Agenda

- **Indexing and Slicing on 1D** ✅
  - `Indexing` ✅
  - `Slicing` ✅
  - `Masking` (Fancy Indexing)✅
- **Operation on array**✅
  - `np.any` , `np.all` ✅
- **Universal Functions (ufunc) on 1D array**✅
  - Aggregate Function/ Reduction functions - `sum()` , `mean()` ,✅ `min()` , `max()` ✅
- **Usecase: calculate NPS**✅
  - loading data: `np.loadtxt()` ✅
- **2-D arrays (Matrices)**✅
  - `reshape()` ✅
  - Transpose
  - Converting Matrix back to Vector - `flatten()`
- **Introduction to use case**
- **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()`

Indexing and Slicing on 1D

`Indexing`

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

`Slicing`

In [13]:
```
a= np.arange(11,19)
a
```
Out[13]:
```
array([11, 12, 13, 14, 15, 16, 17, 18])
```
In [14]:
```
a[2:7:1]
```
Out[14]:
```
array([13, 14, 15, 16, 17])
```
In [15]:
```
a[2::1]
```
Out[15]:
```
array([13, 14, 15, 16, 17, 18])
```
In [16]:
```
a[-6:-1:1]
```
Out[16]:
```
array([13, 14, 15, 16, 17])
```
In [17]:
```
a[4:2:1]
```
Out[17]:
```
array([], dtype=int64)
```
In [18]:
```
a[4:2:-1]
```
Out[18]:
```
array([15, 14])
```
In [19]:
```
a[-4:-2:-1]
```
Out[19]:
```
array([], dtype=int64)
```
In [20]:
```
a[1:-2:2]
```
Out[20]:
```
array([12, 14, 16])
```
In [21]:
```
a[6:-7:-3]
```
Out[21]:
```
array([17, 14])
```
In [22]:
```
a[5::-2]
```
Out[22]:
```
array([16, 14, 12])
```
In [23]:
```
a[:-3:-3]
```
Out[23]:
```
array([18, 15])
```
In [24]:
```
a[::-1]
```
Out[24]:
```
array([18, 17, 16, 15, 14, 13, 12, 11])
```
In [ ]:

`Masking` (Fancy Indexing)

In [25]:
```
a
```
Out[25]:
```
array([11, 12, 13, 14, 15, 16, 17, 18])
```
In [26]:
```
a+10
```
Out[26]:
```
array([21, 22, 23, 24, 25, 26, 27, 28])
```
In [27]:
```
a-10
```
Out[27]:
```
array([1, 2, 3, 4, 5, 6, 7, 8])
```
In [28]:
```
a
```
Out[28]:
```
array([11, 12, 13, 14, 15, 16, 17, 18])
```
In [30]:
```
mask=a<15
mask
```
Out[30]:
```
array([ True,  True,  True,  True, False, False, False, False])
```
In [31]:
```
a[mask]
```
Out[31]:
```
array([11, 12, 13, 14])
```
In [32]:
```
a[a<15]
```
Out[32]:
```
array([11, 12, 13, 14])
```
In [33]:
```
a
```
Out[33]:
```
array([11, 12, 13, 14, 15, 16, 17, 18])
```
In [34]:
```
a[a%2==0]
```
Out[34]:
```
array([12, 14, 16, 18])
```
In [35]:
```
a[~(a%2==0)]
```
Out[35]:
```
array([11, 13, 15, 17])
```
In [41]:
```
mask1= a%2==0
mask2= a%3==0
mask1
```
Out[41]:
```
array([False,  True, False,  True, False,  True, False,  True])
```
In [42]:
```
mask2
```
Out[42]:
```
array([False, False, False,  True, False, False,  True])
```
In [43]:
```
a
```
Out[43]:
```
array([11, 12, 13, 14, 15, 16, 17, 18])
```
In [44]:
```
a[mask1 or mask2]
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_1461/480664235.py in <module>
----> 1 a[mask1 or mask2]

ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()
```
In [45]:
```
a[mask1 | mask2]
```
Out[45]:
```
array([12, 14, 15, 16, 18])
```
In [46]:
```
a[mask1 and mask2]
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_1461/1739789875.py in <module>
----> 1 a[mask1 and mask2]

ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()
```
In [47]:
```
a[mask1 & mask2]
```
Out[47]:
```
array([12, 18])
```
In [48]:
```
a[a%2==0 | a%3==0]
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_1461/2393799418.py in <module>
----> 1 a[a%2==0 | a%3==0]

ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()
```
In [49]:
```
a[(a%2==0) |(a%3==0)]
```
Out[49]:
```
array([12, 14, 15, 16, 18])
```
In [50]:
```
a[(a%2==0) &(a%3==0)]
```
Out[50]:
```
array([12, 18])
```

Operation on array

`np.any`, `np.all`

In [51]:
```
a=np.array([1,2,3,4,5])
b=np.array([6,7,8,9,10])
c=np.array([6,7,8,9,10,11,12])
```
In [52]:
```
a+10
```
Out[52]:
```
array([11, 12, 13, 14, 15])
```
In [53]:
```
a<10
```
Out[53]:
```
array([ True,  True,  True,  True,  True])
```
In [54]:
```
a+b
```
Out[54]:
```
array([ 7,  9, 11, 13, 15])
```
In [56]:
```
 # a+c # error
```
In [57]:
```
a
```
Out[57]:
```
array([1, 2, 3, 4, 5])
```
In [58]:
```
b
```
Out[58]:
```
array([ 6,  7,  8,  9, 10])
```
In [59]:
```
a
```
Out[59]:
```
array([1, 2, 3, 4, 5])
```
In [60]:
```
b
```
Out[60]:
```
array([ 6,  7,  8,  9, 10])
```
In [61]:
```
a<b
```
Out[61]:
```
array([ True,  True,  True,  True,  True])
```
In [62]:
```
a>b
```
Out[62]:
```
array([False, False, False, False, False])
```
In [63]:
```
a==b
```
Out[63]:
```
array([False, False, False, False, False])
```
In [64]:
```
np.all(a<b)
```
Out[64]:
```
True
```
In [65]:
```
a=np.array([1,2,3,4,11])
b=np.array([6,7,8,9,10])
```
In [66]:
```
a<b
```
Out[66]:
```
array([ True,  True,  True,  True, False])
```
In [67]:
```
np.all(a<b)
```
Out[67]:
```
False
```
In [68]:
```
np.any(a<b)
```
Out[68]:
```
True
```
In [69]:
```
a==b
```
Out[69]:
```
array([False, False, False, False, False])
```
In [70]:
```
np.any(a==b)
```
Out[70]:
```
False
```
In [ ]:

Ufunctions

In [71]:
```
a=np.arange(1,11)
a
```
Out[71]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```
In [72]:
```
# max
np.max(a)
```
Out[72]:
```
10
```
In [73]:
```
a.max()
```
Out[73]:
```
10
```
In [74]:
```
# min
np.min(a)
```
Out[74]:
```
1
```
In [75]:
```
a.min()
```
Out[75]:
```
1
```
In [76]:
```
#mean
np.mean(a)
```
Out[76]:
```
5.5
```
In [77]:
```
a.mean()
```
Out[77]:
```
5.5
```
In [78]:
```
#sum
np.sum(a)
```
Out[78]:
```
55
```
In [79]:
```
a.sum()
```
Out[79]:
```
55
```

NPS Biz case

In [80]:
```
data=np.loadtxt("/Users/nikhilsanghi/Downloads/01_dsml-course-main-live/batches/2_Sept_Beg_Tue_Oct_Beg_Tue/02_Numpy_2/survey.txt")
data
```
Out[80]:
```
array([ 7., 10.,  5., ...,  5.,  9., 10.])
```
In [81]:
```
len(data)
```
Out[81]:
```
1167
```
In [83]:
```
data.shape[0]
```
Out[83]:
```
1167
```
In [85]:
```
data.size
```
Out[85]:
```
1167
```
In [86]:
```
total=data.shape[0]
total
```
Out[86]:
```
1167
```
In [87]:
```
data<7
```
Out[87]:
```
array([False, False,  True, ...,  True, False, False])
```
In [90]:
```
detractors=data[data<7].shape[0]
detractors
```
Out[90]:
```
332
```
In [91]:
```
promotors = data[data>8].shape[0]
promotors
```
Out[91]:
```
609
```
In [93]:
```
passives=data[(data==7) | (data==8)].shape[0]
passives
```
Out[93]:
```
226
```
In [95]:
```
perc_promotors=promotors/total
perc_promotors
```
Out[95]:
```
0.5218508997429306
```
In [96]:
```
perc_detractors=detractors/total
perc_detractors
```
Out[96]:
```
0.28449014567266495
```
In [98]:
```
nps= (perc_promotors-perc_detractors)*100
nps
```
Out[98]:
```
23.73607540702657
```
In [ ]:

2D Numpy array

In [100]:
```
a=np.array([[1,2,3],[4,5,6]])
a
```
Out[100]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```
In [101]:
```
a.ndim
```
Out[101]:
```
2
```
In [102]:
```
a.shape
```
Out[102]:
```
(2, 3)
```
In [104]:
```
a.size
```
Out[104]:
```
6
```
In [105]:
```
len(a)
```
Out[105]:
```
2
```
In [106]:
```
a=np.arange(1,13)
a
```
Out[106]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```
In [107]:
```
a.ndim
```
Out[107]:
```
1
```
In [108]:
```
a.shape
```
Out[108]:
```
(12,)
```
In [109]:
```
a.reshape((3,4))
```
Out[109]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```
In [110]:
```
a.reshape((1,12))
```
Out[110]:
```
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 [112]:
```
a.reshape((12,1))
```
Out[112]:
```
array([[ 1],
       [ 2],
       [ 3],
       [ 4],
       [ 5],
       [ 6],
       [ 7],
       [ 8],
       [ 9],
       [10],
       [11],
       [12]])
```
In [113]:
```
a
```
Out[113]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```
In [114]:
```
a.reshape((2,6))
```
Out[114]:
```
array([[ 1,  2,  3,  4,  5,  6],
       [ 7,  8,  9, 10, 11, 12]])
```
In [116]:
```
a.reshape((6,2))
```
Out[116]:
```
array([[ 1,  2],
       [ 3,  4],
       [ 5,  6],
       [ 7,  8],
       [ 9, 10],
       [11, 12]])
```
In [117]:
```
a.reshape((3,4))
```
Out[117]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```
In [118]:
```
a.reshape((4,3))
```
Out[118]:
```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [10, 11, 12]])
```
In [119]:
```
a.reshape((5,2))
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_1461/699923399.py in <module>
----> 1 a.reshape((5,2))

ValueError: cannot reshape array of size 12 into shape (5,2)
```
In [120]:
```
a
```
Out[120]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```
In [121]:
```
a.reshape((-1,2))
```
Out[121]:
```
array([[ 1,  2],
       [ 3,  4],
       [ 5,  6],
       [ 7,  8],
       [ 9, 10],
       [11, 12]])
```
In [122]:
```
a.reshape((2,-1))
```
Out[122]:
```
array([[ 1,  2,  3,  4,  5,  6],
       [ 7,  8,  9, 10, 11, 12]])
```
In [123]:
```
a.reshape((-1,-1))
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_1461/831918109.py in <module>
----> 1 a.reshape((-1,-1))

ValueError: can only specify one unknown dimension
```
In [124]:
```
a.reshape((-1,))
```
Out[124]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```
In [125]:
```
a.reshape((-1,5))
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_1461/3939929223.py in <module>
----> 1 a.reshape((-1,5))

ValueError: cannot reshape array of size 12 into shape (5)
```
In [126]:
```
a.reshape((,6))
```
```
  File "/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_1461/183478571.py", line 1
    a.reshape((,6))
               ^
SyntaxError: invalid syntax
```
In [127]:
```
a.reshape((3,2,2))
```
Out[127]:
```
array([[[ 1,  2],
        [ 3,  4]],

       [[ 5,  6],
        [ 7,  8]],

       [[ 9, 10],
        [11, 12]]])
```
In [128]:
```
a.reshape((3,2,-1))
```
Out[128]:
```
array([[[ 1,  2],
        [ 3,  4]],

       [[ 5,  6],
        [ 7,  8]],

       [[ 9, 10],
        [11, 12]]])
```
In [130]:
```
a.reshape((3,-1,2))
```
Out[130]:
```
array([[[ 1,  2],
        [ 3,  4]],

       [[ 5,  6],
        [ 7,  8]],

       [[ 9, 10],
        [11, 12]]])
```
In [131]:
```
a.reshape((-1,2,2))
```
Out[131]:
```
array([[[ 1,  2],
        [ 3,  4]],

       [[ 5,  6],
        [ 7,  8]],

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