**NumPy Assignment Questions on Placement Dataset**

The dataset placement\_multi\_linear\_reg.csv contains three columns: cgpa (students' CGPA), package (salary package in LPA), and placement (0 for not placed, 1 for placed). Use the NumPy library to solve the following questions. Assume the dataset is loaded into a NumPy array for analysis.

**Question 1: Basic Array Operations**

* **Task**: Load the dataset into a NumPy array, excluding the header. Extract the cgpa and package columns into separate arrays.
* **Steps**:
  1. Use np.genfromtxt or np.loadtxt to load the dataset, skipping the header row.
  2. Create two separate arrays: one for cgpa and one for package.
  3. Compute the mean and standard deviation of both cgpa and package arrays.
* **Output**: Print the shape of the full dataset array, the first 5 elements of the cgpa and package arrays, and their mean and standard deviation.

**Question 2: Filtering and Counting**

* **Task**: Identify students with a CGPA above 7.0 who were placed (placement = 1).
* **Steps**:
  1. Create a boolean mask to filter students with cgpa > 7.0 and placement == 1.
  2. Use the mask to extract the corresponding rows from the dataset.
  3. Count the number of students meeting these criteria.
* **Output**: Print the number of students and the first 3 rows of the filtered data (if any).

**Question 3: Data Normalization**

* **Task**: Normalize the package column to a range of [0, 1] using min-max normalization.
* **Steps**:
  1. Extract the package column into a NumPy array.
  2. Apply the min-max normalization formula: (x - min(x)) / (max(x) - min(x)).
  3. Verify that the normalized values lie between 0 and 1.
* **Output**: Print the first 5 normalized package values and the minimum and maximum of the normalized array.

**Question 4: Correlation Analysis**

* **Task**: Compute the correlation coefficient between cgpa and package to explore their relationship.
* **Steps**:
  1. Extract the cgpa and package columns as arrays.
  2. Use np.corrcoef to calculate the Pearson correlation coefficient between cgpa and package.
* **Output**: Print the correlation coefficient matrix and interpret the result in one sentence.

**Question 5: Grouping and Aggregation**

* **Task**: Calculate the average package for students who were placed versus those who were not.
* **Steps**:
  1. Create two boolean masks: one for placement == 1 and one for placement == 0.
  2. Use these masks to extract the package values for each group.
  3. Compute the mean package for each group using np.mean.
* **Output**: Print the average package for placed and not placed students, rounded to 2 decimal places.

**Question 6: Array Manipulation**

* **Task**: Create a new array that combines cgpa and package for students with cgpa > 6.5 and sort it by package in descending order.
* **Steps**:
  1. Filter the dataset to include only rows where cgpa > 6.5.
  2. Extract the cgpa and package columns for these rows into a new 2D array.
  3. Sort the array by the package column in descending order using np.argsort.
* **Output**: Print the first 5 rows of the sorted array.

**Question 7: Binning CGPA**

* **Task**: Bin the cgpa values into three categories: Low (CGPA < 6.0), Medium (6.0 ≤ CGPA < 7.5), and High (CGPA ≥ 7.5).
* **Steps**:
  1. Use np.digitize to assign each cgpa value to one of the three bins.
  2. Count the number of students in each bin using np.bincount or np.unique with return\_counts=True.
* **Output**: Print the number of students in each CGPA category.

**Question 8: Missing Value Handling**

* **Task**: Simulate missing values in the package column by randomly setting 10% of the values to np.nan, then replace them with the median package value.
* **Steps**:
  1. Create a copy of the package column array.
  2. Randomly select 10% of the indices using np.random.choice and set those values to np.nan.
  3. Compute the median of the non-missing package values using np.nanmedian.
  4. Replace the np.nan values with the median.
* **Output**: Print the number of missing values introduced and the first 5 values of the modified package array.

**Bonus Question: Advanced Array Operations**

* **Task**: For students who were placed, compute the Euclidean distance of their (cgpa, package) pairs from the mean (cgpa, package) of placed students.
* **Steps**:
  1. Filter the dataset for placement == 1 and extract cgpa and package columns.
  2. Compute the mean cgpa and mean package for these students.
  3. Calculate the Euclidean distance for each student’s (cgpa, package) pair from the mean pair using np.linalg.norm.
  4. Identify the student with the maximum distance.
* **Output**: Print the maximum distance and the corresponding student’s cgpa and package values.

**Notes**

* Use numpy for all computations and avoid using pandas or other libraries.
* Ensure proper handling of edge cases (e.g., empty arrays, invalid values).
* For loading the dataset, you can assume the file is accessible and use code like:
* import numpy as np

data = np.genfromtxt('placement\_multi\_linear\_reg.csv', delimiter=',', skip\_header=1)

**Ohidul Alam Nannu**

KUET BME’20

RoyalxLLC