Day_28_211123

January 23, 2024

[5]:
```
import numpy as np
```

When we have a different dataTypes in a single array there is a priority followed

- string > Float > int > Boolean

[6]:
```
a = np.array([1,2,3,4,5,6,7,8,9,10,'a',2.55])
```

[7]:
```
a
```

[7]:
```
array(['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'a', '2.55'],
      dtype='<U32')
```

Assigning DataTypes when creating array

[8]:
```
a = np.array([[1,2,3,4],[2,3,5,4]],dtype='float')
a
```

[8]:
```
array([[1., 2., 3., 4.],
       [2., 3., 5., 4.]])
```

When ever we want help about any function we can use help

[9]:
```
#help(np.array([1,2,3]))
```

Changing the DataType after creating array

[10]:
```
b = a.astype('int')
b
```

[10]:
```
array([[1, 2, 3, 4],
       [2, 3, 5, 4]])
```

[11]:
```
arr = np.arange(1,40,0.5)
```

[12]:
```
arr
```

[12]:
```
array([ 1. ,  1.5,  2. ,  2.5,  3. ,  3.5,  4. ,  4.5,  5. ,  5.5,  6. ,
        6.5,  7. ,  7.5,  8. ,  8.5,  9. ,  9.5, 10. , 10.5, 11. , 11.5,
       12. , 12.5, 13. , 13.5, 14. , 14.5, 15. , 15.5, 16. , 16.5, 17. ,
       17.5, 18. , 18.5, 19. , 19.5, 20. , 20.5, 21. , 21.5, 22. , 22.5,
```

```
       23. , 23.5, 24. , 24.5, 25. , 25.5, 26. , 26.5, 27. , 27.5, 28. ,
       28.5, 29. , 29.5, 30. , 30.5, 31. , 31.5, 32. , 32.5, 33. , 33.5,
       34. , 34.5, 35. , 35.5, 36. , 36.5, 37. , 37.5, 38. , 38.5, 39. ,
       39.5])
```

Where will return the particular index value of the element based on condition

[13]:
```
np.where((arr>9) & (arr<39))
```

[13]:
```
(array([17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
        51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
        68, 69, 70, 71, 72, 73, 74, 75], dtype=int64),)
```

[14]:
```
np.where(arr>2)
```

[14]:
```
(array([ 3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
        20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
        37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
        54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70,
        71, 72, 73, 74, 75, 76, 77], dtype=int64),)
```

[15]:
```
a
```

[15]:
```
array([[1., 2., 3., 4.],
       [2., 3., 5., 4.]])
```

Using multiple condition in where

[16]:
```
np.where(a>3,a,a*100)
```

[16]:
```
array([[100., 200., 300.,   4.],
       [200., 300.,   5.,   4.]])
```

[17]:
```
own = np.array([2,5,4,6,-3,-7,5,-3,-9])
np.where(own<0,own*10,own/10)
```

[17]:
```
array([  0.2,   0.5,   0.4,   0.6, -30. , -70. ,   0.5, -30. , -90. ])
```

0.1 Airbnb is a company send us the data in 1D array

[18]:
```
!gdown 1c0ClC8SrPwJq5rrkyMKyPn80nyHcFikK
```

```
Downloading…
From: https://drive.google.com/uc?id=1c0ClC8SrPwJq5rrkyMKyPn80nyHcFikK
To: C:\Data\Data_science\Data Science RIA\3 Python\Codes\survey.txt

  0%|          | 0.00/2.55k [00:00<?, ?B/s]
100%|##########| 2.55k/2.55k [00:00<?, ?B/s]
```

[19]:
```
score = np.loadtxt("survey.txt", dtype="int")
```

[20]:
```
score
```

[20]:
```
array([ 7, 10,  5, …,  5,  9, 10])
```

[21]:
```
score.shape
```

[21]:
```
(1167,)
```

[22]:
```
score.ndim
```

[22]:
```
1
```

[23]:
```
score.size
```

[23]:
```
1167
```

[24]:
```
score.min()
```

[24]:
```
1
```

[25]:
```
score.max()
```

[25]:
```
10
```

[26]:
```
promoters = score[score>=9].shape[0]
```

[27]:
```
detractors = score[score<=6].shape[0]
```

[28]:
```
neutral = score[(score>6)&(score<9)].shape[0]
```

[29]:
```
nps = ( (promoters/(score.shape[0])) - (detractors/(score.shape[0])) ) * 100
```

[30]:
```
print(f"Net Promoter Score: {round(nps,2)}")
```

```
Net Promoter Score: 23.74
```

1 Creating an empty array with shape

[31]:
```
arr = np.empty(shape=score.shape,dtype='U20')
```

[32]:
```
arr.shape
```

[32]:
```
(1167,)
```

[33]:
```
arr # Here U1 indicates that Unicode<lenght of element in array>
```

[33]:
```
array(['', '', '', …, '', '', ''], dtype='<U20')
```

1.0.1 Converting Continous to Categorical data

[34]:
```
arr[score>=9] = "promoters"
arr[(score>=7) & (score<=8)] = "passive"
arr[score<=6] = "detractors"
```

[35]:
```
arr
```

[35]:
```
array(['passive', 'promoters', 'detractors', …, 'detractors',
       'promoters', 'promoters'], dtype='<U20')
```

[36]:
```
arr.shape
```

[36]:
```
(1167,)
```

[37]:
```
arr[arr=='promoters'].size
```

[37]:
```
609
```

[38]:
```
arr[arr=='passive'].size
```

[38]:
```
226
```

[39]:
```
arr[arr=='detractors'].size
```

[39]:
```
332
```

What if there are more no of elements we use unique function to get unique elements

[40]:
```
unique, count = np.unique(arr,return_counts='True')
```

[41]:
```
count
```

[41]:
```
array([332, 226, 609], dtype=int64)
```

[42]:
```
pod = count[0] / count.sum() * 100
```

[43]:
```
pop = count[2] / count.sum() * 100
```

[44]:
```
nps = pop - pod
nps
```

[44]:
```
23.73607540702657
```