Working with Geospatial Data

Galina Naydenova, Machine Learning Engineer

https://www.linkedin.com/in/galina-naydenova-msc-fhea-b89856196/

Speaker Introduction: Galina Naydenova

Freelance Machine Learning Engineer

Impact Start-ups, NGOs, Educational Institutions

Bulgaria -> UK -> Japan

Work in Tech and Research

Data Science Manager, OU, UK

Learning Analytics

HEA Fellow, UK

From 2020 Freelance Machine Learning Engineer

Taught Data Science at Le Wagon

Voluntary work

Building AI For Good,
For the People, By the People

Omdena Japan

AI for Social Good

- Lead Machine Learning Engineer
- Product Owner, Mentor
- Projects with the UN, World Resources Institute, ASU, others
- Leader of **Omdena Japan Chapter**

Omdena Japan Projects

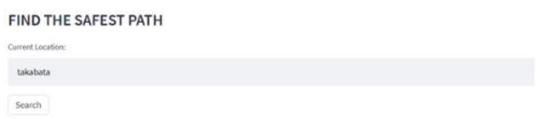
❖ Locally Relevant Challenges ❖ Focus on Learning ❖ Topical Tutorials ❖ Soft skills and teamwork ❖ 4 weeks, 5-8 hrs/w

Finding Paths to Safety Following Natural Disasters

Omdena Japan

COMPLETED

Pathfinding Dashboard


FIND THE SAFEST PATH

Current Location:

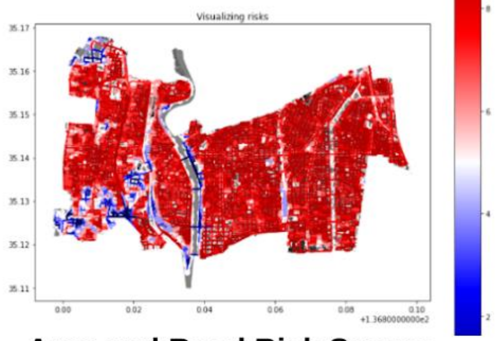
takabata

Search

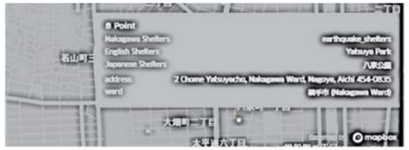
TAKAHATA
ARAKO
NAKA
MIYAWAKICHO
USHIDAT
Ichiyanagi Jr. High School
MATOBACHO
Sasabara Elementary School
SHINGOCHO
aruike Park
HIRATOCHO

Interactive Streamlit App

Data Sourcing and OSM



Area and Road Risk Scores


Point

Nakagawa Shelters	earthquake_shelters
English Shelters	Yatsuya Park
Japanese Shelters	八剱公園
address	2 Chome Yatsuyacho, Nakagawa Ward, Nagoya, Aichi 454-0835
ward	中川区 (Nakagawa Ward)

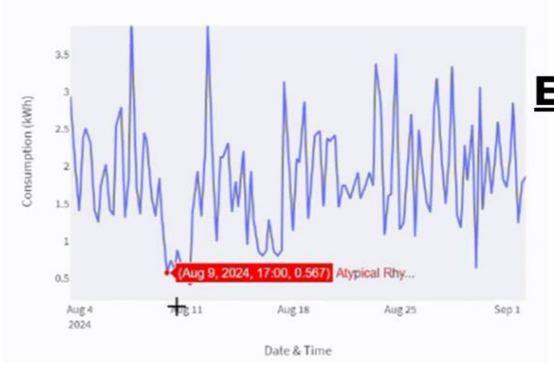
Bilingual Shelter Information

Monitoring the Wellbeing of the Elderly and Providing Support with IoT

Omdena Japan

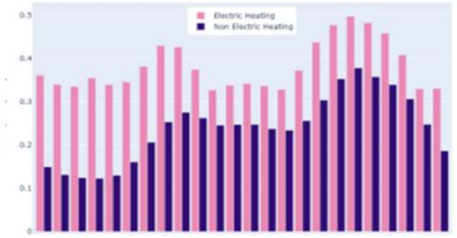
COMPLETED

Anomaly Detection

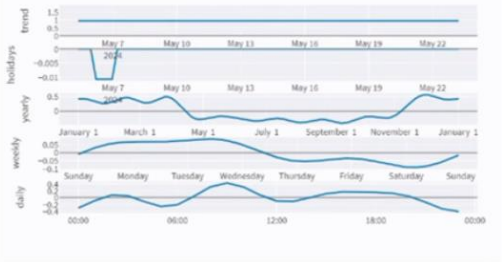


Interactive Dashboard

Daily Patterns



Energy Consumption Data



Time-Series Analysis

https://omdena.com/local-chapters/japan-chapter/

Workshop Agenda

1. Introduction to Geospatial Data
2. Geo Data sources in Japan
3. Reading, manipulating and visualizing geospatial data
4. The Geopandas Python package - common operations
5. Use of geospatial data. Examples of use in AI for Good projects

Introduction to Geospatial Data

Geospatial data are data for which a specific location is associated with each record.

- (similarities) It is data.
- A lot of the operations we will be doing with geospatial data are very similar to those we would do with non-spatial data
- (differences) every observation has a location and can be "put on a map“
- Allows us to look at spatial relationships between the data
- Geospatial data is the combination of the data itself, and the information it carries, and its location
- For example Census data – the real, valuable information is the data itself (e.g. population characteristics), with added value of the location

Types of data. Two ways to ‘see’ the world:

- Raster - encodes the world as a continuous surface represented by a grid, such as the pixels of an image. Examples: altitude data or satellite images
- Vector - collection of discrete objects using points, lines and polygons. For example, discrete features where buildings are represented as polygons and roads as lines

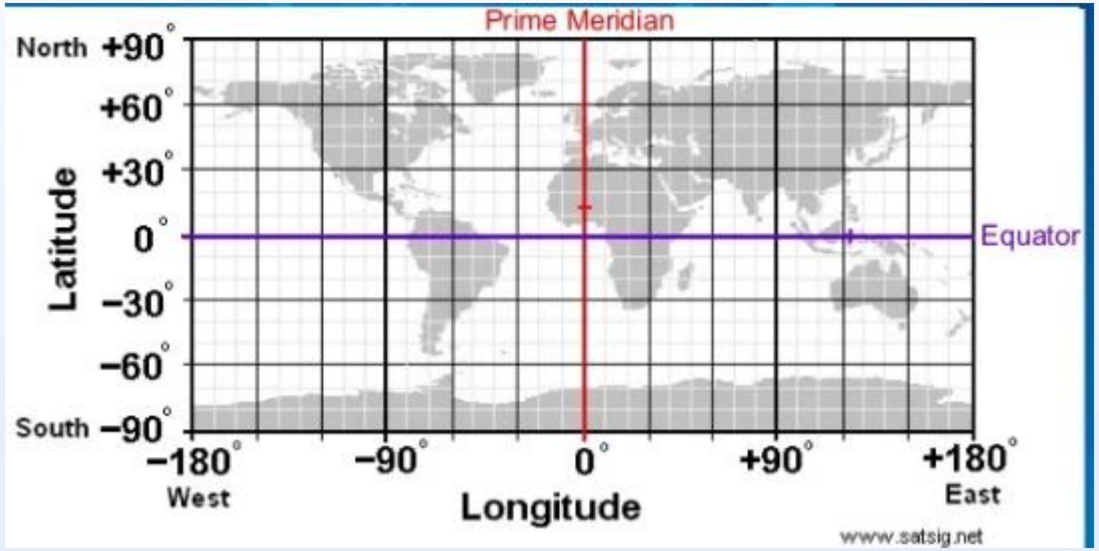
Introduction to Geospatial Data

Types of VECTOR Data:

Points – a point geometry: a single location with X and Y coordinates

Line strings – is a group of connected points – e.g roads;

Polygon – a closed line that encircles an area. e.g. countries. Variations: Multipolygon, 3D Polygons



X coord is Longitude
Y coord is Latitude

Feature attributes:

The information about our vector features (e.g. type, value). Our collection of features, for example all the prefectures of Japan, combined with its attributes, we end up with a table. Similarities with tabular data and the Python pandas package.

Sources of Geospatial data in Japan

National Land Numerical Information Download Site (mlit.go.jp)

- Water and Land Data (features, elevation, topography)
- Administrative and Policy areas
- Disaster and Disaster Prevention Areas (data behind https://disaportal.gsi.go.jp/)
- Regional data – points of interest (schools, fuel stations, etc)
- Transportation data – roads, rail and bus routes, traffic flow by station
- Population projection

❖ Free to use

❖ Generally in geo format (.geojson, .shp)

❖ Normally available on prefecture level

Sources of Geospatial data in Japan

Tokyo Metropolitan Government Open Data Catalog Website

- ❖ Collection of varied information, not necessarily geospatial, not always structured
- ❖ Mainly in .csv format
- ❖ Use cases, annual Hackathon

Other sources:

- Portal Site of Official Statistics of Japan (e-stat.go.jp) - census data, admin boundaries data, time-series
- Ward city planning sites, e.g. Setagaya i-map (zoning, local info)
- Specialized data sites, e.g. Seismic data from J-SHIS (bosai.go.jp)
- Many others (some commercial). Google Maps

When working with Japanese geo data

- ❖ Careful with Google Translate (especially with units, years)
- ❖ Problems with character encoding
- ❖ Availability dependent on locally supplied data
- ❖ Beware of data volumes

Using Geospatial data. Geopandas

GeoPandas is a library for working with tabular, geospatial vector data, extending the pandas DataFrame

It can work with most specific file formats for geospatial data, such as GeoJSON files, GeoPackage files (gpkg) , or shape (.shp) files, which are specialized in storing spatial data.

The equivalent of pandas Dataframe is Geodataframe. It has always a "**geometry**" column, that holds the location information. The other columns are the attributes that describe each of the geometries.

Hands on task:
Create a record/datapoint from collected coordinates, and create a Point geometry in a geodataframe

Question:
Where is Le Wagon Tokyo office?

Using Geospatial data. Geopandas

Non-geo-specific formats (for example csv) can be read in a similar way. Example – Tokyo LG Open data – the coordinates are in columns in the csv file. By setting them as geometry coordinates they are no longer just numeric columns, but acquire another meaning

.Read_file – it can read csv (like in above), and also specialized geo format

.to_file – GeoPandas can read geospatial file formats with the read_file function, but it can also write such files. This is done with the `to_file` method. The first argument is the name of the resulting file, or a full path. In addition, you need to specify which file format you want to write using the "driver" keyword.

.geometry - returns the geometry column, regardless of the name. Recognized the geometry automatically

.plot() – plots the geometry without the need to specify

Using Geospatial data. Visualizing

How to display:

(in notebook)

- The .plot() method
- With matplotlib (can display overlapping maps too) – seen later

(With exporting the file)

- http://geojson.io/ - just drag and drop
- kepler.gl – can have different tolerances to incomplete data

Question:
Where is Le Wagon Tokyo office?

Hands on task:
Visualize the geo data point in different ways

Display of the location on a basemap. Can be done in a notebook, with the **contextily** package - https://geopandas.org/en/stable/gallery/plotting_basemap_background.html

Using Geospatial data. Visualizing

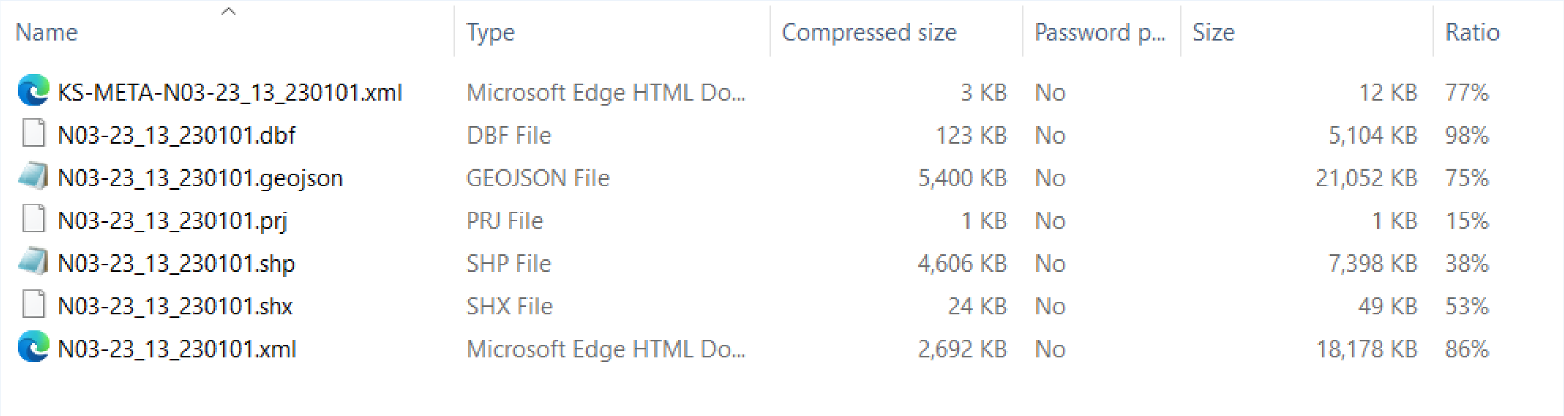
Question:
Where is Le Wagon Tokyo office?

Hands on task:
Make your own basemap data

Download the administrative boundaries for Tokyo

[National Land Numerical Information | Administrative Boundaries Data (mlit.go.jp)]

.zip file content



Name	Type	Compressed size	Password p...	Size	Ratio
KS-META-N03-23_13_230101.xml	Microsoft Edge HTML Do...	3 KB	No	12 KB	77%
★ N03-23_13_230101.dbf	DBF File	123 KB	No	5,104 KB	98%
⇨ N03-23_13_230101.geojson	GEOJSON File	5,400 KB	No	21,052 KB	75%
★ N03-23_13_230101.prj	PRJ File	1 KB	No	1 KB	15%
★ N03-23_13_230101.shp	SHP File	4,606 KB	No	7,398 KB	38%
★ N03-23_13_230101.shx	SHX File	24 KB	No	49 KB	53%
N03-23_13_230101.xml	Microsoft Edge HTML Do...	2,692 KB	No	18,178 KB	86%

The **shape (.shp)** file has multiple components ★ (. shp, .shx, .prj, .dbf) You need all of them for it to work
We will work with the .geojson file ⇨

Using Geospatial data. Filtering

Filtering

- taking a subset of the dataframe by filtering on one of the attributes.

Let's take the dataset with all the Tokyo wards. There is a column indicating the area code.

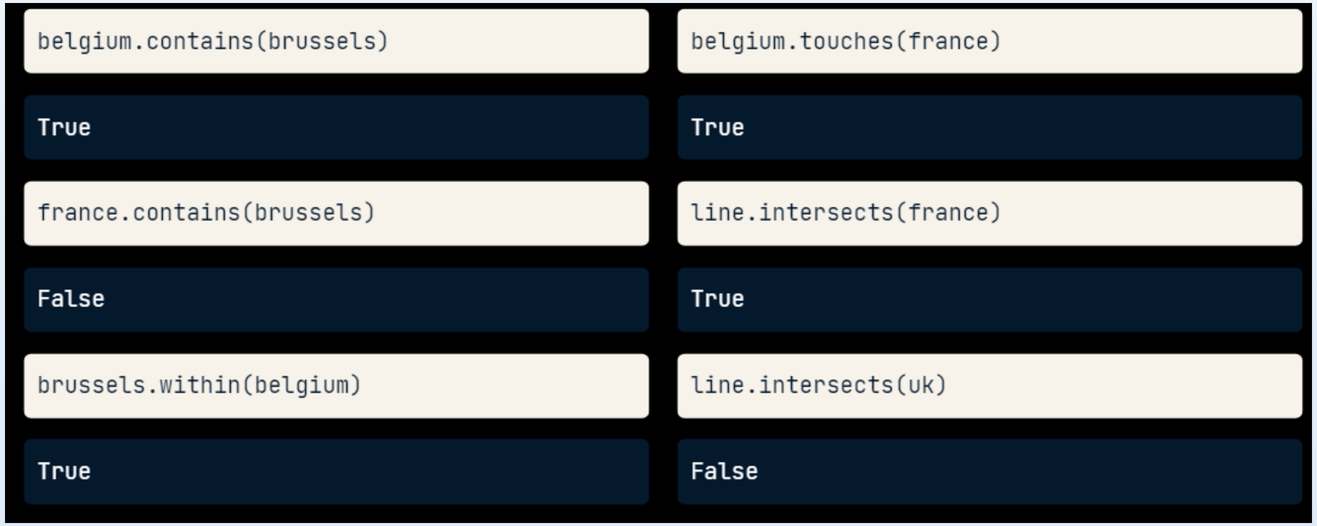
So now we can do a filtering operation to look for all Tokyo 23 special wards.

(basic pandas functionality)

Hands on task: Locate, explore, read in the Tokyo wards. Filter the 23 special wards only. Display on map. Plot the multiple datasets

Spatial relationships. The Shapely package

- extract one of the values of the geometry column, using the loc attribute of a Dataframe
- check type - shapely.geometry.point.Point - a shapely point object
- **Shapely** is a **Python** package to work with geometric objects.
- It provides the Point, LineString and Polygon geometry objects, and is used by GeoPandas under the hood.
- The geometry column of a GeoDataFrame, which is a GeoSeries, thus consists of Shapely objects
- Comes with built-in spatial methods, such as area (e.g. for polygons) and distance
- allows us to spatially relate different geometries – ‘within’, ‘contains’, ‘touches’, ‘intersects’.



```
belgium.contains(brussels)
True
france.contains(brussels)
False
brussels.within(belgium)
True
```

```
belgium.touches(france)
True
line.intersects(france)
True
line.intersects(uk)
False
```

Hands on task: Determine which Tokyo ward ‘contains’ the Le Wagon office

Image: Datacamp

Calculating Distance. CRS (Coordinate System)

- Geographic coordinates: we define a position on the globe in degrees of latitude and longitude
- Going from the globe to a flat map is what we call a *projection*
- We project the surface of the earth onto a 2D plane, this creates distortions
- Some projection systems will try to preserve the area size of geometries
- Other projection systems try to preserve angles, such as the Mercator projection
- Every projection system will always have some distortion of area, angle or distance
- Most geospatial formats contain a string representing the CRS. If the file contains CRS, this is read automatically.
- The most popular is EPSG:4326, also called WGS84. Example: Google Maps
- In GeoPandas, the CRS information is stored in the crs attribute
- If there is no information, but you know which CRS the data are expressed in, you can add it manually

Hands on task: Set CRS for both the Bicycle Parking and the Le Wagon Tokyo datasets

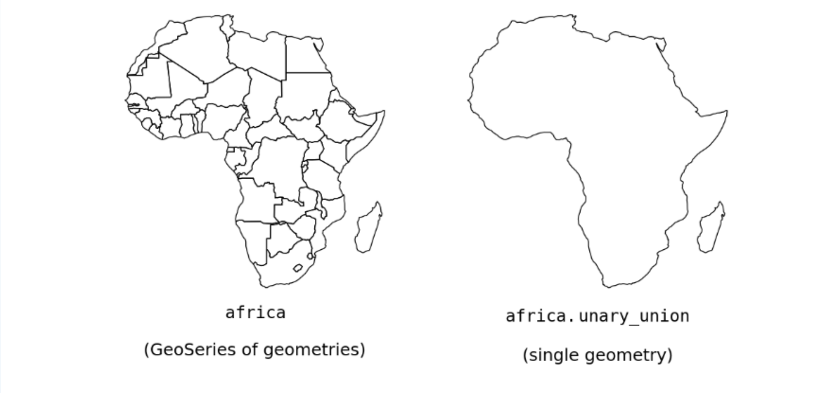
Hands on task: Apply distance calculation and visualize the closest point

Hands on task: Apply distance calculation and identify and visualize the closest bus line. Any problems?

Creating new geometries

Hands on task:
Locate, explore, read in the bus data. Check which lines are in Meguro. Demonstrate spatial relationship (e.g. within), get the bus lines in Meguro only. Create new dataset

Unary union – when you want to take the union of a whole series of geometries



`africa`

(GeoSeries of geometries)

`africa.unary_union`

(single geometry)

Hands on task:
Create a unary union for Tokyo, and check whether our point is within Tokyo

Image: Datacamp

Creating new geometries – Spatial Join

- We can use one of the spatial operations provided by GeoPandas to check the spatial relationship
- Bringing information from other sets, e.g. ward name to the bicycle parking dataset
- spatial join - joining on location (rather than on a shared column or index)

sjoin function. Arguments:

- the geodataframe to which we want add information, (in our case - bicycle parking locations)
- the geodataframe that contains the information we want to add (in our case - ward name)
- which spatial relationship we want to use to match both datasets. (in our case - "within“)

```
joined=gpd.sjoin(bpark_df,admin,op='within')
```

Hands on task:
Add ward name to the bicycle parking locations with sjoin. Simplify dataset

Bonus hands on task:
Using unary union, create a dataset of bus lines within Tokyo. Save into geojson file and visualize.

Your turn: Uses of Geospatial Data

- Road characteristics
- Exploring local areas
- Disaster planning
- Assistance for people with disabilities
- Ecological risk
- Bicycle routes
- Finding mid-point between locations
- And many, many more

THANK YOU!

Galina Naydenova

https://www.linkedin.com/in/galina-naydenova-msc-fhea-b89856196/