

Artificial Intelligence Lab, B.Tech 4th Semester

Instructions

1. This is only for practice. No need to submit it.
2. Complete it by 12:00 PM today. Your completion will be reviewed by the Teaching Assistants.

Practice Assignment 3

1. (a) Create a text file named "data.txt" and write the following data into it:
1, John, 25
2, Jane, 30
3, Bob, 22
4, Alice, 28

(b) Write a Python function `read_data` that reads the data from "data.txt" and returns a list of dictionaries, where each dictionary represents a record with keys 'ID', 'Name', and 'Age'.

(c) Write another function `write_data` that takes the file path and a list of dictionaries (similar to the output of the previous function) and writes the data into the file. Ensure that the file is overwritten with the new data.

(d) Implement a function `update_age(file_path, id, new_age)` that updates the age of the person with the given ID in the "data.txt" file.
2. Implement a function called `process_data(input_filename, output_filename)` that reads a list of numbers from the file specified by `input_filename`, squares each number, and writes the squared numbers to the file specified by `output_filename`.
3. Install NumPy if not already installed (`pip install numpy`).

(a) Consider the following data stored in a file named `data1.txt` and Implement the following tasks in your Python program:
1,2,3,4,5
6,7,8,9,10
11,12,13,14,15

i. Read the data from `data1.txt` and store it in a NumPy array.
ii. Calculate the mean and standard deviation for each set of numbers in the array.

- iii. Create a new NumPy array that contains the mean and standard deviation for each set of numbers. Each row should represent one set of numbers, and the columns should be labeled appropriately.
 - iv. Write the new array to a new text file named "results.txt". Each line in the file should correspond to one set of numbers, and the values should be separated by commas.
Test your program with the provided "data.txt" file and ensure that the "results.txt" file is generated correctly.
- (b) Write a function called `calculate_statistics(data)` that takes a NumPy array as input and returns a dictionary containing the following statistics: Median, Minimum, Maximum
- (c) Write a function called `multiply_matrices(matrix1, matrix2)` that takes two NumPy matrices as input and returns their product.
- (d) Create a NumPy array `arr` with 10 random integers between 1 and 100 (inclusive). Write a function called `filter_odd_numbers(arr)` that takes this array as input and returns a new array containing only the odd numbers from the original array.
- (e) Create a NumPy array with 20 integers.
 - i. Use array slicing to extract the first half and the second half of the array.
 - ii. Modify specific elements in the array using indexing.
- (f) Generate two random NumPy arrays of any dimension with the same shape. Display both arrays.
 - i. Concatenate the two arrays horizontally and vertically. Display the results of both concatenations.
 - ii. Sort both arrays along a specified axis (choose any axis). Display the sorted arrays.
 - iii. Perform element-wise addition, subtraction, and multiplication on the two arrays. Display the results of each operation.
- (g)
 - i. Download the "iris.csv" dataset from this link (<https://archive.ics.uci.edu/dataset/53/iris>) or provide a similar dataset in CSV format.
 - ii. Write a Python script to load the dataset into a Pandas DataFrame.
 - iii. Display the first 5 rows of the dataset using the `head()` method.
[Hint. Syntax of head method is `dataframe.head(n)` where `n` is optional. Default value of `n` is 5]
 - iv. Calculate and print the following statistics for each numerical column in the dataset: mean, median, minimum, maximum.
[Hint: You can use describe function to compute the basic statistical details]