

Zadania związane z interpolacją `scipy.interpolate`

October 15, 2024

1 Podstawowa interpolacja liniowa

Napisz funkcję, która wykonuje interpolację liniową dla zestawu danych, a następnie pozwala użytkownikowi na wprowadzenie nowej wartości, dla której interpolacja zwróci wynik.

Wyniki

- Użytkownik wprowadza wartość `x_new`.
- Program oblicza i zwraca interpolowaną wartość `y_new` dla tej wartości, korzystając z interpolacji liniowej.
- Interpolacja zwraca poprawną wartość na podstawie istniejących danych.

Zestaw danych

```
import numpy as np

# Zestaw danych
x = np.array([0, 1, 2, 3, 4])
y = np.array([0, 1, 4, 9, 16])
```

2 Porównanie różnych metod interpolacji

Napisz program, który wykonuje interpolacje dla zestawu danych przy użyciu trzech różnych metod interpolacji: liniowej, kwadratowej i sześciennej. Następnie narysuj wykres, który porównuje oryginalne dane z wynikami każdej metody interpolacji.

Wyniki

- Program generuje wykres porównujący oryginalne dane z wynikami interpolacji dla trzech metod:
 - Interpolacja liniowa
 - Interpolacja kwadratowa
 - Interpolacja sześcienna
- Na wykresie są wyraźnie widoczne różnice między metodami interpolacji, a dane oryginalne są zaznaczone jako punkty.

Zestaw danych

```
import numpy as np

# Zestaw danych
x = np.array([0, 1, 2, 3, 4, 5])
y = np.array([0, 1, 8, 27, 64, 125])
```

3 Interpolacja wielowymiarowa

Zaimplementuj interpolację wielowymiarową dla danych dwuwymiarowych (np. dane 2D w formie siatki). Stwórz funkcję, która interpoluje wartości na siatce 2D przy użyciu funkcji `scipy.interpolate.griddata`.

Wyniki

- Program generuje siatkę punktów 2D i wykonuje interpolację dla wartości na tej siatce.
- Wynik interpolacji jest wyświetlany jako obraz 2D z kolorami reprezentującymi interpolowane wartości.
- Oryginalne punkty i wartości są widoczne na tym samym wykresie.

Zestaw danych

```
import numpy as np

# Zestaw danych 2D
points = np.random.rand(100, 2)
values = np.sin(points[:, 0] * 10) * np.cos(points[:, 1] * 10)
```

4 Interpolacja z brakującymi wartościami

Napisz program, który interpoluje brakujące wartości w zestawie danych. Dane mogą zawierać brakujące wartości oznaczone jako `None` lub `NaN`, a program powinien interpolować te wartości, wykorzystując sąsiednie punkty.

Wyniki

- Program identyfikuje brakujące wartości w zestawie danych (oznaczone jako `None` lub `NaN`).
- Wykonuje interpolację na podstawie istniejących wartości, pomijając brakujące elementy.
- Brakujące wartości są zastępowane wynikami interpolacji, a dane są zwracane jako kompletne.
- Oryginalne i uzupełnione dane mogą być wyświetlane do porównania.

Zestaw danych

```
import numpy as np

# Zestaw danych z brakującymi wartościami
x = np.array([0, 1, 2, 3, 4, 5])
y = np.array([0, 1, np.nan, 9, np.nan, 25])
```