



Uniwersytet
Bielsko-Bialski

Regresja Liniowa na podstawie SVD

Zadanie Regresja

Sprawozdanie z ćwiczeń
Matematyka Konkretna

Data wykonania:
28.06.2025

Autor:
Bartosz Bieniek 058085

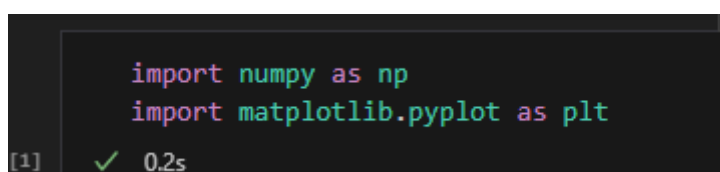
1. Cel ćwiczenia

Zadanie dotyczy obliczenia wieloliniowej regresji z użyciem macierzy pseudoodwrotnej dla zależności. Kod zamieszczono w repozytorium [Github](#).

2. Przebieg ćwiczenia

1. Import bibliotek

Zaimportowano biblioteki numpy oraz matplotlib, które umożliwiły przeprowadzenie obliczeń numerycznych oraz graficzną prezentację wyników. numpy wykorzystano do operacji macierzowych, a matplotlib.pyplot do wizualizacji danych i osi głównych.



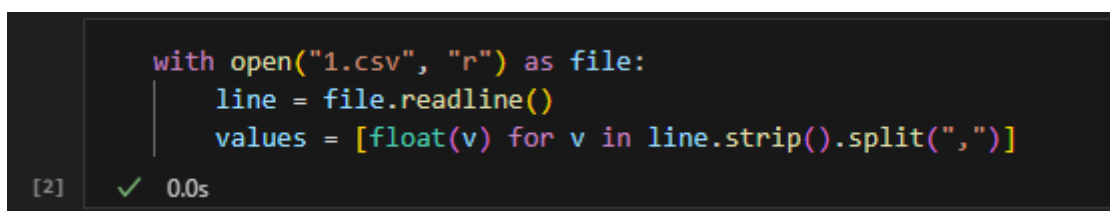
```
import numpy as np
import matplotlib.pyplot as plt
```

[1] ✓ 0.2s

Rys. 1. Import bibliotek.

2. Wczytanie danych z pliku CSV

Wczytano dane z pliku tekstowego w formacie .csv, zakładając, że dane zapisano jako jeden wiersz liczb oddzielonych przecinkami. Odczytane wartości przekonwertowano na typ float, co umożliwiło ich dalsze przetwarzanie.



```
with open("1.csv", "r") as file:
    line = file.readline()
    values = [float(v) for v in line.strip().split(",")]
```

[2] ✓ 0.0s

Rys. 2. Wczytanie danych z pliku CSV.

3. Przekształcenie danych do formy (n, 2)

Z wczytanego jednowymiarowego wektora danych utworzono macierz dwuwymiarową, w której każda para kolejnych wartości interpretowana była jako punkt (x,y) . Pozwoliło to na dalsze analizy geometryczne w przestrzeni dwuwymiarowej.

```
[3] data = np.array(values).reshape(-1, 2)
✓ 0.0s
```

Rys. 3. Przekształcenie danych do formy (n, 2).

4. Obliczenie środka ciężkości

Wyznaczono środek ciężkości (średnią arytmetyczną współrzędnych) dla wszystkich punktów. Punkt ten posłużył jako nowe odniesienie dla układu współrzędnych w dalszej analizie.

```
[4] center = np.mean(data, axis=0)
✓ 0.0s
```

Rys. 4. Obliczanie środka ciężkości.

5. Przesunięcie danych względem środka

Dla każdego punktu obliczono jego przemieszczenie względem środka ciężkości. Operacja ta zcentrowała dane wokół punktu (0, 0), co jest standardową procedurą przed obliczeniem macierzy kowariancji.

```
[5] centered_data = data - center
✓ 0.0s
```

Rys. 5. Przesunięcie danych względem środka.

6. Obliczenie macierzy kowariancji

Na zcentrowanych danych obliczono macierz kowariancji, która opisuje współzmiennność współrzędnych x i y. Macierz ta stanowi podstawę do dalszego wyznaczenia kierunków największej zmienności danych (czyli osi głównych).

```
[6] cov_matrix = np.cov(centered_data, rowvar=False)
✓ 0.0s
```

Rys. 6. Obliczanie macierzy kowariancji.

7. Obliczenie wartości i wektorów własnych

Z macierzy kowariancji wyznaczono wartości własne (opisujące wariancję wzdłuż osi) oraz odpowiadające im wektory własne (czyli kierunki tych osi). Posłużyły one do określenia orientacji i skali głównych kierunków rozrzutu danych.

```
eig_vals, eig_vecs = np.linalg.eigh(cov_matrix)
```

[7] ✓ 0.0s

Rys. 7. Obliczanie wartości i wektorów własnych.

8. Sortowanie od największej wartości własnej

Wartości własne oraz wektory własne posortowano malejąco względem wariancji, aby oś pierwsza odpowiadała kierunkowi największego rozrzutu danych. Ułatwiło to interpretację wyników oraz wizualizację orientacji danych.

```
order = np.argsort(eig_vals)[::-1]
eig_vals = eig_vals[order]
eig_vecs = eig_vecs[:, order]
```

[8] ✓ 0.0s

Rys. 8. Sortowanie

9. Obliczenie kąta obrotu (w radianach)

Na podstawie pierwszego wektora własnego wyznaczono kąt jego nachylenia względem osi x. Obliczony kąt odpowiada orientacji głównej osi danych względem układu współrzędnych i wyrażony został w radianach.

```
angle_rad = np.arctan2(eig_vecs[1, 0], eig_vecs[0, 0])
```

[9] ✓ 0.0s

Rys. 9. Obliczanie kąta obrotu

10. Wizualizacja wyniku

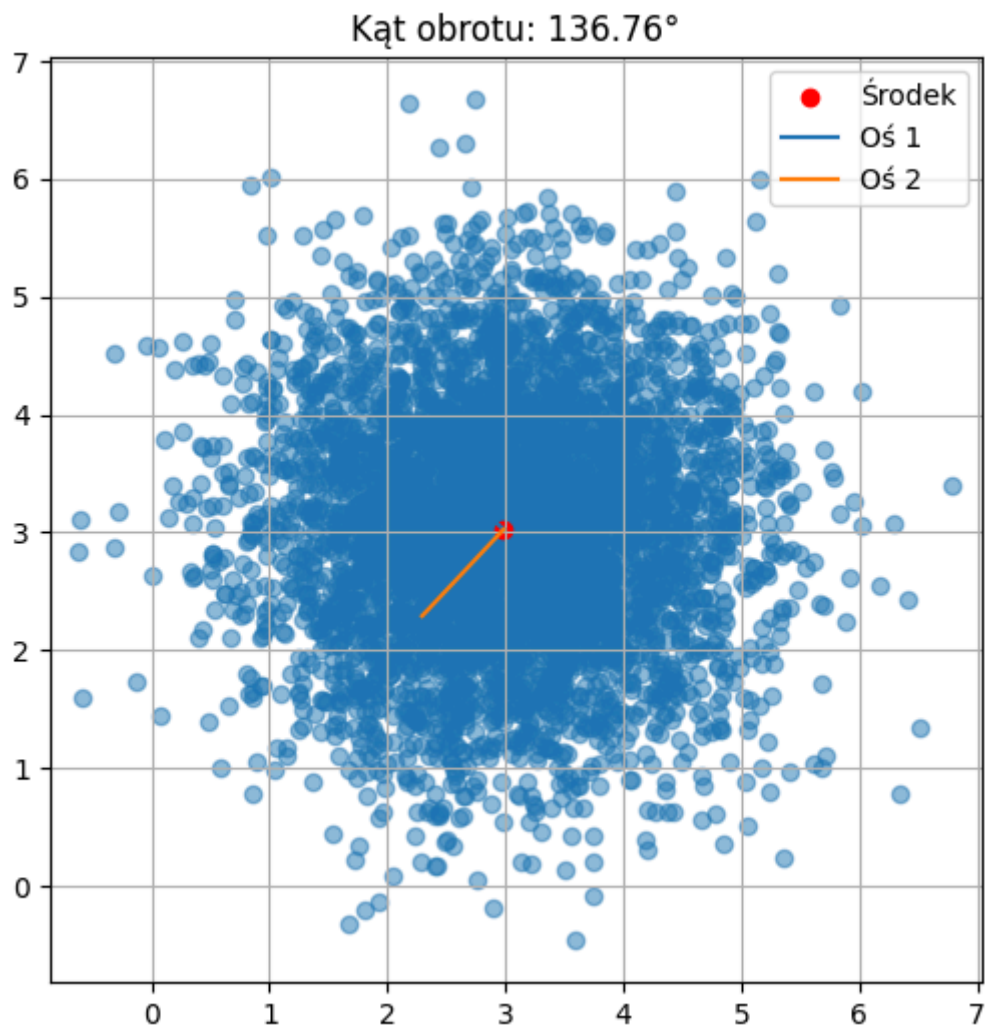
```

plt.figure(figsize=(6, 6))
plt.scatter(data[:, 0], data[:, 1], alpha=0.5)
plt.scatter(center[0], center[1], color='red', label='Środek')
for i in range(2):
    vec = eig_vecs[:, i]
    scale = np.sqrt(eig_vals[i])
    plt.plot(
        [center[0], center[0] + vec[0] * scale],
        [center[1], center[1] + vec[1] * scale],
        label=f'Oś {i+1}'
    )
plt.axis('equal')
plt.legend()
plt.title(f'Kąt obrotu: {np.degrees(angle_rad):.2f}°')
plt.grid(True)
plt.show()

```

[10] ✓ 0.1s

Rys. 10. Program wizualizacji



Rys. 11. Wizualizacja

3. Wnioski

Na podstawie przeprowadzonej analizy wyznaczono Środek geometryczny zbioru danych oraz główne kierunki ich rozrzutu w przestrzeni dwuwymiarowej. Otrzymane osie główne wskazały kierunki największej i najmniejszej wariancji, co pozwoliło na określenie orientacji danych względem układu współrzędnych. Obliczony kąt obrotu potwierdził, że dane są wyraźnie zorientowane wzdłuż określonego kierunku, co może mieć znaczenie w kontekście ich dalszego przetwarzania lub redukcji wymiarowości.