# تطبيق جميع التمارين الموجودة بالمحاضرة

from statistics import mean

import numpy as np

#from PIL import Image

Array=[80,85,90,95,100,105,110,115,120,125,77,22,1,5,21,83,254,717]

even=[]

odd=[]

y=int(input("Enter num:"))

s=1

for i in Array:

if i%2==0:

even+=[i]

else:

odd+=[i]

print(":زوجي\t",even)

print(odd,":فردي\t")

for i in Array:

if i ==y:

s=y

print("found") if s==y else print("not")

print("--------------------------------------------------")

print(max(Array))

print(min(Array))

print(mean(Array))

print("--------------------------------------------------")

point1=[1, 2, 3]

point2 = [4, 5, 6]

distance = sum([(i-j)\*\*2 for i,j in zip(point1, point2)])\*\*0.5

print ("distance:", distance)

distance = np.linalg.norm(np.array(point1) - np.array(point2))

print ("distance:", distance)

print("--------------------------------------------------")

data = np.array([[1, 2, np.nan], [4, np.nan, 6]])

print("The Data Pre-cleaning:\n",data)

col\_mean = np.nanmean(data, axis=0)

data[np.isnan(data)] =col\_mean[np.isnan(data).nonzero()[1]]

print ("The Data Post-cleaning:\n" , data)

print("--------------------------------------------------")

x=np.array([6, 2, 3, 4])

y=np.array([2, 4,5, 4])

coefficients=np.polyfit(x,y,1)

print ("Coefficients (Slope, Intercept):", coefficients)

# print("--------------------------------------------------")

import numpy as np

Calorie\_burnage [240, 250, 260, 270, 280, 290, 300, 310, 320, 330]

Average\_calorie\_burnage np.mean(Calorie\_burnage)

print(Average\_calorie\_burnage)

Print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

import numpy as np

بيانات ضغط الدم لـ ۱۰۰ مريض ضغط القباضي، انبساطي #

blood\_pressure np.random.randint(80, 200, (100, 2))

حساب المتوسط 1 #

mean\_bp np.mean(blood\_pressure, axis=0)

تصنيف المرض ارتفاع ضغط إذا كان الانقباضي > ١٤٠ 20 # high\_bp blood\_pressure [blood\_pressure[:0] > 140]

print("high\_bp:\n", high\_bp)

حساب الارتباط بين الضغط الانقباضي والانبساطي 3 #

correlation = np.corrcoef(blood\_pressure[:, 0], blood\_pressure[:,1])

print("correlation:\n", correlation)

# قم بزيارة موقع مكتبة NumPy

## NumPy v2.2 Manual NumPy documentation

قم بالاطلاع على المرجع pdf للمكتبة والمرفق مع المحاضرة وحاول تطبق على الأقل ٤٠ تمرين من تمارين المكتبة بشرط تكون تمارين متنوعة بين السهولة والمتوسطة والمتقدمة بشرط ان لا تكون من الأوامر والامثلة التي ذكرت في االمحاضرة

===== المستوى السهل (15 تمرين) =====

1. إنشاء مصفوفة بتوزيع بواسون

arr1 = np.random.poisson(lam=5, size=10)

2. حساب الوسط الهندسي

from scipy.stats import gmean

arr2 = gmean([1, 4, 16])

3. إيجاد الفرق بين مصفوفتين

arr3 = np.setdiff1d([1,3,5,7], [2,4,6])

4. تكرار المصفوفة

arr4 = np.tile([1,2,3], 3)

5. نسخ عميق للمصفوفة

arr5 = np.array([1,2,3])

arr5\_deep = arr5.copy()

6. حساب القيم الفريدة وتكراراتها

vals, counts = np.unique([1,1,2,3,3,3], return\_counts=True)

7. إنشاء مصفوفة قطرية

diag\_arr = np.diagflat([1,2,3])

8. توليد نقاط تشيبيشيف

chebyshev = np.polynomial.chebyshev.chebpts1(5)

9. تقسيم المصفوفة لأجزاء متساوية

split\_arr = np.array\_split(np.arange(10), 3)

10. حساب معاملات الانحدار

x = [1,2,3,4]

y = [3,5,7,9]

coeffs = np.polyfit(x, y, deg=1)

11. إنشاء شبكة إحداثيات

xgrid, ygrid = np.meshgrid(np.linspace(0,1,3), np.linspace(0,1,3))

12. حساب الفروقات المتتالية

diff\_arr = np.ediff1d([10, 15, 21, 30])

13. إنشاء مصفوفة بتوزيع طبيعي

arr13 = np.random.normal(loc=10, scale=2, size=5)

14. تعبئة المصفوفة (Padding)

arr14 = np.pad([1,2,3], (2,3), 'constant')

15. حساب الضرب الخارجي

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

===== المستوى المتوسط (15 تمرين) =====

16. تمثيل الأعداد بالثنائي

bin\_repr = np.binary\_repr(10, width=8)

17. تغيير حجم المصفوفة

arr17 = np.resize([1,2,3,4], (3,3))

18. إنشاء مصفوفة فانديرموند

vander = np.vander([1,2,3], 3)

19. تطبيق دالة على المصفوفة

def custom\_func(x,y):

return x\*y

arr19 = np.fromfunction(custom\_func, (3,3))

20. حساب التكامل

from scipy.integrate import quad

result, \_ = quad(lambda x: x\*\*2, 0, 4)

21. تحويل الإحداثيات الديكارتية لكروية

def cartesian\_to\_spherical(x, y, z):

r = np.sqrt(x\*\*2 + y\*\*2 + z\*\*2)

theta = np.arctan2(y, x)

phi = np.arctan2(np.sqrt(x\*\*2 + y\*\*2), z)

return r, theta, phi

r, theta, phi = cartesian\_to\_spherical(1,1,1)

22. حساب الفروقات مع إضافة قيم

diff2d = np.diff([[1,3],[5,9]], axis=0, prepend=0)

23. ضغط المصفوفة

arr23 = np.compress([True, False, True], [10,20,30])

24. التحليل القطاعي (SVD)

U, s, Vh = np.linalg.svd([[1,2],[3,4],[5,6]])

25. تحويل الأعداد لبتات

bits = np.unpackbits(np.array([10], dtype=np.uint8))

26. حساب المعكوس العام

inv\_matrix = np.linalg.pinv([[1,2],[3,4]])

# 27. إنشاء مصفوفة هيلبرت

from scipy.linalg import hilbert

hilb\_matrix = hilbert(3)

28. انحدار متعدد الحدود

x = [1,2,3,4]

y = [1,4,9,16]

poly\_coeffs = np.polyfit(x, y, 2)

29. توليد أرقام بتوزيعات مختلفة

exponential = np.random.exponential(scale=1.0, size=5)

gamma = np.random.gamma(shape=2, size=5)

logistic = np.random.logistic(loc=0, scale=1, size=5)

30. حساب التباين المشترك

cov\_matrix = np.cov(np.random.rand(3,10))

===== المستوى المتقدم (10 تمارين) =====

31. تحويل فورييه السريع

from numpy.fft import fft

spectrum = fft([1,2,1,2])

32. محاكاة حركة براونية

brownian = np.cumsum(np.random.normal(size=100))

33. حل معادلة تفاضلية

from scipy.integrate import solve\_ivp

def diff\_eq(t, y): return -0.5 \* y

sol = solve\_ivp(diff\_eq, [0, 10], [10], t\_eval=np.linspace(0,10,100))

34. دالة هافيسايد

heaviside = np.heaviside([-1.5, 0, 2.0], 0.5)

35. تجميع آينشتاين

einsum\_result = np.einsum('i,j->ij', [1,2,3], [4,5,6])

36. حساب مصفوفة جاكوبي

def func(x):

return np.array([x[0]\*x[1], x[0]\*\*2])

jac = np.gradient(func([1,2]))

37. تضمين متعدد الأبعاد

from sklearn.manifold import MDS

mds = MDS(n\_components=2)

embedded = mds.fit\_transform(np.random.rand(10,5))

38. تحويل فورييه متعدد الأبعاد

nd\_fft = np.fft.fftn(np.random.rand(3,3,3))

39. إنشاء مصفوفة كتلية

block = np.block([[np.eye(2), np.ones((2,2))], [np.zeros((2,2)), np.eye(2)]])

40. حساب التدرج متعدد الأبعاد

grad = np.gradient([[1,2,3],[4,5,6],[7,8,9]], axis=(0,1))

اكتب ملخص عن اهم ١٠ من الأوامر التي شدتك للانتباه من مكتبة NumPy من ٤٠ التمرين - بشرط ان لا تكون من الأوامر والامثلة التي ذكرت في المحاضرة.

1. np.emath.sqrt() # حساب الجذر التربيعي مع دعم الأعداد المركبة

2. np.setdiff1d() # إيجاد العناصر الموجودة في المصفوفة الأولى فقط

3. np.ediff1d() # حساب الفروقات بين العناصر المتتالية

4. np.binary\_repr() # تمثيل الأعداد الصحيحة في النظام الثنائي

5. np.unpackbits() # تحويل الأعداد إلى بتات مفردة

6. np.polynomial.chebyshev.chebpts1() # توليد نقاط تشيبيشيف من النوع الأول

7. np.heaviside() # دالة هافيسايد الخطوة (Step function)

8. np.gradient() # حساب التدرج لمصفوفات متعددة الأبعاد

9. np.block() # بناء مصفوفات كتلية من كتل متعددة

10. np.linalg.pinv() # حساب المعكوس العام للمصفوفة (للمصفوفات

١١. غير المربعة)

أمثله استخدام سريعة لكل أمر

# 1. دعم الجذور التربيعية للأعداد المركبة

np.emath.sqrt([-1, 4, -9])

# 2. إيجاد الفرق بين مصفوفتين

np.setdiff1d([1,3,5], [2,3,4])

# 3. حساب الفروقات المتتالية

np.ediff1d([10,15,21])

# 4. تحويل الأعداد لنظام ثنائي

np.binary\_repr(10, width=8)

# 5. تفكيك البايتات

np.unpackbits(np.array([10], dtype='uint8'))

# 6. توليد نقاط تشيبيشيف

np.polynomial.chebyshev.chebpts1(5)

# 7. دالة الخطوة

np.heaviside([-1,0,2], 0.5)

# 8. حساب التدرج

np.gradient([[1,2],[3,4]])

# 9. بناء مصفوفة كتلية

np.block([[np.eye(2), np.ones(2)]])

# 10. معكوس معمم للمصفوفة

np.linalg.pinv([[1,2],[3,4],[5,6]])