Introduction to NumPy

Hello, everyone!

Sebelum kita memulai kelas, kita awali dengan:

1. Berdoa
2. siapkan diri

Pertemuan lalu...

Let’s Recall

pertemuan sebelumnya kita sudah belajar fungsi pada python. Ada yang inget apa aja?

Target

1. What is Numpy?
2. Numpy usage?
3. Creating matrix in Numpy
4. Array manipulation in Numpy

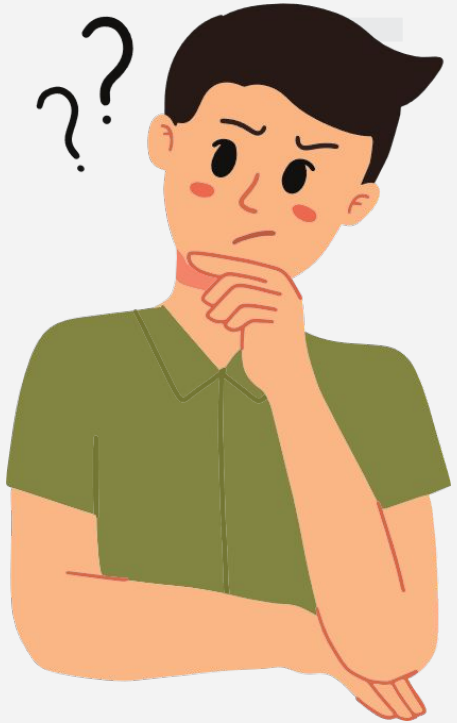
Tools

https://colab.research.google.com/

1. What is Numpy

Apaitu Numpy?

- NumPy, yang merupakan singkatan dari Numerical Python, adalah perpustakaan yang terdiri dari objek array multidimensi dan kumpulan rutinitas untuk memproses array tersebut.
- Numpy ini menggunakan array atau tempat penyimpanan dan pengelompokkan data dengan tipe yang sama.
- NumPy dapat melakukan operasi matematika dan logika pada array dapat dilakukan.
- Numpy ini merupakan library open source, artinya bisa diakses siapa saja secara gratis

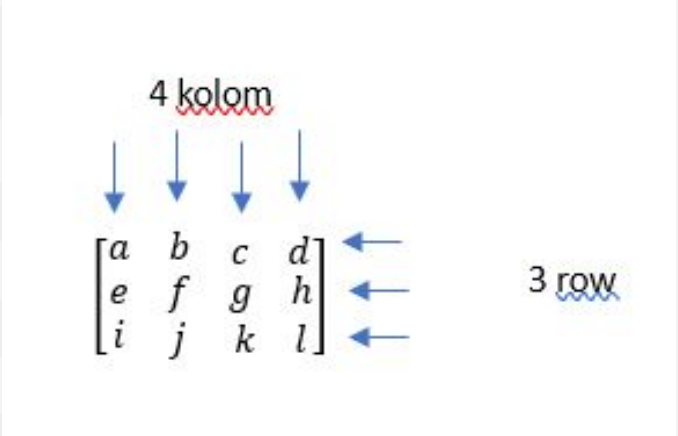


2. Numpy usage

3. Creating matrix in Numpy

Pengenalan Matriks

- Matriks adalah struktur data dua dimensi yang bilangan-bilangannya disusun menjadi baris dan kolom.
- Kita dapat membuat matriks di Numpy menggunakan fungsi seperti array(), ndarray() atau matriks(). Fungsi matriks secara default membuat larik 2D khusus dari masukan yang diberikan. Inputnya harus dalam bentuk string atau seperti objek array.
- Contoh Matriks ini merupakan matriks berukuran 3x4. karena mempunyai 3 baris dan 4 kolom.



Python Matrix

```
[9] import numpy as np
    #creating matrix from string
    A = np.matrix('1,2,3,4;5,6,7,8;9,10,11,12')
    print("Array created using string is :\n", A)
```

```
Array created using string is :
 [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

Contoh Matriks

- Line 1 merupakan contoh matriks 3x4
- B[1] artinya kita akan memanggil baris ke-2 saja

```
[1] import numpy as np
    #creating matrix from string
    B = np.matrix('1, 4, 5, 12; -5, 8, 9, 0;-6, 7, 11, 19')
    print("Array created using string is :\n", B)
```

```
Array created using string is :
 [[ 1  4  5 12]
 [-5  8  9  0]
 [-6  7 11 19]]
```

```
[2] type(B)
```

```
numpy.matrix
```

```
[3] print("B =", B)
    print("B[1] =", B[1])      # 2nd row
```

```
B = [[ 1  4  5 12]
 [-5  8  9  0]
 [-6  7 11 19]]
B[1] = [[-5  8  9  0]]
```

Matrix Functions in NumPy

- Transpose
  - fungsi: matrix.T
  - contoh gambar line 4

```
[4] B.T

matrix([[ 1, -5, -6],
        [ 4,  8,  7],
        [ 5,  9, 11],
        [12,  0, 19]])
```

- Inverse
  - fungsi: matrix.I
  - contoh gambar line 6

```
[5] C = np.matrix('1, 4, 5; -5, 8, 9;-6, 7, 19')
    print("Array created using string is :\n", C)

Array created using string is :
 [[ 1  4  5]
 [-5  8  9]
 [-6  7 19]]
```

```
[6] C.I

matrix([[ 0.27987421, -0.12893082, -0.01257862],
        [ 0.12893082,  0.15408805, -0.10691824],
        [ 0.0408805 , -0.09748428,  0.08805031]])
```

Matrix Functions in NumPy

- Diagonal
  - fungsi: matrix.diagonal()
  - contoh gambar line 9

```
[9]  C.diagonal()

matrix([[ 1,  8, 19]])
```

- Maximum and Minimum Indices
  - fungsi: matrix.argmax() dan matrix.argmin()
  - contoh gambar line 16 adalah contoh maksimum
  - contoh gambar line 19 adalah contoh minimum

```
[15] C

matrix([[ 1,  4,  5],
        [-5,  8,  9],
        [-6,  7, 19]])

[16] C.argmax(0)

matrix([[0, 1, 2]])

[18] C.argmin(0)

matrix([[2, 0, 0]])
```

Matrix Functions in NumPy

- Cumulative Sum and Product of a Given Matrix
  - fungsi: matrix.cumprod() kumulatif product
  - fungsi: matrix.cumsum() kumulatif sum
  - contoh gambar line 23 dan 24 kumulatif product. 0 artinya bari dan 1 artinya kolom
  -

```
[22] C

matrix([[ 1,  4,  5],
        [-5,  8,  9],
        [-6,  7, 19]])
```

```
[23] C.cumprod(0)

matrix([[  1,   4,   5],
        [ -5,  32,  45],
        [ 30, 224, 855]])
```

```
[24] C.cumprod(1)

matrix([[   1,    4,   20],
        [  -5,  -40, -360],
        [  -6,  -42, -798]])
```

```
[25] C.cumsum(0)

matrix([[  1,   4,   5],
        [ -4,  12,  14],
        [-10,  19,  33]])
```

```
[26] C.cumsum(1)

matrix([[ 1,  5, 10],
        [-5,  3, 12],
        [-6,  1, 20]])
```

Matrix Functions in NumPy

- Dot Product
  - fungsi: matrix.dot()
  - contoh gambar line 9, artinya martix C akan dikalikan 5 setiap element nya

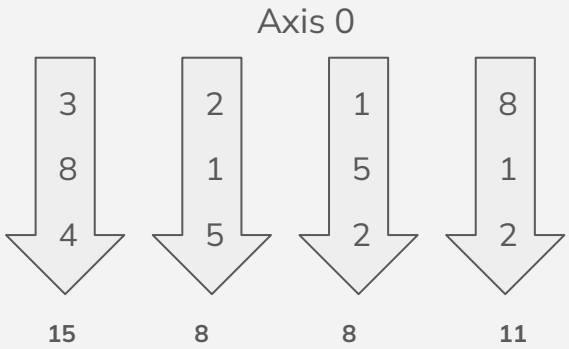
```
[28] C

     matrix([[ 1,  4,  5],
             [-5,  8,  9],
             [-6,  7, 19]])

[29] C.dot(5)

     matrix([[  5,  20,  25],
             [-25,  40,  45],
             [-30,  35,  95]])
```

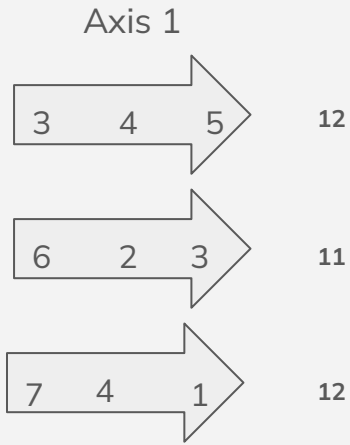
Advantages of Matrix in NumPy

- Salah satu keuntungan utama menggunakan matriks NumPy adalah matriks ini menggunakan lebih sedikit ruang memori dan memberikan kecepatan runtime yang lebih baik jika dibandingkan dengan struktur data serupa di python (daftar dan tupel).
- Matriks NumPy mendukung beberapa fungsi ilmiah tertentu seperti jumlah kumulatif berdasarkan elemen, hasil kali kumulatif, transpose konjugasi, dan invers perkalian, dll. Daftar atau string python gagal mendukung fitur ini.

4. Array manipulation in Numpy

Pendahuluan





Kita dapat dengan mudah membentuk ulang, menggabungkan, dan membagi array dengan fitur manipulasi array ekstensif NumPy. Contoh manipulasi array:

- Reshaping Arrays
- Concatenating Arrays
- Splitting Numpy Arrays

Creating Arrays

Langkah pertama dalam memanipulasi array adalah membuat array. NumPy menyediakan berbagai fungsi untuk membuat array dengan bentuk dan tipe data berbeda. Berikut adalah beberapa fungsi pembuatan array yang paling umum digunakan:

- `np.array` : membuat array dari daftar atau tuple Python
- `np.zeros` : membuat array yang diisi dengan nol
- `np.ones` : membuat array berisi satuan
- `np.empty` : membuat array yang tidak diinisialisasi dengan nilai acak
- `np.arange` : membuat array dengan nilai yang berjarak sama
- `np.linspace` : membuat larik dengan sejumlah nilai dengan jarak yang sama antara titik awal dan akhir.

Reshaping Arrays

```
[32] import numpy as np

     my_array = np.array([1, 2, 3, 4, 5, 6])

     # Reshape the array to a 3x2 matrix
     reshaped_array = my_array.reshape(3,2)

     print("Original Array:\n", my_array)
     print("Reshaped Array:\n", reshaped_array)
```

```
Original Array:
 [1 2 3 4 5 6]
Reshaped Array:
 [[1 2]
 [3 4]
 [5 6]]
```

Dengan menggunakan fungsi reshape() Anda dapat mengubah bentuk array. Pastikan pembentuk array baru memiliki jumlah komponen yang sama dengan array lama.

Concatenating Arrays

```
[35] import numpy as np

     array1 = np.array([1, 2, 3])
     array2 = np.array([4, 5, 6])

     # Concatenate the arrays vertically
     vertical_concatenation = np.concatenate((array1, array2))

     # Concatenate the arrays horizontally
     horizontal_concatenation = np.concatenate((array1, array2), axis=0)

     print("Vertical Concatenation:", vertical_concatenation)
     print("Horizontal Concatenation:", horizontal_concatenation)

     Vertical Concatenation: [1 2 3 4 5 6]
     Horizontal Concatenation: [1 2 3 4 5 6]
```

Kita dapat menggabungkan (kombinasikan) dua atau lebih array menggunakan fungsi seperti numpy.concatenate(). Pastikan dimensi sepanjang sumbu yang ditentukan cocok.

Splitting Numpy Arrays

```
[36] import numpy as np

     my_array = np.array([1, 2, 3, 4, 5, 6])

     # Split the array into three equal parts
     split_arrays = np.split(my_array, 3)

     print("Split Arrays:", split_arrays)

     Split Arrays: [array([1, 2]), array([3, 4]), array([5, 6])]
```

Kita dapat membagi array menjadi beberapa array yang lebih kecil menggunakan fungsi seperti numpy.split() atau numpy.hsplit() untuk pemisahan horizontal.

Aggregation Functions in Numpy

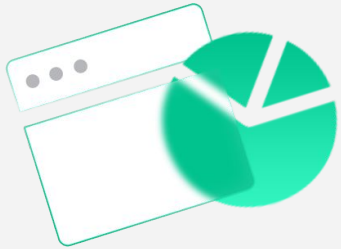
```
[37] import numpy as np

     array1 = np.array([1, 2, 3, 4, 5])

     total = np.sum(array1)

     print("Array:", array1)
     print("Sum:", total)

     array2= np.array([3, 1, 2, 4, 5])

     mean = np.mean(array2)
     median = np.median(array2)

     print("Array:", array2)
     print("Mean:", mean)
     print("Median:", median)
```

```
Array: [1 2 3 4 5]
Sum: 15
Array: [3 1 2 4 5]
Mean: 3.0
Median: 3.0
```

Banyak fungsi agregasi tersedia di NumPy. Kita dapat menggunakan fungsi NumPy untuk menghitung statistik seperti jumlah, mean, median, dan lainnya.

NEXT!!

Introduction to Pandas (Basic Dataframe)



Terima Kasih