योग: कर्मसु कौशलम्

Darshan
UNIVERSITY

Data Mining

Lab - 2

Name: Harmik Rathod

Enrollment No: 24010101680

Numpy & Perform Data Exploration with Pandas

Numpy

1. NumPy (Numerical Python) is a powerful open-source library in Python used for numerical and scientific computing.
2. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on them efficiently.
3. NumPy is highly optimized and written in C, making it much faster than using regular Python lists for numerical operations.
4. It serves as the foundation for many other Python libraries in data science and machine learning, like pandas, TensorFlow, and scikit-learn.
5. With features like broadcasting, vectorization, and integration with C/C++ code, NumPy allows for cleaner and faster code in numerical computations.

Step 1. Import the Numpy library

In [11]:

```
import numpy as np
```

Step 2. Create a 1D array of numbers

In [41]:

```
arr=np.array([10,20,30,40,50])
arr
```

```
Out[41]: array([10, 20, 30, 40, 50])
```

```
In [15]: arr1=np.arange(11)
         arr1
```

```
Out[15]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
In [17]: arr2=np.arange(1,11)
         arr2
```

```
Out[17]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

Step 3. Reshape 1D to 2D Array

```
In [24]: arr3=np.arange(12).reshape(3,4)
         arr3
```

```
Out[24]: 2
```

```
In [26]: arr3.dtype #datatype
```

```
Out[26]: dtype('int32')
```

```
In [28]: arr3.size  #size of array
```

```
Out[28]: 12
```

```
In [29]: arr3.ndim #dimesional of array
```

```
Out[29]: 2
```

Step 4. Create a Linspace array

```
In [31]: # np.linspace(start,end,noofele)
         arr4=np.linspace(0,5,10)
         arr4
```

```
Out[31]: array([0.        , 0.55555556, 1.11111111, 1.66666667, 2.22222222,
                2.77777778, 3.33333333, 3.88888889, 4.44444444, 5.        ])
```

Step 5. Create a Random Numbered Array

```
In [38]: arr5=np.random.rand(10)
         arr5
```

```
Out[38]: array([0.79396546, 0.36140335, 0.86861998, 0.63020115, 0.73472398,
                0.58807095, 0.6907898 , 0.46554542, 0.43849539, 0.46143955])
```

```
In [ ]: 
```

Step 6. Create a Random Integer Array

In [44]:

```
arr6=np.random.randint(1,10,size=10)
arr6
```

Out[44]:

```
array([5, 2, 5, 1, 6, 9, 1, 2, 7, 1])
```

In [ ]:

Step 7. Create a 1D Array and get Max,Min,ArgMax,ArgMin

In [47]:

```
arr7=np.random.randint(1,20, size=10)
arr7
```

Out[47]:

```
array([19,  2, 14,  3, 15, 13,  2, 11,  5, 12])
```

In [48]:

```
arr7.min() #value return
```

Out[48]:

```
2
```

In [49]:

```
arr7.max() #value return
```

Out[49]:

```
19
```

In [51]:

```
arr7.argmax() #index return
```

Out[51]:

```
0
```

In [52]:

```
arr7.argmin() #index return
```

Out[52]:

```
1
```

Step 8. Indexing in 1D Array

In [55]:

```
arr8=np.array([1,2,3,4,5,6,7,8])
```

In [56]:

```
arr8[2]
```

Out[56]:

```
3
```

Step 9. Indexing in 2D Array

In [59]:

```
arr9=np.arange(1,13).reshape(3,4)
arr9
```

Out[59]:

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

In [61]:

```
arr9[0][0]
```

Out[61]:

```
1
```

In [62]:

```
arr9[-1][-1]
```

Out[62]:

```
12
```

In [65]:

```
arr9[1][3]
```

Out[65]:

```
8
```

Step 10. Conditional Selection

In [67]:

```
arr10=np.arange(1,17).reshape(4,4)
arr10
```

Out[67]:

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [13, 14, 15, 16]])
```

In [69]:

```
arr10[arr10>10]
```

Out[69]:

```
array([11, 12, 13, 14, 15, 16])
```

In [73]:

```
arr10[arr10<10]
```

Out[73]:

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

🔥 You did it! 10 exercises down — you're on fire! 🔥

Pandas

Step 1. Import the necessary libraries

In [76]:

```
import pandas as pd
```

Step 2. Import the dataset from this address.

In [123…

```
users=pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u
```

Step 3. Assign it to a variable called users and use the 'user_id' as index

In [81]:

```
users=pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u
users
```

Out[81]:

	age	gender	occupation	zip_code
user_id				
1	24	M	technician	85711
2	53	F	other	94043
3	23	M	writer	32067
4	24	M	technician	43537
5	33	F	other	15213
...	...	...	...	...
939	26	F	student	33319
940	32	M	administrator	02215
941	20	M	student	97229
942	48	F	librarian	78209
943	22	M	student	77841

943 rows × 4 columns

Step 4. See the first 25 entries

In [91]:

```
users.head(25)
```

Out[91]:

	age	gender	occupation	zip_code
user_id				
1	24	M	technician	85711
2	53	F	other	94043
3	23	M	writer	32067
4	24	M	technician	43537
5	33	F	other	15213
6	42	M	executive	98101
7	57	M	administrator	91344
8	36	M	administrator	05201
9	29	M	student	01002
10	53	M	lawyer	90703
11	39	F	other	30329
12	28	F	other	06405
13	47	M	educator	29206
14	45	M	scientist	55106
15	49	F	educator	97301
16	21	M	entertainment	10309
17	30	M	programmer	06355
18	35	F	other	37212
19	40	M	librarian	02138
20	42	F	homemaker	95660
21	26	M	writer	30068
22	25	M	writer	40206
23	30	F	artist	48197
24	21	F	artist	94533
25	39	M	engineer	55107

Step 5. See the last 10 entries

In [92]:

```
users.tail(10)
```

Out[92]:

user_id	age	gender	occupation	zip_code
934	61	M	engineer	22902
935	42	M	doctor	66221
936	24	M	other	32789
937	48	M	educator	98072
938	38	F	technician	55038
939	26	F	student	33319
940	32	M	administrator	02215
941	20	M	student	97229
942	48	F	librarian	78209
943	22	M	student	77841

Step 6. What is the number of observations in the dataset?

In [87]:

```
users.shape[0]
```

Out[87]:

```
943
```

Step 7. What is the number of columns in the dataset?

In [126…

```
users.shape[1]
```

Out[126…

```
1
```

Step 8. Print the name of all the columns.

In [88]:

```
users.columns
```

Out[88]:

```
Index(['age', 'gender', 'occupation', 'zip_code'], dtype='object')
```

Step 9. How is the dataset indexed?

In [90]:

```
users.index
```

Out[90]:

```
Index([  1,   2,   3,   4,   5,   6,   7,   8,   9,  10,
       ...
       934, 935, 936, 937, 938, 939, 940, 941, 942, 943],
      dtype='int64', name='user_id', length=943)
```

Step 10. What is the data type of each column?

```
In [94]: users.dtypes
```

```
Out[94]: age            int64
         gender        object
         occupation    object
         zip_code      object
         dtype: object
```

Step 11. Print only the occupation column

```
In [121… users.occupation
```

```
Out[121… user_id
         1         technician
         2              other
         3             writer
         4         technician
         5              other
                    ...
         939          student
         940    administrator
         941          student
         942        librarian
         943          student
         Name: occupation, Length: 943, dtype: object
```

Step 12. How many different occupations are in this dataset?

```
In [122… users.occupation.nunique()
```

```
Out[122… 21
```

Step 13. What is the most frequent occupation?

```
In [102… users.occupation.value_counts().head(1)
```

```
Out[102… occupation
         student    196
         Name: count, dtype: int64
```

Step 14. Summarize the DataFrame.

```
In [104… users.describe()
```

Out[104…

	age
count	943.000000
mean	34.051962
std	12.192740
min	7.000000
25%	25.000000
50%	31.000000
75%	43.000000
max	73.000000

Step 15. Summarize all the columns

In [106…

```
users.describe(include='all')
```

Out[106…

	age	gender	occupation	zip_code
count	943.000000	943	943	943
unique	NaN	2	21	795
top	NaN	M	student	55414
freq	NaN	670	196	9
mean	34.051962	NaN	NaN	NaN
std	12.192740	NaN	NaN	NaN
min	7.000000	NaN	NaN	NaN
25%	25.000000	NaN	NaN	NaN
50%	31.000000	NaN	NaN	NaN
75%	43.000000	NaN	NaN	NaN
max	73.000000	NaN	NaN	NaN

Step 16. Summarize only the occupation column

In [113…

```
users.age.describe()
```

Out[113…

```
count    943.000000
mean      34.051962
std       12.192740
min        7.000000
25%       25.000000
50%       31.000000
75%       43.000000
max       73.000000
Name: age, dtype: float64
```

Step 17. What is the mean age of users?

In [112…

```
users.age.mean()
```

Out[112…

```
34.05196182396607
```

Step 18. What is the age with least occurrence?

In [120…

```
users.age.value_counts().tail()
```

Out[120…

```
age
7     1
66    1
11    1
10    1
73    1
Name: count, dtype: int64
```

You're not just learning, you're mastering it. Keep aiming higher! 🚀

In [ ]:

```

```