Member-only story

Geocode with Python

How to Convert physical addresses to Geographic locations → Latitude and Longitude

Abdishakur · Follow

Published in Towards Data Science · 5 min read · Sep 15, 2019

Photo by Thor Alvis on Unsplash

Datasets are rarely complete and often require pre-processing. Imagine some datasets have only an address column without latitude and longitude columns to represent your data geographically. In that case, you need to

convert your data into a geographic format. The process of converting addresses to geographic information — Latitude and Longitude — to map their locations is called **Geocoding**.

Geocoding is the computational process of transforming a physical address description to a location on the Earth's surface (spatial representation in numerical coordinates) — Wikipedia

In this tutorial, I will show you how to perform geocoding in Python with the help of Geopy and Geopandas Libraries. Let us install these libraries with Pip if you have already Anaconda environment setup.

```
pip install geopandas
pip install geopy
```

If you do not want to install libraries and directly interact with the accompanied Jupyter notebook of this tutorial, there are Github link with MyBinder at the bottom of this article. This is a containerised environment that will allow you to experiment with this tutorial directly on the web without any installations. The dataset is also included in this environment so there is no need to download the dataset for this tutorial.

Geocoding Single Address

To geolocate a single address, you can use Geopy python library. Geopy has different Geocoding services that you can choose from, including Google Maps, ArcGIS, AzureMaps, Bing, etc. Some of them require API keys, while others do not need.

Geopy

As our first example, we use Nominatim Geocoding service, which is built on top of OpenStreetMap data. Let us Geocode a single address, the Eifel tower in Paris.

```
locator = Nominatim(user_agent="myGeocoder")
location = locator.geocode("Champ de Mars, Paris, France")
```

We create `locator` that holds the Geocoding service, Nominatim. Then we pass the locator we created to geocode any address, in this example, the Eifel tower address.


```
print("Latitude = {}, Longitude = {}".format(location.latitude,
location.longitude))
```

Now, we can print out the coordinates of the location we have created.

```
Latitude = 48.85614465, Longitude = 2.29782039332223
```

Try some different addresses of your own. In the next section, we will cover how to geocode many addresses from Pandas Dataframe.

Geocoding addresses from Pandas

Let us read the dataset for this tutorial. We use an example of Store addresses dataset for this tutorial. The CSV file is available in this link.

Download the CSV file and read it in Pandas.

```
df = pd.read_csv("addresses.csv")
df.head()
```

The following table provides the first five rows of the DataFrame table. As

you can see, there are no latitude and longitude columns to map the data.

	Typ	Nr	Namn	Address1	Address3	Address4	Address5	Telefon
0	Butik	102	Fältöversten	Karlaplan 13	115 20	STOCKHOLM	Stockholms län	08/662 22 89
1	Butik	104	NaN	Nybrogatan 47	114 39	STOCKHOLM	Stockholms län	08/662 50 16
2	Butik	106	Garnisonen	Karlavägen 100 A	115 26	STOCKHOLM	Stockholms län	08/662 64 85
3	Butik	110	NaN	Hötorgshallen	111 57	STOCKHOLM	Stockholms län	08/56849241
4	Butik	113	Sergel	Drottninggatan 45	111 21	STOCKHOLM	Stockholms län	08/21 47 44

Ddataframe

We concatenate address columns into one that is appropriate for geocoding. For example, the first address is:

```
Karlaplan 13,115 20,STOCKHOLM,Stockholms län, Sweden
```

We can join address columns in pandas like this to create an address column for the geocoding:

Once we create the address column, we can start geocoding as below code snippet.

```python
from geopy.extra.rate_limiter import RateLimiter

# 1 - conveneint function to delay between geocoding calls
geocode = RateLimiter(locator.geocode, min_delay_seconds=1)
# 2- - create location column
df['location'] = df['ADDRESS'].apply(geocode)
# 3 - create longitude, laatitude and altitude from location column (returns tuple)
df['point'] = df['location'].apply(lambda loc: tuple(loc.point) if loc else None)
# 4 - split point column into latitude, longitude and altitude columns
df[['latitude', 'longitude', 'altitude']] = pd.DataFrame(df['point'].tolist(), index=
```

geocode3.py hosted with ❤ by GitHub view raw

- #1 — We first delay our Geocoding 1 second between each address. This is convenient when you are Geocoding a large number of physical addresses as the Geocoding service provider can deny access to the service.
- #2 — Create a `df['location']` column by applying `geocode` we created.
- #3 — Third, we can create latitude, longitude, and altitude as a single tuple column.

- #4 — Finally, We split latitude, longitude, and altitude columns into three separate columns.

The above code produces a Dataframe with latitude and longitude columns that you can map with any Geographic visualisation tool of your choice. Let us look at the first few raws of our DataFrame, but first, we will clean out the unwanted columns.

```
df = df.drop(['Address1', 'Address3', 'Address4',
'Address5','Telefon', 'ADDRESS', 'location', 'point'], axis=1)

df.head()
```

	Typ	Nr	Namn	latitude	longitude	altitude
0	Butik	102	Fältöversten	59.338315	18.089960	0.0
1	Butik	104	NaN	59.337207	18.079098	0.0
2	Butik	106	Garnisonen	59.335380	18.100626	0.0
3	Butik	110	NaN	59.334327	18.062604	0.0
4	Butik	113	Sergel	59.332481	18.062809	0.0

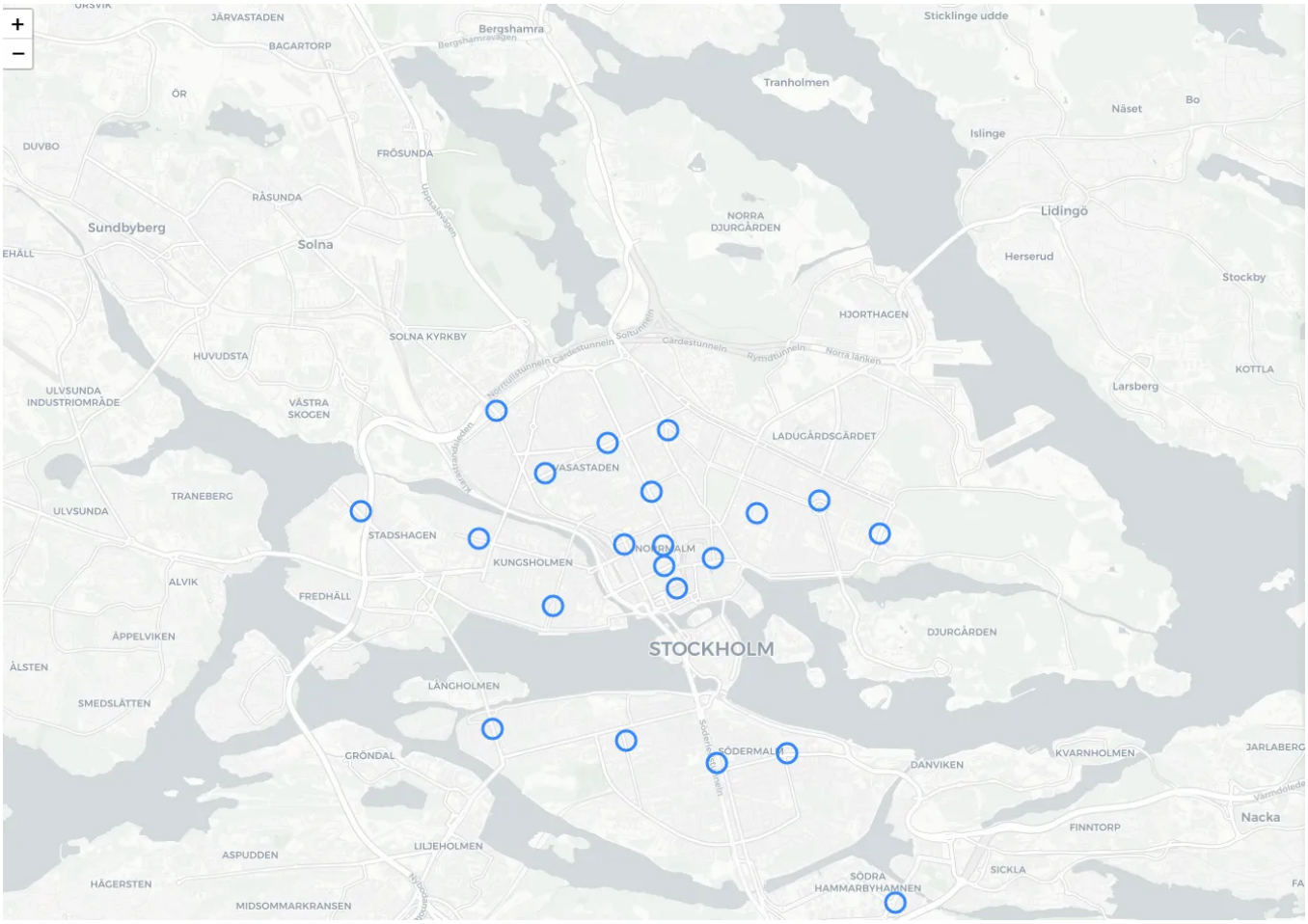
cleaned table with latitude and longitude

I will use Folium to map out the points we created but feel free to use any other Geovisualization tool of your choice. First, we display the locations as a circle map with Folium.

```
map1 = folium.Map(
    location=[59.338315,18.089960],
    tiles='cartodbpositron',
    zoom_start=12,
)
df.apply(lambda row:folium.CircleMarker(location=[row["latitude"], row["longitude"]]).a
map1
```

folium1.py hosted with ♥ by GitHub view raw

The map produced below shows the geocoded addresses as circles.



Map

Or if you prefer a dark background with an aggregated cluster of points, you can do the following:

```
folium_map = folium.Map(location=[59.338315,18.089960],
                        zoom_start=12,
                        tiles='CartoDB dark_matter')


FastMarkerCluster(data=list(zip(df['latitude'].values, df['longitude'].values))).add_t
folium.LayerControl().add_to(folium_map)
folium_map
```

folium2.py hosted with ❤ by GitHub view raw

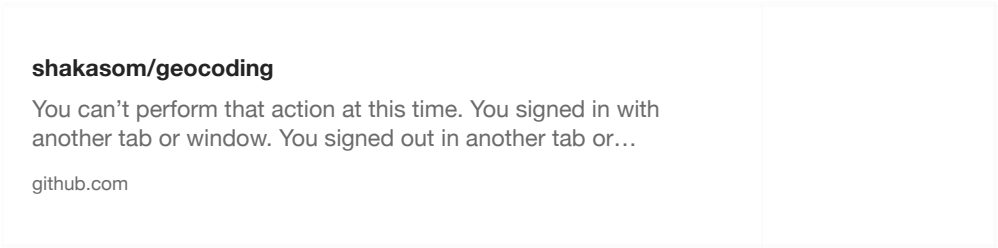
Below is a dark background map with Clustered points map in Folium.

Clustered map

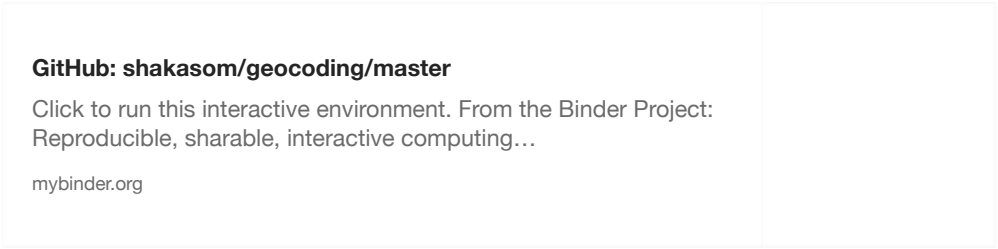
Conclusion

Geocoding is a critical task in many location tasks that require coordinate systems. In this article, we have seen how to do geocoding in Python. There are a lot of other services that provide either free or paid geocoding services that you can experiment within GeoPy. I find Google Maps geocoding services more powerfull than the Openstreetmap services we have used in this tutorial, but it requires an API key.

To interact and experiment with this tutorial without any installation, I created a Binder. Go this GitHub repository and click on launch binder.



Or directly to the Jupyter notebook Binder link here:



Pandas Python Geospatial Geographicdatascience Geocoding

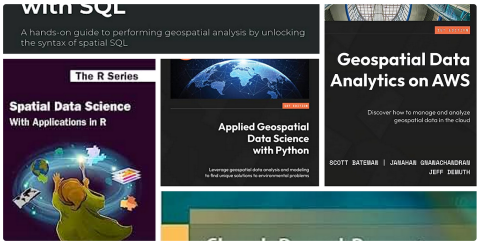
Written by Abdishakur

Follow

5.6K Followers · Writer for Towards Data Science

Writing about Geospatial Data Science, AI, ML, DL, Python, SQL, GIS | Top writer | 1m views.

More from Abdishakur and Towards Data Science



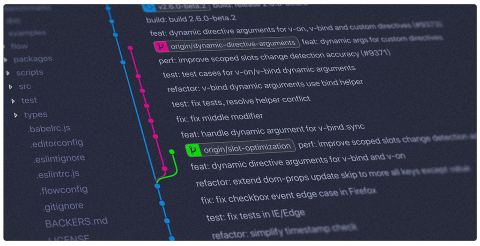
Abdishakur in Spatial Data Science

6 New Must-Read Books in Geospatial Data Science

Unlock the Power of Geospatial Data Science with These New Books

· 5 min read · Jul 10

124 3



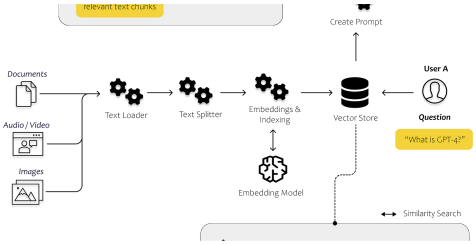
Miriam Santos in Towards Data Science

Pandas 2.0: A Game-Changer for Data Scientists?

The Top 5 Features for Efficient Data Manipulation

7 min read · Jun 27

1.96K 24



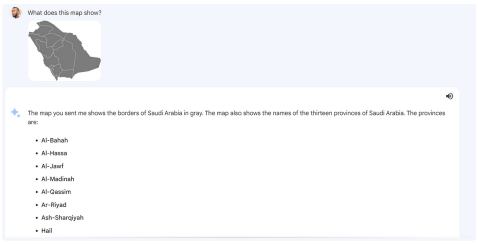
Dominik Polzer in Towards Data Science

All You Need to Know to Build Your First LLM App

A step-by-step tutorial to document loaders, embeddings, vector stores and prompt

· 26 min read · Jun 21

3.1K 28



Abdishakur in Spatial Data Science

Google Bard's New Photo Prompt with Geospatial Data

Is Google Bard's New Photo Prompt an excellent tool for Geospatial Data?

· 4 min read · Jul 17

47

See all from Abdishakur | See all from Towards Data Science

Recommended from Medium

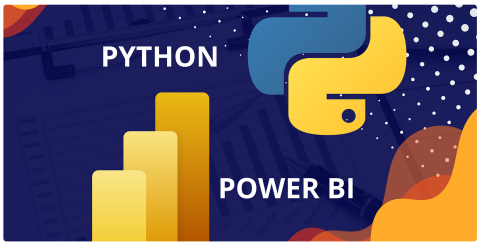
Manoj Das

Geoplotlib: Exploring the World with Python🗺️

A Comprehensive Guide to Visualizing Geographical Data using Geoplotlib in

7 min read · Mar 4

80 2



Deepesh Nishad in CodeX

Unleashing the Power of Power BI and Geopandas: Building

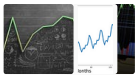
Geospatial data visualisation is a powerful tool for understanding patterns and trends in

11 min read · Jan 29

63

Lists

Coding & Development
11 stories · 70 saves



Predictive Modeling w/ Python
18 stories · 165 saves

New_Reading_List
174 stories · 36 saves

Practical Guides to Machine Learning
10 stories · 184 saves

Dr. Shouke Wei

Geodata Exploration and Visualization with GeoPandas

GeoPandas provides easy methods to explore geodata and visualize geodata with

✦ · 11 min read · Mar 20

42 2

PiotrDev

Reverse Geocode with Python, real example with

Reverse Geocode takes a latitude/longitude coordinate and returns the country and city.

2 min read · Mar 4

57 2

Rich

Hash cracking cheatsheet

TL;DR cheatsheet for cracking various hash types with john & hashcat, with a footnote on

4 min read · Mar 22

Sixing Huang in Geek Culture

Visualize the German Railway System with Kepler.gl

Create maps for stations and routes in Jupyter

✦ · 7 min read · Nov 9, 2022

23

See more recommendations