NumPy

1. Arrays

- Create a 2D array (matrix):

```
np.array([[1,2],[3,4]]) # 2x2 matrix
np.array(np.arange(16 )).reshape((4,4))
```

- Evenly spaced numbers over a specified interval(Useful for plotting):

```
np.linspace(0, 10, 11)  # array([0., 1., ..., 10.])
np.arange(0, 10)        # Same as linspace without fractional
intervals
```

Array Properties

Command	Description
`x.ndim`	Number of Dimensions
`x.dtype`	Data type of an array
`x.T`	Transpose of an array
`x.shape`	Gives the shape of an array

Array Operations

- Checks if array elements meets a condition : `np.all(x>0)` , `np.all(x>2,where [True, True,False])` --> Checking only the values that are masked as `True` if the condition apply to them
  example:

```
data = np.array([1,14,10,20,40])
valid_mask = np.array([False,True,False,True,True])
```

```
np.all(data>10,where=valid_mask) # Returns True  if all elemets
that mask apply to them >10
```

- Square root : `np.sqrt(x)`
- Square every element : `x**2`
- Mean/variance/standard deviation : `np.mean(x)` , `np.var(x)` , `np.std(x)`
- correlation matrix between x,y : `np.corrcoef(x,y)`
- Return only the unique elements on the array : `np.unique(x)`

Random Numbers

- Normal distribution: (Sampling from $N(\mu, \sigma)$)

```
np.random.normal(loc=0,scale=1,size=100) # 100 sample from N(0,1)
```

- Reproduce Randomness

```
rng =np.random.default_rng(seed=12) #always same  random numbers
rng.normal(0,1,10) #same sample each time
```

Indexing & Slicing

- **2D array** :
  `A[1,2]` --> Second row Third column
  `A[[1,2]]` --> all of Second row and Third row
  `A[:[0,2]]` --> First column and Third column
- **submatrices**:

```
A[[1,3],[0,2]] # will only select the elements at [1,2] [0,2]
np.array([[1,2]][:,[0,2]]) #sub-setting work around
# array([[4, 6],
#       [12,14]])
---------------
idx = np.ix_([1,3],[0,2])
A[idx] # more efficient
---------------
A[1:4:2,0:2:1] #[start:stop:step] stop-value not included
```

```
# Selecting rows starting from 1th row till 4th with two steps
# Selecting Column starting from 0th column till 2th with one step
```

- Boolean Indexing:
  Numpy treat Boolean's and integers differently

```
boolean_rows= np.zeros(A_matrix.shape[0],bool) # 0 th row of
A_matrix
boolean_rows[[1,2]]= True
boolean_cols = np.zeros(A_matrix.shape[:1],bool)
boolean_cols[[0,1,3]] = True
A1=A_matrix[np.array([0,1,1,0])]
A2=A_matrix[boolean_rows]
np.all(A2==A1) # return False
idx_bool =np.ix_(boolean_rows,boolean_cols)
A_matrix[idx_bool]
```

- The sub-matrix generated by the integers array is a mesh : the first ,second,first ,second rows of `A_matrix`
- The sub-matrix generated by the boolean array is only the first and second rows of `A_matrix`
- Its also another way to get the sub-matrix from the `A_matrix`

---

tags:
#machine-learning #numpy #islp

created: 2025-05-22