalColorMapper,factor_cmap
from bokeh.layouts import row, column
from bokeh.models import GeoJSONDataSource, LinearColorMapper, ColorBar,␣
 ↪NumeralTickFormatter, CategoricalColorMapper
import numpy as np
import pandas as pd
```

[71]:
```
# Define function to switch from lat/long to mercator coordinates
def x_coord(x, y):

    lat = x
    lon = y

    r_major = 6378137.000
    x = r_major * np.radians(lon)
    scale = x/lon
    y = 180.0/np.pi * np.log(np.tan(np.pi/4.0 +
        lat * (np.pi/180.0)/2.0)) * scale
    return (x, y)
# Define coord as tuple (lat,long)

top_ten['coordinates'] = list(zip(top_ten['lat'], top_ten['lon']))
# Obtain list of mercator coordinates
mercators = [x_coord(x, y) for x, y in top_ten['coordinates'] ]
```

[72]:
```
# Create mercator column in our df
top_ten['mercator'] = mercators

# Split that column out into two separate columns - mercator_x and mercator_y
top_ten[['mercator_x', 'mercator_y']] = top_ten['mercator'].apply(pd.Series)
```

[81]:
```
top_ten
```

[81]:

	business_id	name	stars	distance_to_user	address	lat	lon	location_ind
19303	userlocation	You are here	5.0	0.000	800 Griffiths Way Vancouver BC	49.277909	-123.108942	userlocation
10958	z2tHNBxoQ-_CoczbhzLNrQ	TAKO	4.0	0.075	601 Expo Boulevard, Vancouver, BC V6B 0J5	49.278883	-123.108234	recommendation
12668	dMMu2LgKX_Cg-SwlRV-PBw	Odo Sushi	4.0	0.135	82 keefer Place, Vancouver, BC V6B 6C1	49.279475	-123.107134	recommendation
5557	sUxHZhLvTrLE4eoewdv-ug	Pizzeria Ludica	4.0	0.149	189 Keefer Place, Vancouver, BC V6B 6L4	49.280043	-123.108487	recommendation
14738	7Y9dgVNnW6gxeEkdK-gzgg	Papparoti	4.5	0.150	193 Keefer Place, Unit 103, Vancouver, BC V6B 6C1	49.280069	-123.108635	recommendation
17503	Ebu8VPDeehcRbxzaDN5fDg	Charisma Cafe & Dessert House	4.0	0.152	181 Keefer Place, Unit 101, Vancouver, BC V6B 6C1	49.280076	-123.108333	recommendation
6159	LjdbthVdtLYKSi7iVAFlOg	Jam Cafe on Beatty	4.5	0.158	556 Beatty Street, Vancouver, BC V6B 2L3	49.280155	-123.109542	recommendation
11677	_4R46MNkwx9MeOyt0YfNxA	Chambar	4.0	0.161	568 Beatty Street, Vancouver, BC V6B 2L3	49.280146	-123.109925	recommendation
4570	XAml4XXnE1a6RZuGmxpbiA	The Dirty Apron Cooking School	4.5	0.180	540 Beatty Street, Vancouver, BC V6B 2L3	49.280506	-123.109143	recommendation
574	cLGh_q9jWTpp53tsj29S-w	The Dirty Apron Delicatessen	4.5	0.185	540 Beatty St, Vancouver, BC V6B 2L3	49.280578	-123.109191	recommendation
1841	DBnvmIwc6H0BC7JeZlxbbw	Kim Son	4.0	0.192	88 W Pender Street, Unit 2019, Vancouver, BC V…	49.280290	-123.106736	recommendation

```
                            coordinates  \
19303  (49.2779085, -123.10894170979472)
10958         (49.2788834, -123.1082342)
12668   (49.2794748603, -123.107133843)
5557   (49.2800427278, -123.1084868116)
14738         (49.2800689, -123.1086346)
17503         (49.2800759, -123.1083327)
6159           (49.280155, -123.1095422)
11677  (49.2801460008, -123.1099253281)
4570          (49.2805061, -123.1091426)
574           (49.2805779, -123.1091912)
1841    (49.2802904756, -123.106735535)

                                      mercator    mercator_x    mercator_y
19303   (-13704424.703233147, 6322148.887233217) -1.370442e+07  6.322149e+06
10958   (-13704345.943603067, 6322315.239159108) -1.370435e+07  6.322315e+06
12668  (-13704223.452422136, 6322416.1645077905) -1.370422e+07  6.322416e+06
5557    (-13704374.064197747, 6322513.065177328) -1.370437e+07  6.322513e+06
14738    (-13704390.51592718, 6322517.531217344) -1.370439e+07  6.322518e+06
17503    (-13704356.90857291, 6322518.725701968) -1.370436e+07  6.322519e+06
6159    (-13704491.549497025, 6322532.223390063) -1.370449e+07  6.322532e+06
11677   (-13704534.199122025, 6322530.687758192) -1.370453e+07  6.322531e+06
4570    (-13704447.066228503, 6322592.135639856) -1.370445e+07  6.322592e+06
574     (-13704452.476355756, 6322604.387755998) -1.370445e+07  6.322604e+06
1841    (-13704179.112978397, 6322555.341106509) -1.370418e+07  6.322555e+06
```

[74]:
```
chosentile = get_provider(Vendors.CARTODBPOSITRON)
```

[75]:
```
source = ColumnDataSource(data=top_ten)
```

[87]:
```
p = figure(title = 'Nearest Restaurants',
           x_axis_type="mercator", y_axis_type="mercator", x_axis_label =␣
 ↪'Longitude',
           y_axis_label = 'Latitude')
```

[88]:
```
color_mapper = CategoricalColorMapper(palette= ['red', 'blue'], factors =␣
 ↪['userlocation', 'recommendation'])
```

[89]:
```
p.add_tile(chosentile)

p.circle(x = 'mercator_x', y = 'mercator_y', source=source, size=30, fill_alpha␣
 ↪= 0.7,
         color={'field':'location_ind', 'transform': color_mapper}, legend =␣
 ↪'label')

output_notebook()
```

```
show(p)
```

```
BokehDeprecationWarning: 'legend' keyword is deprecated, use explicit
'legend_label', 'legend_field', or 'legend_group' keywords instead
```

[ ]:

[ ]: