Programowanie Aplikacji Geoinformacyjnych

1060-GI000-ISP-5002

Lab 8. Wprowadzenie do analizy danych

Analiza danych w Pythonie

Analiza danych w Pythonie

- Heterogeniczne (niejednorodne) struktury danych: listy, tuple, zbiory i słowniki.
- Trudno przetwarzać dane tabelaryczne.
- Moduł numpy wprowadza pojęcie array

Pandas

- Pandas to biblioteka języka Python używana do pracy ze zbiorami danych.
- Posiada funkcje do analizy, czyszczenia, eksploracji i manipulowania danymi.
- Nazwa „Pandas” odnosi się zarówno do „Panel Data”, jak i „Python Data Analysis” i została stworzona przez Wesa McKinneya w 2008 roku.
- Pandas pozwala analizować zbiory big data i wyciągać wnioski na podstawie teorii statystycznych.
- Pandas potrafi wyczyścić nieuporządkowane zbiory danych i sprawić, że będą czytelne i spójne.
- Odpowiednio przygotowane dane są podstawą nauki o danych.

Pandas

- Pandas to jedna z najbardziej wszechstronnych i popularnych bibliotek służących analizie danych w języku Python.
- Głównym celem twórców pandas było dostarczenie prostego w użyciu narzędzia do manipulacji danymi tabelarycznymi w Python.
- Pandas stanowi podstawę wielu innych bibliotek analitycznych, a jego znajmość znacząco ułatwi zrozumienie bardziej zaawansowanych zagadnień i narzędzi (w tym narzędzi big data).

Literatura

- Biblioteka Pandas - https://pandas.pydata.org/
- Podręcznik biblioteki Pandas - https://pandas.pydata.org/docs/pandas.pdf
- Pandas Tutorial - https://www.w3schools.com/python/pandas/default.asp

Pandas

Narzędzia wykorzystywane w bibliotece Pandas:

- pandas - Analiza danych w języku Python
- NumPy - Obsługa danych liczbowych
- matplotlib - Wizualizacja wyników w formie wykresów
- SciPy – obliczenia naukowe w Pythonie

Pandas – typy danych

- Fundamentalnym elementem biblioteki pandas są struktury danych.
- Pandas wprowadza dwie struktury danych:
- DataSeries / Series (https://www.w3schools.com/python/pandas/pandas_series.asp)
- DataFrame (https://www.w3schools.com/python/pandas/pandas_dataframes.asp)

Pandas – DataSeries

- Series (https://www.w3schools.com/python/pandas/pandas_series.asp)
- jednowymiarowa, indeksowana tablica danych dowolnego typu zgodnego z Python

Pandas – DataFrame

- DataFrame (https://www.w3schools.com/python/pandas/pandas_dataframes.asp)
- Tabelaryczna struktura danych, która jest w stanie przechowywać w poszczególnych kolumnach zmienne o różnych typach zgodnych z Python.
- Może być utożsamiana z tabelą z relacyjnej bazy danych lub zbiorem obiektów Series.
- Jest to typ danych najczęściej wykorzystywany w bibliotece pandas

Pandas – przykład

```
import pandas as pd

# Załadowanie pliku do DataFrame
df = pd.read_csv(,plik.csv')

# SELECT * FROM df LIMIT 5
print(df.head())

# SELECT * FROM df WHERE 'age’ > 30
print(df[df['age'] > 30])

# UPDATE df SET 'salary’ = 50000 WHERE 'age’ > 30)
df.loc[df['age'] > 30, 'salary'] = 50000
print(df.head())

# DELETE FROM df WHERE 'age’ < 25)
df = df[df['age'] >= 25]
print(df.head())
```

Pandas – przykład

```
import pandas as pd

data1 = {
    'id': [1, 2, 3, 4],
    'name': ['Alice', 'Bob', 'Charlie', 'David']
}

data2 = {
    'id': [3, 4, 5, 6],
    'age': [23, 34, 45, 56]
}

df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)

# INNER JOIN
print(pd.merge(df1, df2, on='id', how='inner'))

# LEFT JOIN
print(pd.merge(df1, df2, on='id', how='left'))

# RIGHT JOIN
print(pd.merge(df1, df2, on='id', how='right'))

# FULL JOIN (OUTER JOIN)
print(pd.merge(df1, df2, on='id', how='outer'))
```

Analiza danych przestrzennych w Pythonie

GeoPandas

- Biblioteka GeoPandas stanowi rozszerzenie biblioteki pandas o obsługę danych przestrzennych.
- Do głównych możliwości GeoPandas zaliczają się:
  - odczyt i zapis danych przestrzennych,
  - obliczenia na tych danych (odległość, powierzchnia),
  - analizy relacji przestrzennej pomiędzy obiektami,
  - prosta wizualizacja tych danych.

GeoPandas

- GeoPandas to projekt open source, którego celem jest opracowanie biblioteki narzędzi języka Python ułatwiających pracę z danymi geoprzestrzennymi.
- GeoPandas rozszerza typy danych używane przez pandas, aby umożliwić operacje przestrzenne na typach geometrycznych.
- Operacje geometryczne wykonywane są przez bibliotekę shapely.
- Geopandas ponadto używa biblioteki fiona w celu uzyskania dostępu do plików danych oraz biblioteki matplotlib do wyświetlania i drukowania wyników.

GeoPandas

- Dokumentacja GeoPandas - https://geopandas.org/docs.html
- Instalacja: https://geopandas.org/getting_started/install.html
- Wprowadzenie https://geopandas.org/getting_started/introduction.html

- GeoSeries https://geopandas.org/docs/user_guide/data_structures.html#geoseries
- GeoDataFrame https://geopandas.org/docs/user_guide/data_structures.html#geodataframe

- Praca z plikami danych https://geopandas.org/docs/user_guide/io.html#reading-spatial-data
- Praca z bazą danych przestrzennych: https://geopandas.org/docs/user_guide/io.html#spatial-databases

- Przykłady: https://github.com/geopandas/geopandas
- Introduction to Geospatial Data in Python https://www.datacamp.com/community/tutorials/geospatial-data-python
- https://towardsdatascience.com/geopandas-hands-on-geospatial-relations-and-operations-a6e7047d7ba1
- https://towardsdatascience.com/plotting-maps-with-geopandas-428c97295a73

GeoPandas

- GeoPandas wprowadza rozszerzenie podstawowych struktur danych biblioteki pandas.
- GeoDataFrame jest więc rozszerzeniem pandas DataFrame pozwalającym na przechowywanie kolumn geometrycznych i realizację operacji przestrzennych.
- Obiekty przestrzenne mogą być przechowywane pod postacią GeoSeries, które z kolei stanowi rozszerzenie klasy Series z pandas.
- GeoDataFrame posiada więc pełnie możliwości pandas DataFrame oraz umożliwia przechowywanie standardowych danych w postaci kolumn typu Series oraz danych przestrzennych z użyciem GeoSeries.
- Obiekt GeoSeries może przechowywać geometrię dowolnego typu, a także dopuszczalne jest mieszanie typów geometrii w ramach pojedynczej kolumny.
- Każdy GeoSeries posiada atrybut GeoSeries.crs przechowujący informacje o układzie współrzędnych danej kolumny.
- Pojedynczy GeoDataFrame może zawierać wiele kolumn GeoSeries (przykładowo z różnymi układami współrzędnych), a jedna z nich postrzegana jest jako aktywna i to domyślnie na jej obiektach przeprowadzane będą operacje przestrzenne.

Shapely

- Shapely to pakiet języka Python, który umożliwia przeprowadzenie rozbudowanych analiz i manipulacji na obiektach geometrycznych.
- Pozwala on na operacje na obiektach takich jak punkty, linie, płaszczyzny i kolekcje geometrii.
- Jest oparty na silniku przestrzennej bazy danych PostGIS i zbliżony do niej funkcjonalnością.
- Shapely stanowi również podstawę biblioteki GeoPandas i wraz z nią tworzy wszechstronny zestaw narzędzi do operacji na danych przestrzennych w Python.
- Dokumentacja Shapely - https://shapely.readthedocs.io/en/stable/manual.html

Shapely – model danych

- Shapely korzysta z trzech abstrakcyjnych rodzajów obiektów geometrycznych:
  - punkty (Point),
  - krzywe (Curve),
  - powierzchnie (Surface).
- Każdy obiekt, niezależnie od typu, jest powiązany z trzeba zbiorami punktów:
  - wnętrze (interior),
  - granica (boundary),
  - zewnętrze (exterior).
- Wnętrze punktu zawiera dokładnie jeden punkt, granica żadnego punktu, a zewnętrze to zbiór wszystkich innych punktów. Wymiar punktu wynosi 0.
- Wnętrze krzywej zawiera nieskończoną liczbę punktów wzdłuż jej długości, granica to dwa punkty na obu końcach krzywej. Zewnętrze to wszystkie inne punkty, a wymiar krzywej to 1.
- Wnętrze powierzchni składa się z nieskończonej liczby punktów, granica składa się z jednej lub kilku krzywych, a zewnętrze to zbiór wszystkich pozostałych punktów włączając te wewnątrz ewentualnych otworów wewnątrz powierzchni. Wymiar powierzchni to 2.

Shapely – model danych

- Ten abstrakcyjny podział stanowi podstawę do rozumienia relacji przestrzennych w Shapely, a poszczególne obiekty są implementowane przez konkretne klasy języka Python:
- Punkt jest implementowany przez klasę `Point`
- Krzywa jest implementowana w oparciu o klasy `LinearString` oraz `LinearRing`
- Powierzchnia jest reprezentowana w postaci klasy `Polygon`

GeoPandas – przykład

```
import geopandas as gpd
from shapely.geometry import box

# Załadowanie pliku GeoJSON do GeoDataFrame
gdf = gpd.read_file('plik.geojson')

# SELECT * FROM gdf LIMIT 5
print(gdf.head())

# Definicja obszaru (bounding box)
minx, miny, maxx, maxy = 20, 51, 22, 53
bbox = box(minx, miny, maxx, maxy)

# Wybranie danych wewnątrz obszaru (bounding box)
print(gdf[gdf.intersects(bbox)])

# Konwersja geometrii do układu EPSG:4326 (WGS 84)
gdf_wgs = gdf.to_crs(epsg=4326)
print(gdf_wgs.head())
```