Lab Exercise 2: NumPy package

Note:

* Prepare a PDF document and name the file as “Lab2_RegisterNo.pdf”.
* PDF file should consist Question No, Code, and Result for each Question.
* File Should be headed with your Register number, Slot number, Lab Exercise number.

* * *

1. Create a NumPy Boolean array of 3 * 3 with True values.

2. Extract Odd numbers from any array.

3. Replace all Odd numbers with -1 without affecting original array.

4. Reshape the Given sequential array in to ‘n’ rows.

5. Combine two arrays vertically and horizontally.

6. Common items between two arrays.

7. From array ‘a’ remove all items present in array ‘b’

8. Print the matching positions of two element arrays.

9. Extract all numbers between range of indices from the given array.

10. Convert a Scalar function to work on NumPy arrays.

11. Swap two columns/rows of a 2D NumPy array.

12. Reverse the rows/columns of a 2D NumPy array.

13. Create a 2D array of shape 5x3 to contain random decimal numbers between 5 and 10. Print only three decimal places for a floating-point number.

14. Suppress the scientific notation in printing floating point number.

15. Print the full numpy array ‘a’ without truncating.

16. Create a 4x2 integer array and Prints its attributes: shape, dimensions, length of each element of the array.

17. Create a 5X2 integer array from a range between 100 to 200 such that the difference between each element is 10.

18. Return array of odd rows and even columns from below NumPy array.

19. Split the array into four equal-sized sub-arrays.

20. Sort following NumPy array: 1: Sort array by the second row 2: Sort the array by the second column

21. Print max from axis 0 and min from axis 1 from the 2-D array.

22. Delete the second column from a given array and insert the new column in its place.

23. Find the positions of:

- elements in x where its value is more than its corresponding element in y, and

- elements in x where its value is equals to its corresponding element in y.

24. Write a program to multiply two matrices of size (100,100) using for loops and also with NumPy methods. Compare the time of execution of both cases.

25. Write a program to execute the steps below using NumPy:

$$\boldsymbol{z_{ij} = \sum_{k=1}^{n} w_{ik}x_{kj} \ and \ \sigma_{ij}(z_{ij}) = \frac{1}{1+e^{-z_{ij}}}}$$

Where, $\boldsymbol{w \ and \ x}$ are the matrices of random numbers having dimensions $(\boldsymbol{m}, \boldsymbol{n}) \ \boldsymbol{and} \ (\boldsymbol{n}, \boldsymbol{k})$, respectively. $\boldsymbol{\sigma}(\boldsymbol{z})$ is a function which performs above defined operation on elements of $\boldsymbol{z}$.

26. Create two vectors $\boldsymbol{y \ and \ \hat{y}}$ having same dimensions, where $\hat{y}$ should consist of random numbers between [0,1] and y should contain $\boldsymbol{0s \ and \ 1s}$, for example $\boldsymbol{y} =$ $[\boldsymbol{0}, \boldsymbol{1}, \boldsymbol{1}, \boldsymbol{0}, \boldsymbol{10}, \boldsymbol{0}, \boldsymbol{1}, \dots, \boldsymbol{1}]$. Compute the given expression:

$$\boldsymbol{O = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log_2(\hat{y}_i) + (1 - y_i)\log_2(1 - \hat{y}_i)]}$$

where n is the total number of elements in $y \ and \ \hat{y}$.