

```
In [1]: print("Welcome to Numpy-2-Cont")
```

Welcome to Numpy-2-Cont

Agenda

- **Usecase: calculate NPS** ✓
 - loading data: `np.loadtxt()` ✓
 - `np.empty()` ✓
 - `np.unique()` ✓
- **Introduction to use case** ✓
- **2-D arrays (Matrices)** ✓
 - `reshape()`
 - 2 Questions ✓
 - Transpose ✓
 - Converting Matrix back to Vector - `flatten()` ✓
- **Indexing and Slicing on 2D** ✓
 - Indexing ✓
 - Slicing
 - Masking (Fancy Indexing)
- **Universal Functions (ufunc) on 2D**
 - Aggregate Function/ Reduction functions - `sum()`, `mean()`, `min()`, `max()`
 - Axis argument
 - Logical Operations
 - Sorting function - `sort()`, `argsort()`
- **Use Case: Fitness Data analysis**
 - Loading data set and EDA using numpy
 - `np.argmax()`

```
In [2]: import numpy as np
```

```
In [3]: !gdown 1c0ClC8SrPwJq5rrkyMKyPn80nyHcFikK
```

Downloading...
From: <https://drive.google.com/uc?id=1c0ClC8SrPwJq5rrkyMKyPn80nyHcFikK>
To: /Users/nikhilsanghi/Downloads/01_dsml-course-main-live/batches/1_Aug_Beg_Mon/03_Numpy_2_cont/survey.txt
100%|██| 2.55k/2.55k [00:00<00:00, 2.86MB/s]

```
In [4]: score=np.loadtxt("/Users/nikhilsanghi/Downloads/01_dsml-course-main-live/batches/1_Aug_Beg_Mon/02_Numpy_2/survey.
```

```
In [5]: score
```

```
Out[5]: array([ 7, 10,  5, ...,  5,  9, 10])
```

```
In [6]: type(None)
```

```
Out[6]: NoneType
```

```
In [17]: arr=np.empty(shape=score.shape,dtype="<U20")  
arr
```

```
Out[17]: array(['', '', '', ..., '', '', ''], dtype='<U20')
```

```
In [ ]:
```

```
In [15]: # np.array(["scaler", "academy", "dsml"], dtype="<U5")
```

```
In [16]: # np.array(["scaler", "academy", "dsml", "dhsujdlhfdskfhdska\fhk"], dtype="U10")
```

```
In [19]: score<7
```

```
Out[19]: array([False, False,  True, ...,  True, False, False])
```

```
In [ ]: score[score<7]
```

```
In [20]: arr[score<7]="detractors"
```

```
In [21]: arr
```

```
Out[21]: array(['', '', 'detractors', ..., 'detractors', '', ''], dtype='<U20')
```

```
In [22]: score
```

```
Out[22]: array([ 7, 10,  5, ...,  5,  9, 10])
```

```
In [ ]:
```

```
In [24]: a=np.array([1,2,3,4,5])  
a
```

```
Out[24]: array([1, 2, 3, 4, 5])
```

```
In [25]: a<3
```

```
Out[25]: array([ True,  True, False, False, False])
```

```
In [27]: b=np.empty(shape=a.shape,dtype="i4")  
b
```

```
Out[27]: array([ 0,  0,  0,  0, 101], dtype=int32)
```

```
In [29]: a[a<3]=0
```

```
In [30]: a
```

```
Out[30]: array([0, 0, 3, 4, 5])
```

```
In [31]: b[a<3]=200
```

```
In [32]: b
```

```
Out[32]: array([200, 200,  0,  0, 101], dtype=int32)
```

```
In [ ]:
```

```
In [33]: arr[score>8]="promoters"
```

```
In [34]: arr[(score==7) | (score==8)]="passives"
```

```
In [35]: arr
```

```
Out[35]: array(['passives', 'promoters', 'detractors', ..., 'detractors',  
              'promoters', 'promoters'], dtype='<U20')
```

```
In [36]: score
```

```
Out[36]: array([ 7, 10,  5, ...,  5,  9, 10])
```

```
In [ ]:
```

```
In [38]: a=score[:10]
```

```
In [39]: a
```

```
Out[39]: array([ 7, 10,  5,  9,  9,  4,  7,  9,  9,  9])
```

```
In [40]: b=np.empty(shape=a.shape,dtype="U20")  
b
```

```
Out[40]: array(['', '', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], dtype='<U20')
```

```
In [41]: a>7
```

```
Out[41]: array([False,  True, False,  True,  True, False, False,  True,  True,  
              True])
```

```
In [43]: b[a<7]="detractors"
```

```
In [45]: b
```

```
Out[45]: array(['', ' ', 'detractors', ' ', ' ', 'detractors', ' ', ' ', ' ', ' '],  
              dtype='<U20')
```

```
In [46]: a
```

```
Out[46]: array([ 7, 10,  5,  9,  9,  4,  7,  9,  9,  9])
```

```
In [47]: b[a>8]="promoters"
```

```
In [48]: b
```

```
Out[48]: array(['', 'promoters', 'detractors', 'promoters', 'promoters',  
              'detractors', ' ', 'promoters', 'promoters', 'promoters'],  
              dtype='<U20')
```

```
In [49]: b[(a==7) | (a==8)]="passives"
```

```
In [50]: b
```

```
Out[50]: array(['passives', 'promoters', 'detractors', 'promoters', 'promoters',  
              'detractors', 'passives', 'promoters', 'promoters', 'promoters'],  
          dtype='<U20')
```

```
In [ ]:
```

```
In [51]: arr
```

```
Out[51]: array(['passives', 'promoters', 'detractors', ..., 'detractors',  
              'promoters', 'promoters'], dtype='<U20')
```

```
In [ ]:
```

```
In [52]: np.unique(arr)
```

```
Out[52]: array(['detractors', 'passives', 'promoters'], dtype='<U20')
```

```
In [53]: np.unique(score)
```

```
Out[53]: array([ 1,  4,  5,  7,  8,  9, 10])
```

```
In [57]: result=np.unique(arr,return_counts=True)  
result
```

```
Out[57]: (array(['detractors', 'passives', 'promoters'], dtype='<U20'),  
          array([332, 226, 609]))
```

```
In [56]: arr.unique()÷
```

```
File "/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_73279/2752609629.py", line 1  
arr.unique()÷  
          ^  
SyntaxError: invalid character '÷' (U+00F7)
```

```
In [65]: dictionary={}  
dictionary[result[0][0]]=result[1][0]  
dictionary[result[0][1]]=result[1][1]  
dictionary[result[0][2]]=result[1][2]
```

```
In [66]: result[1][0]
```

```
Out[66]: 332
```

```
In [67]: dictionary
```

```
Out[67]: {'detractors': 332, 'passives': 226, 'promoters': 609}
```

```
In [ ]:
```

```
In [68]: # 2d numpy arrays
```

```
In [69]: a=np.array([[1,2,3],[4,5,6],[7,8,9]])  
a
```

```
Out[69]: array([[1, 2, 3],  
               [4, 5, 6],  
               [7, 8, 9]])
```

```
In [71]: # np.array([[1,2,3],[4,5,6],[7,8,9,10]])
```

```
In [72]: a.ndim
```

```
Out[72]: 2
```

```
In [74]: a.shape
```

```
Out[74]: (3, 3)
```

```
In [75]: b=np.array([[1,2,3],[4,5,6]])  
b
```

```
Out[75]: array([[1, 2, 3],  
               [4, 5, 6]])
```

```
In [76]: b.shape
```

```
Out[76]: (2, 3)
```

```
In [77]: b.ndim
```

```
Out[77]: 2
```

```
In [78]: a
```

```
Out[78]: array([[1, 2, 3],  
               [4, 5, 6],  
               [7, 8, 9]])
```

```
In [79]: a.size
```

```
Out[79]: 9
```

```
In [80]: b.size
```

```
Out[80]: 6
```

```
In [81]: c=np.array([1,2,3,4])  
c
```

```
Out[81]: array([1, 2, 3, 4])
```

```
In [82]: c.size
```

```
Out[82]: 4
```

```
In [83]: a
```

```
Out[83]: array([[1, 2, 3],
               [4, 5, 6],
               [7, 8, 9]])
```

```
In [84]: a.size
```

```
Out[84]: 9
```

```
In [88]: a.shape
```

```
Out[88]: (3, 3)
```

```
In [85]: a.shape[0]
```

```
Out[85]: 3
```

```
In [86]: a.shape[1]
```

```
Out[86]: 3
```

```
In [87]: a.shape[0]*a.shape[1]
```

```
Out[87]: 9
```

```
In [89]: b
```

```
Out[89]: array([[1, 2, 3],
               [4, 5, 6]])
```

```
In [90]: b.shape
```

```
Out[90]: (2, 3)
```

```
In [91]: b.size
```

```
Out[91]: 6
```

```
In [92]: b.shape[0]
```

```
Out[92]: 2
```

```
In [93]: b.shape[1]
```

```
Out[93]: 3
```

```
In [94]: b.shape[0]*b.shape[1]
```

```
Out[94]: 6
```

```
In [ ]:
```

```
In [95]: d=np.array([[1,2,3,4]])  
e=np.array([1,2,3,4])
```

```
In [100]: d
```

```
Out[100]: array([[1, 2, 3, 4]])
```

```
In [96]: d.ndim
```

```
Out[96]: 2
```

```
In [97]: e.ndim
```

```
Out[97]: 1
```

```
In [98]: d.shape
```

```
Out[98]: (1, 4)
```

```
In [99]: e.shape
```

```
Out[99]: (4,)
```

```
In [ ]:
```

```
In [ ]:
```

```
In [102]: a=np.arange(1,13)  
a
```

```
Out[102]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

```
In [103]: a.shape
```

```
Out[103]: (12,)
```

```
In [ ]: # a.reshape(shape=())
```

```
In [ ]: # a.reshape(x,y,...)
```

```
In [ ]: # a.reshape((x,y,...)) # preferable
```

```
In [104]: a.reshape((1,12))
```

```
Out[104]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]])
```

```
In [105... a.reshape((12,1))
```

```
Out[105... array([[ 1],
        [ 2],
        [ 3],
        [ 4],
        [ 5],
        [ 6],
        [ 7],
        [ 8],
        [ 9],
        [10],
        [11],
        [12]])
```

```
In [111... a
```

```
Out[111... array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

```
In [106... a.reshape((2,6))
```

```
Out[106... array([[ 1,  2,  3,  4,  5,  6],
        [ 7,  8,  9, 10, 11, 12]])
```

```
In [107... a.reshape((6,2))
```

```
Out[107... array([[ 1,  2],
        [ 3,  4],
        [ 5,  6],
        [ 7,  8],
        [ 9, 10],
        [11, 12]])
```

```
In [110... a
```

```
Out[110... array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

```
In [108... a.reshape((3,4))
```

```
Out[108... array([[ 1,  2,  3,  4],
        [ 5,  6,  7,  8],
        [ 9, 10, 11, 12]])
```

```
In [109... a.reshape((4,3))
```

```
Out[109... array([[ 1,  2,  3],
        [ 4,  5,  6],
        [ 7,  8,  9],
        [10, 11, 12]])
```

```
In [112... a.reshape((5,4))
```

```
-----
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_73279/3816939626.py in <module>
----> 1 a.reshape((5,4))

ValueError: cannot reshape array of size 12 into shape (5,4)
```

```
In [114... a
```



```
Out[114...] array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

```
In [ ]: # a.reshape((-1,c))
# a.reshape((r,-1))

# a.reshape((-1,-1))
```

```
In [113...] a.reshape((-1,2))
```

```
Out[113...] array([[ 1,  2],
 [ 3,  4],
 [ 5,  6],
 [ 7,  8],
 [ 9, 10],
 [11, 12]])
```

```
In [115...] a.reshape((2,-1))
```

```
Out[115...] array([[ 1,  2,  3,  4,  5,  6],
 [ 7,  8,  9, 10, 11, 12]])
```

```
In [116...] a.reshape((-1,-1))
```

```
-----
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_73279/831918109.py in <module>
----> 1 a.reshape((-1,-1))

ValueError: can only specify one unknown dimension
```

```
In [117...] a.reshape((-1,5))
```

```
-----
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_73279/3939929223.py in <module>
----> 1 a.reshape((-1,5))

ValueError: cannot reshape array of size 12 into shape (5)
```

```
In [ ]:
```

```
In [119...] a=a.reshape(3,4)
a
```

```
Out[119...] array([[ 1,  2,  3,  4],
 [ 5,  6,  7,  8],
 [ 9, 10, 11, 12]])
```

```
In [121...] len(a)
```

```
Out[121...] 3
```

```
In [ ]:
```

```
In [120...] a
```

```
Out[120...] array([[ 1,  2,  3,  4],
 [ 5,  6,  7,  8],
 [ 9, 10, 11, 12]])
```

In [122... `a.T`

Out[122... `array([[1, 5, 9],
[2, 6, 10],
[3, 7, 11],
[4, 8, 12]])`

In [123... `a=np.arange(1,13)`
`a`

Out[123... `array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])`

In [124... `a.T`

Out[124... `array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])`

In [125... `a=np.arange(1,13).reshape((1,12))`
`a`

Out[125... `array([[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]])`

In [126... `a.T`

Out[126... `array([[1],
[2],
[3],
[4],
[5],
[6],
[7],
[8],
[9],
[10],
[11],
[12]])`

In [129... `a=a.reshape((3,4))`
`a`

Out[129... `array([[1, 2, 3, 4],
[5, 6, 7, 8],
[9, 10, 11, 12]])`

In [130... `a.reshape((12,))`

Out[130... `array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])`

In []:

In [131... `a.flatten()`

Out[131... `array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])`

In [132... `a.reshape((1,-1))`

Out[132... `array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])`

```
In [133... a.reshape((-1,1))
```

```
Out[133... array([[ 1],  
        [ 2],  
        [ 3],  
        [ 4],  
        [ 5],  
        [ 6],  
        [ 7],  
        [ 8],  
        [ 9],  
        [10],  
        [11],  
        [12]])
```

```
In [ ]:
```

```
In [134... a.reshape((a.size,))
```

```
Out[134... array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

```
In [ ]:
```

```
In [135... a
```

```
Out[135... array([[ 1,  2,  3,  4],  
        [ 5,  6,  7,  8],  
        [ 9, 10, 11, 12]])
```

```
In [137... a[1][1] # not recommended in DS
```

```
Out[137... 6
```

```
In [139... a
```

```
Out[139... array([[ 1,  2,  3,  4],  
        [ 5,  6,  7,  8],  
        [ 9, 10, 11, 12]])
```

```
In [138... a[1,1]
```

```
Out[138... 6
```

```
In [140... a[2,2]
```

```
Out[140... 11
```

```
In [141... a[-2,-1]
```

```
Out[141... 8
```

```
In [142... a[1,-1]
```

```
Out[142... 8
```

```
In [143... a
```

```
Out[143... array([[ 1,  2,  3,  4],
         [ 5,  6,  7,  8],
         [ 9, 10, 11, 12]])
```

```
In [ ]: # a[rows,columns]
```

```
In [144... a[[0,1,1],[1,2,3]] #recommended
```

```
Out[144... array([2, 7, 8])
```

```
In [145... a[(0,1,1),(1,2,3)] # not recommended
```

```
Out[145... array([2, 7, 8])
```

```
In [ ]: a[[0,1,1],[1,2,3]]
```

```
In [146... a
```

```
Out[146... array([[ 1,  2,  3,  4],
         [ 5,  6,  7,  8],
         [ 9, 10, 11, 12]])
```

```
In [147... a[[-3,-2,-2],[-3,-2,-1]]
```

```
Out[147... array([2, 7, 8])
```

```
In [148... a
```

```
Out[148... array([[ 1,  2,  3,  4],
         [ 5,  6,  7,  8],
         [ 9, 10, 11, 12]])
```

```
In [149... a=np.arange(1,13).reshape((3,4))
a
```

```
Out[149... array([[ 1,  2,  3,  4],
         [ 5,  6,  7,  8],
         [ 9, 10, 11, 12]])
```

```
In [151... a=a.flatten()
a
```

```
Out[151... array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

```
In [154... a=a.reshape((4,3))
```

```
In [155... a
```

```
Out[155... array([[ 1,  2,  3],
         [ 4,  5,  6],
         [ 7,  8,  9],
         [10, 11, 12]])
```

```
In [159... 
```

```
a[[0,1,2],[-2,-1,-1]]
```

Out[159]

array([2, 6, 9])

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js