Numpy

April 4, 2019

0.1 Knowledge Discovery Process

```
Problem Definition

Data Collection

Data Preprocessing

Data Transformation

Data Mining

Data Analysis

Data Visualization
```

1 NumPy

An essential library used for scientific computing in Python.

Holds data in N-dimensional array (ndarray) objects, which can store data in multiple dimensions.

Supports performing efficient array operations through Broadcasting feature.

```
In [ ]: import numpy as np
        import matplotlib.pyplot as plt

In [ ]: array=np.array([[1,2],[2,3]])

In [ ]: type(array)

In [ ]: y=array

In [ ]: print(y.ndim, y.shape, y.size, y.dtype, y.itemsize, y.nbytes)

In [ ]: array=np.array([[1,2,3],[3,4,5]],dtype='float64')

In [ ]: array

In [ ]: array.dtype
```

```
In [ ]: array.flags
```

Now let us focus on creating ndarray,

From Python built-in datatypes : lists or tuples

Using Numpy array creation methods like ones, ones_like, zeros, zeros_like

Using Numpy numeric sequence generators.

Using Numpy random module.

By reading data from a file.

1.0.1 1. ndarrays from Lists

```
In [ ]:
```

Example 1: Using zeros method

```
In [ ]: np.zeros(shape=(2,4))
```

```
In [ ]: np.ones(shape=(2,4))
```

```
In [ ]: np.full(shape=(2,3), fill_value=10.5)
```

arrange method numpy.arange([start,]stop, [step,]dtype=None)

```
In [ ]: np.arange(1,10,2)
```

numpy.linspace(start, stop, #num inbetween, endpoint=True, retstep=False, dtype=None)

```
In [ ]: np.linspace(1,10,30)
```

Random Numbers Generator

```
In [ ]: np.random.seed(10)
```

```
In [ ]: x = np.random.rand(2) # 2 random numbers between 0 and 1
        print(x)
```

```
In [ ]: x=np.random.randn(10)
```

```
In [ ]: plt.hist(x,bins=4)
```

```
In [ ]: np.random.seed(100)
        x = 10 + 2*np.random.randn(3) # normal distribution with mean 10 and sd 2
```

```
In [ ]: x
```

In [ ]:
```
#reading from a file or a string
from io import StringIO
import numpy as np

x = StringIO('''88.25 93.45 72.60 90.90
72.3 78.85 92.15 65.75
90.5 92.45 89.25 94.50
''')

d = np.loadtxt(x,delimiter=' ')

print(d)

#print(d.ndim, d.shape)
```

In [ ]:
```
x = np.array([[-1,0,1], [-2, 0, 2]])

y = np.zeros_like(x)
print(y)
```

In [ ]:
```
z = np.eye(2)
print(z)
```

In [ ]:
```
a = np.array(np.linspace(0,1,24).reshape(2,3,4))
```

1.1 vstack and hstack

In [ ]:
```
a=np.arange(1,5,1).reshape(2,2)
a
```

In [ ]:
```
b=np.linspace(1,20,4).reshape(2,2)
b
```

In [ ]:
```
np.hstack((a,b))
```

In [ ]:
```
np.vstack((a,b))
```

1.2 splitting arrays

In [ ]:
```
b
```

In [ ]:
```
array1,array2=np.vsplit(b,2)
```

In [ ]:
```
array1
```

In [ ]:
```
array2
```

horizontal

```
In [ ]: array1,array2=np.hsplit(b,2)

In [ ]: array1

In [ ]: array2

In [ ]: x=np.arange(90).reshape(3, 15, 2)

In [ ]: x=np.array([1,2,3,4]).reshape(1,4)

In [ ]: x

In [ ]: array=np.array(np.linspace(1,10,8).reshape(2,4))

In [ ]: array

In [ ]: array[1:2,1:3]

In [ ]: x = np.array([[-1, 1], [-2, 2]])


        for row in x:

            print('Row :',row)

In [ ]: x = np.array([[0,1], [2, 3]])


        for a in np.nditer(x):

            print(a)
```