ml-assing2

March 14, 2024

[3]:
```
import numpy as np
```

[2]:
```
print(np.__version__)
```

```
1.26.3
```

[36]:
```
test = np.array([1,2,3,4,5])
test1 = np.array([(1,2,3), (4,5,6)])
```

[10]:
```
print(test)
print(test1)
```

```
[1 2 3 4 5]
[[1 2 3]
 [4 5 6]]
```

[21]:
```
print("shape of the array:", test.shape)
print("length of the array: ", len(test))
print("dimensions of the array: ", test.ndim)
print("data type of array a: ", test.dtype)
```

```
shape of the array: (5,)
length of the array:  5
dimensions of the array:  1
data type of array a:  float64
```

[22]:
```
print("shape of the array:", test1.shape)
print("length of the array: ", len(test1))
print("dimensions of the array: ", test1.ndim)
print("data type of array a: ", test1.dtype)
```

```
shape of the array: (2, 3)
length of the array:  2
dimensions of the array:  2
data type of array a:  int32
```

[23]:
```
c = test.astype(int)#convert array into other datatype
print(c)
```

```
[1 2 3 4 5]
```

[27]:
```
d = np.array([[(1,2,3), (4,5,6)], [(7,8,9), (10,11,12)]])
print(d)
print("shape of the array:", d.shape)
print("length of the array: ", len(d))
print("dimensions of the array: ", d.ndim)
print("data type of array a: ", d.dtype)
```

```
[[[ 1  2  3]
  [ 4  5  6]]

 [[ 7  8  9]
  [10 11 12]]]
shape of the array: (2, 2, 3)
length of the array:  2
dimensions of the array:  3
data type of array a:  int32
```

[13]:
```
t1 = np.zeros((3,4))
print(t1)
t2 = np.ones((3,4))
print(t2)
f = np.arange(10,25,2)
print(f)
h = np.linspace(0,2,9) # equal distance from each element from 1 to 2
print(h)
r = np.random.random((2,3))
print(r)
e = np.empty((2,2))
print(e)
i = np.eye(3)
print(i)
```

```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
[10 12 14 16 18 20 22 24]
[0.   0.25 0.5  0.75 1.   1.25 1.5  1.75 2.  ]
[[0.97466918 0.94023064 0.51741172]
 [0.38217297 0.42256136 0.00248236]]
[[2.54639495e-313 3.39519327e-313]
 [4.24399158e-313 5.09278990e-313]]
[[1. 0. 0.]
 [0. 1. 0.]
```

```
 [0. 0. 1.]]
```

[19]:
```
#loading and saving array
np.save("D:\MIT ADT\Third Year Sem - 2\ML LAB\my_array", test)

temp = np.load("D:\MIT ADT\Third Year Sem - 2\ML LAB\my_array.npy")
print(temp)
```

```
[1.6 2.  3.  4.  5. ]
```

[37]:
```
temp.sum()
print(temp.min())

print(temp.max(axis=0))
print(test1)
print("test 1 - 0: ", test1.max(axis=0))
print("test 1 - 1: ", test1.max(axis=1))

print(np.median(test))
print(np.std(test))

print(np.transpose(test1))
```

```
1.6
5.0
[[1 2 3]
 [4 5 6]]
test 1 - 0:  [4 5 6]
test 1 - 1:  [3 6]
3.0
1.4142135623730951
[[1 4]
 [2 5]
 [3 6]]
```

[34]:
```
test[2:]
```

[34]:
```
array([3, 4, 5])
```

[43]:
```
k =  (test1.ravel()) #more dimension to 1d flattning of the matrix
print(k)

r = k.reshape(2,3)# should be proportion
print(r)
```

```
[1 2 3 4 5 6]
[[1 2 3]
 [4 5 6]]
```

[59]:
```
my_array = np.array([1,2,3,4,5])
z = (np.resize(my_array,2))
print(z)

print(np.append(my_array, 8))

print(np.insert(my_array,[2], 5))

print(np.delete(my_array,[2]))

print(np.dot(test, test))
```

```
[1 2]
[1 2 3 4 5 8]
[1 2 5 3 4 5]
[1 2 4 5]
55
```

[ ]:
```

```