nearest_neighbor_analysis

June 17, 2024

1 Nearest neighbor analysis

1.1 Libraries and settings

[ ]:

```python
# Install specific version of scipy

# Libraries
import folium
import platform
import pandas as pd
import seaborn as sns
import geopandas as gdp
import matplotlib.pyplot as plt

# Ignore warnings
import warnings
warnings.filterwarnings('ignore')

# Import functions to calculate nearest-neighbors
import nn_functions as nn
```

1.2 Prepare geodataframe of apartments data

[ ]:

```python
# Read apartments data
df_app = pd.read_csv('apartments_data_enriched.csv',
                     sep=',',
                     encoding='utf-8')[['web-scraper-order',
                                        'lat',
                                        'lon',
                                        'address_raw',
                                        'bfs_number',
                                        'bfs_name']]

# Convert data frame of apartments data to geodataframe
df_app_geo = gdp.GeoDataFrame(df_app,
                         geometry=gdp.points_from_xy(df_app['lon'],
                                                     df_app['lat']))
```

```
# Set Coordinate Reference System (CRS)
df_app_geo.set_crs(4326, allow_override=True)
df_app_geo.head()
```

```
[ ]:   web-scraper-order        lat       lon  \
    0    1662023695-433  47.255714  8.804976
    1    1662023720-634  47.254879  8.793746
    2    1662023745-834  47.277386  8.800306
    3    1662023701-503  47.277386  8.800306
    4    1662023745-820  47.361378  8.533339

                                 address_raw  bfs_number bfs_name  \
    0  Sunnenbergstrasse 15, 8633 Wolfhausen, ZH         112  Bubikon
    1   Blumenbergstrasse 7, 8633 Wolfhausen, ZH         112  Bubikon
    2                          8608 Bubikon, ZH         112  Bubikon
    3                          8608 Bubikon, ZH         112  Bubikon
    4            Lavaterstr. 63, 8002 Zürich, ZH         261   Zürich

                      geometry
    0  POINT (8.80498 47.25571)
    1  POINT (8.79375 47.25488)
    2  POINT (8.80031 47.27739)
    3  POINT (8.80031 47.27739)
    4  POINT (8.53334 47.36138)
```

1.3 Prepare geodataframe of supermarkets data

```
[ ]: # Read supermarket data and select those with know brand
     df_sup = pd.read_csv('supermarkets_data_enriched.csv',
                          sep=',',
                          encoding='utf-8')[['id',
                                             'lat',
                                             'lon',
                                             'brand',
                                             'bfs_number',
                                             'bfs_name']].dropna()
     print(df_sup.shape)

     # Convert data frame of apartments data to geodataframe
     df_sup_geo = gdp.GeoDataFrame(df_sup,
                            geometry=gdp.points_from_xy(df_sup['lon'],
                                                        df_sup['lat']))

     # Set Coordinate Reference System (CRS)
     df_sup_geo.set_crs(4326, allow_override=True)
     df_sup_geo.head()
```

```
# Subset (example)
#df_sup_geo = df_sup_geo[df_sup_geo['brand'] == 'Migros']

# Alternatively, subset of two brands (example)
df_sup_geo = df_sup_geo[df_sup_geo['brand'].isin(['Migros', 'Coop'])]

df_sup_geo.head()
```

```
(2009, 6)
```

```
[ ]:           id        lat       lon   brand  bfs_number bfs_name  \
     4   36726161  47.226191  8.980329  Migros        3339   Uznach
     5   39768209  47.225069  8.969981    Coop        3339   Uznach
     7   39947904  47.376732  8.542161    Coop         261   Zürich
     8   48932835  47.375020  8.522895  Migros         261   Zürich
     10  79977755  47.340070  8.530546    Coop         261   Zürich

                        geometry
     4   POINT (8.98033 47.22619)
     5   POINT (8.96998 47.22507)
     7   POINT (8.54216 47.37673)
     8   POINT (8.52290 47.37502)
     10  POINT (8.53055 47.34007)
```

1.4 Identify closest supermarkets per apartment and calculate its distance

```
[ ]: # Closest supermarket of each apartment
     closest_supermarkets = nn.nearest_neighbor(df_app_geo,
                                                df_sup_geo,
                                                return_dist=True)

     print(len(closest_supermarkets), '==', len(df_app_geo))

     # Rename the geometry of closest stops gdf so that we can easily identify it
     closest_supermarkets = closest_supermarkets.rename(columns={'geometry':␣
      ↪'closest_sup_geom'})
     closest_supermarkets.head()
```

```
870 == 870
```

```
[ ]:           id        lat       lon   brand  bfs_number         bfs_name  \
     0  1362066985  47.229393  8.821159  Migros        3340  Rapperswil-Jona
     1   956494681  47.253231  8.773446    Coop         153     Hombrechtikon
     2   956494681  47.253231  8.773446    Coop         153     Hombrechtikon
     3   956494681  47.253231  8.773446    Coop         153     Hombrechtikon
     4   262400822  47.364072  8.530945  Migros         261            Zürich
```

```
          closest_sup_geom     distance
0  POINT (8.82116 47.22939)  3406.381465
1  POINT (8.77345 47.25323)  2264.541956
2  POINT (8.77345 47.25323)  3995.788117
3  POINT (8.77345 47.25323)  3995.788117
4  POINT (8.53094 47.36407)   398.327277
```

1.5 Merge closest supermarkets to apartments

[ ]:

```
# Merge supermarkets to apartments
result = pd.merge(closest_supermarkets,
                  df_app_geo,
                  left_index=True,
                  right_index=True)[['web-scraper-order',
                                     'address_raw',
                                     'lat_y',
                                     'lon_y',
                                     'id',
                                     'brand',
                                     'geometry',
                                     'closest_sup_geom',
                                     'distance']]

# Rename columns
results = result.rename(columns={'lat_y': 'lat',
                                 'lon_y': 'lon'},
                        inplace = True)
result.head()
```

[ ]:

```
  web-scraper-order                                address_raw        lat  \
0   1662023695-433  Sunnenbergstrasse 15, 8633 Wolfhausen, ZH  47.255714
1   1662023720-634   Blumenbergstrasse 7, 8633 Wolfhausen, ZH  47.254879
2   1662023745-834                          8608 Bubikon, ZH  47.277386
3   1662023701-503                          8608 Bubikon, ZH  47.277386
4   1662023745-820              Lavaterstr. 63, 8002 Zürich, ZH  47.361378

        lon         id   brand                  geometry  \
0  8.804976  338156406    Volg  POINT (8.80498 47.25571)
1  8.793746  956494681    Coop  POINT (8.79375 47.25488)
2  8.800306  338156406    Volg  POINT (8.80031 47.27739)
3  8.800306  338156406    Volg  POINT (8.80031 47.27739)
4  8.533339  262400822  Migros  POINT (8.53334 47.36138)

          closest_sup_geom     distance
0  POINT (8.82088 47.26967)  2340.709105
1  POINT (8.77345 47.25323)  2264.541956
2  POINT (8.82088 47.26967)  2439.597524
```

```
3  POINT (8.82088 47.26967)  2439.597524
4  POINT (8.53094 47.36407)   398.327277
```

1.6 Summary statistics of distance to closest supermarkets

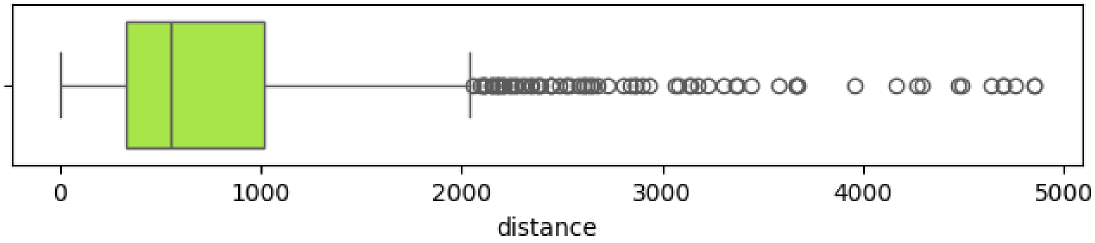
```
[ ]: result['distance'].describe()
```

```
[ ]: count     870.000000
     mean      818.831571
     std       810.177551
     min         2.198898
     25%       324.456254
     50%       550.585590
     75%      1016.089056
     max      4847.253138
     Name: distance, dtype: float64
```

1.7 Boxplot of distance to closest supermarkets

```
[ ]: plt.figure(figsize=(8,1.2))
     plt.ticklabel_format(style='plain')
     sns.boxplot(x=result['distance'],
                 color="greenyellow")
```

```
[ ]: <Axes: xlabel='distance'>
```



1.8 Plotting map with apartments and nearest supermarkets

```
[ ]: # Polygonmap als .json-File (WGS84)
     polys = gdp.read_file("GEN_A4_GEMEINDEN_2019_epsg4326.json")

     # Marker symbols
     url_01 = 'https://raw.githubusercontent.com/pointhi/leaflet-color-markers/
       ↪master/img/marker-icon-blue.png'
     url_02 = 'https://raw.githubusercontent.com/pointhi/leaflet-color-markers/
       ↪master/img/marker-icon-gold.png'
```

```
# Initialisierung der Map
m = folium.Map(location=[47.44, 8.65],
               # tiles='Stamen Toner',
               zoom_start=11)

# Plot Polygonmap of municipalities
folium.Choropleth(
    geo_data=polys,
    name='polys',
    fill_color='transparent',
    line_color='darkred').add_to(m)

# Add lat/lon of apartments
for i in range(0, len(result)):
    folium.Marker(location=(result.iloc[i]['lat'],
                            result.iloc[i]['lon']),
                  popup=result.iloc[i]['address_raw'],
                  icon=folium.features.CustomIcon(url_01,icon_size=(14, 23))).
 ↪add_to(m)

# Add lat/lon of apartments
for i in range(0, len(closest_supermarkets)):
    folium.Marker(location=(closest_supermarkets.iloc[i]['lat'],
                            closest_supermarkets.iloc[i]['lon']),
                  popup=closest_supermarkets.iloc[i]['brand'],
                  icon=folium.features.CustomIcon(url_02,icon_size=(14, 23))).
 ↪add_to(m)

# Layer control
folium.LayerControl().add_to(m)

# Plot map
m
```

```
[ ]: <folium.folium.Map at 0x1fdf0b36f00>
```

1.9 Save data to file

```
[ ]: result.to_csv('apartments_data_with_supermarkets.csv',
                   sep=",",
                   encoding='utf-8',
                   index=False)
```