Task-6

Numpy:-

- import numpy as np
- Take less Time

ex. np.mean(x)

- Time: 0,09 قليل

- Create array

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

x.dtype => int64

-x.shape =>(5,)

-y=np.array([[1,2,3],[5,5,5]])

y.shape =>(4,3)

y.size => 12

x=np.array([1,5,2,3,7],dtype=int64)

print(x)

print('dtype:',x.dtype)

=> [1,2,3]

dtype: int64

x=np.arr.([1,2,3])

n.p.save('my_arr',x)

y=np.load('my_arr.npy')

print(y) =>[1 2 3]

x=np.zeros((3,4),dtype=int)

x=np.ones((4,5))

x=np.full((4,3),5)

=>[[5 5 5]

[5 5 5]

[5 5 5]

[5 5 5]]

الى صفرين

x=np.eye(5) 5x5

Diagonal = 1

x=np.diag([10,20,30,50])

x=np.arange(10)

=>[0,1 ——> 9]

x=np.arr([1,4,7])

1 4 7 10 13

-x=np.linspace(0,25,10)

[0. 2.77... 5.5 — 8.3 - 11.11

13.8 16.6 19.4 22.2 25.]

- x=np.linspace(0,25,10,endpoint=False)

[0 2.5 5 7.5 10 12.5 15 — endPoint

- x=np.arrang(20)

x=np.reshape(x,(4,5))

[[0 1 — 4

5 9]

- x=np.reshape(x,(10,2))

01

23

08:00
- Y = np.arange(20).reshape(10,2)
- X = n.p.linespace(0,25,10,endpoint=False).reshape(5,2)

09:00
0 5
10 15

10:00
- X = np.random.random((3,3))
- X = np.random.randint(4,15,(3,2))

11:00
[[8 5]
[12 14]
[2 2]]

12:00

- X = np.random.normal(0,0.1,size=(1000,1000))

01:00

pr X.mean()
X.std()

02:00
X.max()
X.min()

03:00
,posit (X>0).sum()
,neg (X<0).sum()

04:00

- X = np.array([1,2,...])

05:00
ازواد
X[0] = 1 X[-1] = 3

- X[3] = 20 هيبدل العوض

06:00
- X = np.arange(1,10).reshape(3,3)
- n.p.delete(Y [0], axis = 0)

07:00

W = np.insert(Y, 1, [4,5,6], axis 0
في اتات جديد

08:00

V = np.insert(Y, 1, 5, axis = 1)

- Z = [3 4
  5 6]

Z = np.vstack((X,Y))
[1 2
3 4
5 6]

w = np.hstack((Y, X.reshape(2,1)))
[3 4 1
5 6 2]   X = [1 2]

- X = np.arange(1,20).reshape(4,5)
Z = X[1:4 , 2:5] => 8 9 10 / 13 14 15 / 18 19 20
= X[1: , 2:] =>
Z = X[: , 2] all row in column 2
3 8 13 18
Z = X[:, 2:3] => [3 8 13 ...]

X = np.array([1,2,3,4,5])
Y = np.array([6,7,2,8,4])
=> np.intersect1d(X,Y)
= [2 4]
=> np.setdiff1d(X,Y)
= [1 3 5]
-> np.union1d(X,Y)
= 1 2 3 4 5 6 7 8

- X = np.random.randint(1,11, size(10))
[7 2 4 5 9 3 8 7 9 1]
np.sort(X) => ...
X =>

np.sort(np.unique(X))
1 2 3 4 5 7 8 9

np.random.randint(1,11, size(5,5))
- np.sort(X, axis=0)
- print(X+Y) = print(np.add(X,Y))