geocoding_addresses

June 17, 2024

1 Geocoding addresses using the geoadmin API and Python

1.1 Libraries and settings

[ ]:
```
# Libraries
import os
import requests
import json
import urllib
import fnmatch
import folium
import platform
import pandas as pd
import geopandas as gpd
from IPython.display import clear_output

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

1.2 Geocoding a single address

1.2.1 Define base url for address search

[ ]:
```
# Define base url for address search
base_url= "https://api3.geo.admin.ch/rest/services/api/SearchServer?"

# Set up search parameters: address, origins and type
parameters = {"searchText": "8400 Winterthur, Eichgutstrasse 12",
              "origins": "address",
              "type": "locations",
              }

# Urllib.parse.urlencode turns parameters into url
# print(f"{base_url}{urllib.parse.urlencode(parameters)}")
```

1.2.2 Server request & response

[ ]:
```
# Server request
r = requests.get(f"{base_url}{urllib.parse.urlencode(parameters)}") # Get data␣
 ↪in json-format

# Get data in json-format
data = json.loads(r.content)
data

# Take only the first server response, convert to data frame with relevant infos
df = pd.DataFrame.from_dict(list(data.values())[0][0], orient='columns')
df.iloc[[1,4,5,6,11,12],:1]
```


```
b'{"results":[{"attrs":{"detail":"gruenaustrasse 10 8953 dietikon 243 dietikon
ch zh","featureId":"210185276_0","geom_quadindex":"030002112332130023331","geom_
st_box2d":"BOX(672839.3053930395 251411.8132892886,672839.3053930395
251411.8132892886)","label":"Gr\\u00fcnaustrasse 10 <b>8953 Dietikon</b>","lat":
47.40949249267578,"lon":8.403727531433105,"num":10,"objectclass":"","origin":"ad
dress","rank":7,"x":251411.8125,"y":672839.3125,"zoomlevel":10},"id":1579152,"we
ight":4}]}\n'
```


[ ]:

	attrs
featureId	210185276_0
label	Grünaustrasse 10 8953 Dietikon
lat	47.409492
lon	8.403728
x	251411.8125
y	672839.3125

1.3 Geocoding multiple addresses

1.3.1 Importing apartment data

[ ]:
```
# Get current working directory
print(os.getcwd())

# Show all files in the directory
flist = fnmatch.filter(os.listdir('.'), '*.csv')
for i in flist:
    print(i)

# Read the data to a pandas data frame
df = pd.read_csv('apartments_data_prepared.csv',
                 sep=',',
                 encoding='utf-8')[['web-scraper-order',
                                    'address_raw',
                                    'datetime',
```

```
                                    'rooms',
                                    'area',
                                    'luxurious',
                                    'price_per_m2']][:100] # first 100 records

# Get number of rows and columns
print(df.shape)

# Show first records
df.head(5)
```

```
c:\Users\dimit\Documents\applied_data_science\week_04\spatial_data_analysis\02_P
ython_Geocoding_Addresses
apartments_data_geocoded.csv
apartments_data_prepared.csv
(100, 7)
```

[ ]:

	web-scraper-order	address_raw	\
0	1662023695-433	Sunnenbergstrasse 15, 8633 Wolfhausen, ZH	
1	1662023745-820	Lavaterstr. 63, 8002 Zürich, ZH	
2	1662023742-807	Langfurrenstrasse 5c, 8623 Wetzikon ZH, ZH	
3	1662023804-1290	Sandbuckweg 5A, 8157 Dielsdorf, ZH	
4	1662023739-771	Parkring 59, 8002 Zürich, ZH	

	datetime	rooms	area	luxurious	price_per_m2
0	2022-09-07 09:00:00	3.5	122	1	26.07
1	2022-09-07 09:00:00	2.5	78	0	48.21
2	2022-09-07 09:00:00	5.5	115	0	24.87
3	2022-09-07 09:00:00	3.5	74	0	29.26
4	2022-09-07 09:00:00	5.5	195	1	35.38

1.3.2 Geocoding addresses using the geoadmin API

[ ]:
```
# Define base url
base_url= "https://api3.geo.admin.ch/rest/services/api/SearchServer?"

# Geocode list of adresses
geolocation = []
n = 1
for i in df['address_raw'].astype(str):

    print('Geocoding address',
        n,
        'out of',
        len(df['address_raw']),
        ':',
        i)
    n=n+1
```

```
    clear_output(wait=True)

    try:
        # Set up search parameters - address, origins and type
        parameters = {"searchText": i,
                      "origins": "address",
                      "type": "locations",
                      }

        # Server request
        r = requests.get(f"{base_url}{urllib.parse.urlencode(parameters)}")

        # Get data
        data = json.loads(r.content)

        # Take first server response, convert to df with relevant infos
        df_loc = pd.DataFrame.from_dict(list(data.values())[0][0],
                                        orient='columns')
        geolocation.append(df_loc.iloc[[5,6],0].astype(float))

    except Exception:
        geolocation.append(pd.Series(data={'lat': None, 'lon': None}))

# Write lat and lon to df
df_loc = pd.DataFrame(geolocation,
                      columns=("lat", "lon"),
                      index=range(len(df['address_raw'])))
df['lat'] = df_loc['lat']
df['lon'] = df_loc['lon']
df.head(5)
```

[ ]:

	web-scraper-order	address_raw						
0	1662023695-433	Sunnenbergstrasse 15, 8633 Wolfhausen, ZH						
1	1662023745-820	Lavaterstr. 63, 8002 Zürich, ZH						
2	1662023742-807	Langfurrenstrasse 5c, 8623 Wetzikon ZH, ZH						
3	1662023804-1290	Sandbuckweg 5A, 8157 Dielsdorf, ZH						
4	1662023739-771	Parkring 59, 8002 Zürich, ZH						

	datetime	rooms	area	luxurious	price_per_m2	lat	lon
0	2022-09-07 09:00:00	3.5	122	1	26.07	47.255714	8.804976
1	2022-09-07 09:00:00	2.5	78	0	48.21	47.361378	
2	2022-09-07 09:00:00	5.5	115	0	24.87	47.328693	
3	2022-09-07 09:00:00	3.5	74	0	29.26	47.477493	
4	2022-09-07 09:00:00	5.5	195	1	35.38	47.366898	

```
1  8.533339
2  8.810411
3  8.456285
4  8.528817
```

1.3.3 Read polygon-map with municipalities of the canton of Zuerich

[ ]:
```
# Polygonmap als .json-File (WGS84)
polys = gpd.read_file("GEN_A4_GEMEINDEN_2019_epsg4326.json")
print(type(polys))
polys.head(5)
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
```

[ ]:
```
   BFS            NAME BEZIRKSNAM  ART_TEXT  ART_CODE  \
0  117          Hinwil     Hinwil  Gemeinde         1
1  131        Adliswil     Horgen  Gemeinde         1
2    3      Bonstetten  Affoltern  Gemeinde         1
3  154   Küsnacht (ZH)     Meilen  Gemeinde         1
4  135  Kilchberg (ZH)     Horgen  Gemeinde         1

                                            geometry
0  POLYGON ((8.84778 47.32410, 8.85861 47.32162, …
1  POLYGON ((8.53489 47.32502, 8.53662 47.32100, …
2  POLYGON ((8.46026 47.33326, 8.46753 47.33410, …
3  POLYGON ((8.60977 47.33352, 8.61127 47.32749, …
4  POLYGON ((8.54625 47.33441, 8.54875 47.33113, …
```

1.3.4 Plot map

[ ]:
```
# Initialisierung der Map
m = folium.Map(location=[47.44, 8.65], zoom_start=10)

# Map settings
folium.Choropleth(
    geo_data=polys,
    name='polys',
    fill_color='greenyellow'
).add_to(m)

# Add lat/lon of addresses
df_sub = df.dropna()
for i in range(0, len(df_sub)):
    folium.Marker(location=(df_sub.iloc[i]['lat'],
                            df_sub.iloc[i]['lon']),
                  popup=df_sub.iloc[i]['address_raw']).add_to(m)

# Layer control
```

```
folium.LayerControl().add_to(m)

# Plot map
m
```

[ ]: <folium.folium.Map at 0x131c5416ae0>

1.3.5 Intersect municipality polygon-map with lat and lon (point-in-polygon intersection)

[ ]:
```
# lat/lon to GeoDataFrame
pnts = gpd.GeoDataFrame(df,
                        geometry = gpd.points_from_xy(df['lon'],
                                                      df['lat']))

pnts

# Merge spatial data
data_merged = gpd.sjoin(pnts, polys, how="inner", op='within')

# Get relevant columns
df2 = data_merged[['web-scraper-order',
                   'address_raw',
                   'lat',
                   'lon',
                   'BFS',
                   'NAME']]
df2 = df2.rename(columns = {'BFS': 'bfs_number',
                            'NAME': 'bfs_name'})
df2.head(5)
```

[ ]:

	web-scraper-order	address_raw	lat	lon	bfs_number	bfs_name
0	1662023695-433	Sunnenbergstrasse 15, 8633 Wolfhausen, ZH	47.255714	8.804976	112	Bubikon
1	1662023745-820	Lavaterstr. 63, 8002 Zürich, ZH	47.361378	8.533339	261	Zürich
2	1662023742-807	Langfurrenstrasse 5c, 8623 Wetzikon ZH, ZH	47.328693	8.810411	121	Wetzikon (ZH)
3	1662023804-1290	Sandbuckweg 5A, 8157 Dielsdorf, ZH	47.477493	8.456285	86	Dielsdorf
4	1662023739-771	Parkring 59, 8002 Zürich, ZH	47.366898	8.528817	261	Zürich

1.3.6 Save data to file

[ ]:
```
df2.to_csv('apartments_data_geocoded.csv',
           sep=",",
           encoding='utf-8',
           index=False)
```