Python - NumPy

Jan Popko
Python Grundkurs

Datascience

- die Wissenschaft großer Datenmengen
- es gibt einen Datascience-Stack in Python (eine Sammlung von Programmen, Modulen und Klassen zur Verarbeitung von großen Datenmengen)
- zu den Modulen gehören unter anderen: NumPy, matplotlib, pandas…
- es gibt noch viel mehr Tools zum Auswerten und Bearbeiten von großen Datenmengen

NumPy

- NumPy eignet sich hervorragend für mathematische Berechnungen
- im Vordergrund steht das Rechnen mit Arrays
- NumPy gehört nicht zum Standardumfang von Python
- wir müssen es also nachinstallieren
- Befehl für die Konsole: **pip3 install numpy**

Business Trends Academy (BTA) GmbH
bta BUSINESS TRENDS ACADEMY
Nestorstraße 36
D-10709 Berlin
Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94
Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179

NumPy

Aufgabe:

Schreiben Sie ein Script (ohne NumPy), welches eine Liste von 6 Ganzzahlen erstellt.
Jetzt soll jede Zahl in dieser Liste um 3 erhöht werden und in einer neuen Liste gespeichert und ausgeben werden!

1, 5, 8, 4, 7, 3 → 4, 8, 11, 7, 10, 6

NumPy

Lösung:

```
z = [1, 5, 8, 4, 7, 3]
z2 = []
for element in z:
  z2.append(element + 3) # oder
  z2 += [element + 3]

print(z2)
```

NumPy

NumPy-Arrays:

```
# NumPy importieren und kürzeren Namen vergeben
import numpy as np

#ein NumPy-Array wird erzeugt
a = np.array([1, 2, 3])

# Typ von a wird ausgegeben
print(type(a))

# die Ausgabe mit print() ähnelt der Darstellung von Matrizen
# KEINE KOMMAS!
print(a)
```


Business Trends Academy (BTA) GmbH
Nestorstraße 36
D-10709 Berlin
Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94
Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179

NumPy

Rechenoperationen:

```
# jedes Element des Arrays a wird mit 3 addiert und das neue
  Array in b gespeichert
b = a + 3

# Ausgabe von b
print(b)

# jedes Element des Arrays a wird mit 5 multipliziert und das
  neue Array in c gespeichert
c = a * 5

# Ausgabe von c
print(c)
```

NumPy

Rechenoperationen:

Mittelwert:

```
mw = np.array([1, 5, 8, 4, 7, 3, 2, 2, 6, 8]).mean()
print(mw)
```

Minimum:

```
mw = np.array([1, 5, 8, 4, 7, 3, 2, 2, 6, 8]).min()
print(mw)
```

Maximum:

```
mw = np.array([1, 5, 8, 4, 7, 3, 2, 2, 6, 8]).max()
print(mw)
```

NumPy

Aufgabe:

Schreiben Sie ein Script (**ohne NumPy**), welches folgende Angaben (Grad Celsius) in Grad Fahrenheit umrechnet!

20.1, 20.8, 21.9, 22.5, 22.7, 21.8, 21.3, 20.9, 20.1 →

68.18 69.44 71.42 72.5 72.86 71.24 70.34 69.62 68.18

NumPy

Lösung:

```
celsius = [20.1, 20.8, 21.9, 22.5, 22.7, 21.8, 21.3, 20.9, 20.1]

fahr = []
for element in celsius:
    fahr += [element * 9 / 5 + 32]

print(fahr)
```

NumPy

Aufgabe:

Schreiben Sie ein Script (**mit NumPy**), welches folgende Angaben in Grad Celsius in Grad Fahrenheit umrechnet!

20.1, 20.8, 21.9, 22.5, 22.7, 21.8, 21.3, 20.9, 20.1 →

68.18 69.44 71.42 72.5 72.86 71.24 70.34 69.62 68.18

NumPy

Lösung:

```
celsius = [20.1, 20.8, 21.9, 22.5, 22.7, 21.8, 21.3, 20.9, 20.1]

f = np.array(celsius) * 9 / 5 + 32

print(f)
```

NumPy

Aufgabe:

Schreiben Sie ein Script (**ohne NumPy**), welches eine Liste von 10 Ganzzahlen erstellt.
Jetzt soll der Mittelwert (Durchschnitt) aller Zahlen in der Liste ermittelt werden!

1, 5, 8, 4, 7, 3, 2, 2, 6, 8 → 4.6

NumPy

Lösung:

```
# Mittelwert
z = [1, 5, 8, 4, 7, 3, 2, 2, 6, 8]
summe = 0

for element in z:
    summe += element

mw = summe / len(z)

print(mw)
```

NumPy

Aufgabe:

Schreiben Sie ein Script (**mit NumPy**), welches eine Liste von 10 Ganzzahlen erstellt.
Jetzt soll der Mittelwert (Durchschnitt) aller Zahlen in der Liste ermittelt werden!

1, 5, 8, 4, 7, 3, 2, 2, 6, 8 → 4.6

Business Trends Academy (BTA) GmbH
bta BUSINESS TRENDS ACADEMY
Nestorstraße 36
D-10709 Berlin
Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94
Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179

NumPy

Lösung:

```
mw = np.array([1, 5, 8, 4, 7, 3, 2, 2, 6, 8]).mean()
print(mw)
```

NumPy

Mehrdimensionale Arrays:

```
#ein mehrdimensionales NumPy-Array wird erzeugt
d = np.array([[1, 0], [2, 1]])

# die Ausgabe mit print() ähnelt der Darstellung von Matrizen
# KEINE KOMMAS!
print(d)
```

Angabe des Datentyps:

boolean (0 ist False, alle anderen Zahlen ergeben True)

```
e = np.array([[1, 0], [2, 1]], bool)
print(e)
```

float:

```
f = np.array([[1, 0], [2, 1]], float)
print(f)
```

NumPy

Aufgabe:

Erstellen Sie folgendes mehrdimensionales Array:

```
1 5 8 4
0 3 6 9
4 7 9 2
```

NumPy

Lösung:

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

print(a)
```

NumPy

weitere Funktionen zur Erzeugung von Arrays:

```
#liefert ein Array mit gleichmäßig verteilten Zahlenwerten von
#Start bis Stop, num ist die Anzahl der Zahlen(Default: 50)
z = np.linspace(4, 18, 20)
print(z)

# erzeugt ein Array mit einer Zahlenfolge
# arange(start, stop, step)
z = np.arange(4, 47, 3)
print(z)

# erzeugt ein Array gefüllt mit Einsen
z = np.ones(4) # 4 Elemente
print(z)

z2 = np.ones((4, 3)) # 4 Zeilen | 3 Spalten
print(z2)
```

NumPy

weitere Funktionen zur Erzeugung von Arrays:

```
# erzeugt ein Array gefüllt mit Nullen
z = np.zeros(4) # 4 Elemente
print(z)

# Zufallsarray mit ganzen Zahlen
rand = np.random.randint(0, 100, 10)
print(rand)

# Zufallsarray der Länge 10 mit ganzen Zahlen von 0 bis 99
rand = np.random.randint(0, 100, 10)
print(rand)

# Zufallsarray (Zahlen von 0 bis 99) mit 4 Zeilen und 3 Spalten
rand2 = np.random.randint(0, 100, (4, 3))
print(rand2)
```

NumPy

Vektoren:

```
#Zeilenvektor
v1 = np.array([[1, 9, 1]])
print(v1)

#Spaltenvektor
v2 = np.array([[1], [9], [1]])
print(v2)
```

NumPy

Slicing:

```
# NumPy importieren und kürzeren Namen vergeben
import numpy as np

# eindimensionales Array von 0 bis 8 erzeugen
a = np.arange(9)
print(a)
#Slicing - Elemente von a[0] bis a[3]
print(a[0:4])
#Slicing - Elemente vor a[4]
print(a[:4])
#Slicing - Elemente nach a[3]
print(a[4:])
```


Business Trends Academy (BTA) GmbH
Nestorstraße 36
D-10709 Berlin
Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94
Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179

NumPy

Arrays verändern mit reshape():

```
# NumPy importieren und kürzeren Namen vergeben
import numpy as np

# eindimensionales Array von 0 bis 11 erzeugen
a = np.arange(12)
print(a)
#neues Array mit veränderter Struktur: 3 Zeilen, 4 Spalten
b = a.reshape(3, 4)
print(b)
```


Business Trends Academy (BTA) GmbH
Nestorstraße 36
D-10709 Berlin
Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94
Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179

NumPy

Arrays verändern mit ravel():

```
# NumPy importieren und kürzeren Namen vergeben
import numpy as np

# Array mit 2 Zeilen und 2 Spalten erzeugen
a = np.array([[10, 20], [50, 60]])
print(a)
# ravel() macht aus dem mehrdimensionalen Array ein
# eindimensionales
b = a.copy().ravel()
print(b)
```


Business Trends Academy (BTA) GmbH
Nestorstraße 36
D-10709 Berlin
Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94
Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179

NumPy

Arrays sortieren mit sort():

```
ar= np.random.randint(0, 11, 5)
print(ar)


# Aufsteigend
ar_sort = np.sort(ar)
print(ar_sort)


# Absteigend
ar_sort = -np.sort(-ar)
print(ar_sort)
```

NumPy

Mathematische Funktionen:

- **sin()**
- **cos()**
- **tan()**
- **sqrt()**
- **log()**

NumPy

Wertetabelle für Sinusfunktion:

```
# NumPy importieren und kürzeren Namen vergeben
import numpy as np

# eindimensionales Array von 0 bis 10 erzeugen, x-Koordinate
x = np.arange(11)
print(x)
# y-Koordinate für f(x) = sin(x)
y = np.sin(x)
print(y)
```

einfache Visualisierung

- für einfache Darstellungen von Graphen kann das Modul Matplotlib verwendet werden
- Matplotlib gehört nicht zum Standardumfang von Python
- wir müssen es also nachinstallieren
- Befehl für die Konsole: **pip3 install matplotlib**

NumPy und Matplotlib

```
# NumPy importieren und kürzeren Namen vergeben
import numpy as np

# Matplotlib einbinden und kürzeren Namen vergeben
import matplotlib.pyplot as plt

# eindimensionales Array von 0 bis 20 erzeugen, x-Koordinate,
# Schrittweite 0.2
x = np.arange(0, 20, 0.2)
# y-Koordinate für f(x) = sin(x)
y = np.sin(x)

# Fenster wird erzeugt, x und y werden übergeben, Fenster wird
# angezeigt
plt.plot(x, y)
plt.show()
```