NumPy and Pandas Aggregations Assignment

Mustafa AbdulRazek

8 Sept 2024

Task 1: Using NumPy Aggregation Functions

Objective: Learn how to use basic NumPy aggregation functions like `sum`, `min`, `max`, and `mean`.

Task Description

Aggregation functions are essential for quickly analyzing data within arrays. In this task, you will practice using various aggregation functions to compute sums, minima, maxima, and means.

Task

Write a Python script to:

- Create a NumPy array of 10 random floating-point numbers between 0 and 1.
- Compute and print the sum, minimum, maximum, and mean of the array using NumPy functions.
- Compare the performance of Python's built-in `sum()` function with NumPy's `np.sum()` using the `%timeit` command.

Example Output

```
Array: [0.1, 0.5, 0.3, ...]
Sum: 5.6
Min: 0.1
Max: 0.9
Mean: 0.56
```

Task 2: Aggregating Multidimensional Arrays

Objective: Understand how to perform aggregations along different axes in multidimensional arrays.

Task Description

Aggregation functions can operate along rows or columns of a 2D array. This task will help you learn how to specify axes for aggregation.

Task

Write a Python script to:

- Create a 3x4 NumPy array with random values between 0 and 1.
- Compute the sum of all elements in the array.
- Find the minimum value in each column (`axis=0`).
- Find the maximum value in each row (`axis=1`).

Example Output

```
Array:
[[0.8, 0.2, 0.5, 0.1],
 [0.3, 0.9, 0.4, 0.6],
 [0.7, 0.1, 0.3, 0.8]]

Sum of all elements: 5.7
Min in each column: [0.3, 0.1, 0.3, 0.1]
Max in each row: [0.8, 0.9, 0.8]
```

Task 3: Basic Pandas Series Operations

Objective: Learn how to create and manipulate a Pandas Series.

Task Description

The Pandas Series is similar to a one-dimensional array but offers additional capabilities, such as labeled indices. This task introduces you to creating and using Series.

Task

Write a Python script to:

- Create a Pandas Series from a list of numbers `[10, 20, 30, 40]`.
- Set custom indices `['a', 'b', 'c', 'd']` for the Series.
- Access and print the element at index `'b'`.
- Modify the element at index `'c'` to 35.

Example Output

```
Series:
a    10
b    20
c    30
d    40

Element at 'b': 20
Modified Series:
a    10
b    20
c    35
d    40
```

Task 4: Creating and Exploring Pandas DataFrame

Objective: Learn how to create a DataFrame and explore its structure.

Task Description

DataFrames are two-dimensional structures in Pandas that store data in rows and columns. This task will help you understand how to create and explore a DataFrame.

Task

Write a Python script to:

- Create a DataFrame using a dictionary with keys `['Name', 'Age', 'Score']` and corresponding lists of values.
- Print the DataFrame.
- Access and print the 'Age' column.
- Print the row with index 1.

Example Output

```
DataFrame:
   Name  Age  Score
0  John   25     85
1  Jane   22     90
2  Mike   28     88

'Ages' column:
0    25
1    22
2    28

Row at index 1:
Name     Jane
Age        22
Score      90
```

Task 5: Basic DataFrame Aggregations

Objective: Practice performing basic aggregation operations on DataFrames.

Task Description

Aggregations such as sum, mean, and max can also be performed on DataFrames, making them powerful tools for data analysis.

Task

Write a Python script to:

- Create a DataFrame with random integer values, containing 5 rows and 3 columns.
- Compute the sum of each column.
- Compute the mean of each row.
- Find the maximum value in the DataFrame.

Example Output

```
DataFrame:
   0  1  2
0  5  3  2
1  6  4  8
2  7  5  9
3  2  1  3
4  8  6  4

Column sums: [28, 19, 26]
Row means: [3.3, 6.0, 7.0, 2.0, 6.0]
Max value: 9
```