

# Datenverarbeitung und maschinelles Lernen WS

2024/25

Übung 2

NumPy

NumPy ist eine Python-Bibliothek, die für numerische Aufgaben und zur Verarbeitung mehrdimensionaler Arrays verwendet wird.

## Aufgabe 1: Untersuchung der Array-Eigenschaften

Erstellen Sie einen 5x2-Array mit ganzen Zahlen (int) und geben Sie folgende Informationen zurück: Shape, Dimensions, Größe der Elemente in Bytes.

## Aufgabe 2: Arrange

Erstellen Sie einen 5x2-Array mit Werten von 100 bis 200 und einer Schrittweite von 20. Printen Sie den Array.

## Aufgabe 3: Spalten

Erstellen Sie ein 4x4-Array und geben Sie nur die Elemente in der dritten Spalte zurück.

## Aufgabe 4: Rechnen mit Arrays

Gegeben sind zwei Arrays:

```
array1 = np.array([[8, 7, 6], [5, 4, 3]])    array2 = np.array([[2, 1, 0], [9, 8, 7]])
```

Führen Sie eine elementweise Subtraktion durch und quadrieren Sie anschließend jedes Element des Ergebnisses.

## Aufgabe 5: Aufteilen eines Arrays

Erstellen Sie einen 8x2 Array mit Werten zwischen 50 und 65 und teilen Sie ihn in vier gleiche Teile. Printen Sie diese.

## Aufgabe 6: Arrays sortieren

Sortieren Sie die folgende 4x3-Matrix:

```
sample_matrix = np.array([[10, 40, 30], [70, 20, 60], [50, 90, 80], [15, 25, 35]])
```

Sortieren Sie zuerst nach den Werten in der letzten Zeile und dann nach den Werten in der zweiten Spalte.

## Aufgabe 7: Ersetzen einer Spalte

Gegeben sei die 4x3-Matrix:

```
matrix = np.array([[12, 22, 32], [42, 52, 62], [72, 82, 92], [102, 112, 122]])
```

Ersetzen Sie die letzte Spalte durch ein Array von Nullen.

## Aufgabe 8: Matrixoperationen

Erstellen Sie eine 2x2-Matrix:

`matrix = np.array([[7, 5], [3, 2]])`. Berechnen Sie ihre Spur und Determinante und ermitteln Sie anschließend die Inverse, wenn die Matrix invertierbar ist.

## Aufgabe 9: Statistik

Ihnen stehen 4 Datenreihen zur Verfügung. Berechnen Sie mit NumPy den Mittelwert, den Median, die Quartile und die Standardabweichung für jede Datenreihe.

`x1 = np.array([1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0])`

`y1 = np.array([2.1, 3.9, 5.8, 7.7, 11.0, 12.0, 14.3, 16.2, 19.0, 19.9])`

`x2 = np.array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])`

`y2 = np.array([1.23, 1.48, 1.79, 2.16, 3.00, 3.32, 4.18, 5.05, 6.69, 7.32])`

## Aufgabe 10: Empirische Varianz

Teil 1: Empirische Varianz verstehen

Erklären Sie, was mit „empirischer Varianz“ gemeint ist. Warum dividieren wir bei der Berechnung der empirischen Varianz für eine Stichprobe durch  $n-1$  statt durch  $n$ ?

Teil 2: Berechnung mit Python

- Gegeben seien die folgenden Beispieldaten: `x=[4,8,6,5,3,7,9,5]`
- Schreiben Sie ein Python-Skript, um die empirische Varianz mit der folgenden Formel zu berechnen:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Teil 3: Herleitung

- Leiten Sie die vereinfachte Formel für die empirische Varianz mithilfe der obigen Gleichung her:

$$s^2 = \frac{1}{n-1} \left[ \left( \sum_{i=1}^n x_i^2 \right) - n\bar{x}^2 \right]$$

## Aufgabe 11: Datenvorverarbeitung

Beschreiben Sie die Bedeutung der Datenvorverarbeitung. Erläutern Sie drei Methoden zum Umgang mit fehlenden Daten und geben Sie konkrete Situationen an, in denen jede Technik anwendbar ist.