Coding Blocks | Online

Machine Learning Online

Assignment – 3 Numpy, Pandas

Questions For Practice

Part-I Numpy Questions

Objective:
To build a strong foundation in using NumPy for numerical computations, including creating arrays, performing operations, and leveraging its powerful functionalities.

Assignment Tasks
Task 1: Creating Arrays

1. Write a Python program to:
    - Create a 1D NumPy array of 10 integers ranging from 1 to 10.
    - Create a 2D NumPy array of shape (3, 3) with integers from 1 to 9.

- Create a 3D NumPy array with random floating-point numbers of shape (3, 5, 3).

2. Display the shape, size, and datatype of each array.

Task 2: Array Indexing and Slicing

1. Create a NumPy array from the following list:
   data = [10, 20, 30, 40, 50, 60, 70, 80, 90]
2. Write a Python program to:
   - Retrieve the first three elements of the array.
   - Retrieve every alternate element of the array.
   - Reverse the array.
3. Perform the above operations and display the results.

Task 3: Mathematical Operations

1. Create two NumPy arrays, A and B, each with 5 random integers between 1 and 20.
2. Write a Python program to:
   - Add, subtract, multiply, and divide the two arrays element-wise.
   - Compute the dot product of the arrays.
   - Find the mean, median, standard deviation, and variance of array A.
   - Identify the maximum and minimum values in array B and their indices.

Task 4: Reshaping and Transposing

1. Create a 1D NumPy array of 12 integers ranging from 1 to 12.
2. Write a Python program to:
   - Reshape the array into a 2D array of shape (4, 3).
   - Reshape the array into a 3D array of shape (2, 2, 3).
   - Transpose the reshaped 2D array and display its shape.

Task 5: Boolean Masking and Filtering

1. Create a NumPy array with 15 random integers between 10 and 50.
2. Write a Python program to:
   - Find all elements greater than 25.
   - Replace all elements less than 30 with 0.
   - Count the number of elements divisible by 5.

Task 6: Working with Built-in Functions

1. Use NumPy's built-in functions to:
   - Create an array of 10 equally spaced values between 0 and 1.

- Create an identity matrix of size 4x4.
- Generate a 1D array of 20 random integers between 1 and 100, sort it, and find the 5 largest elements.

Task 7: Generic

1. Create a Python program that:
    - Generates two large random arrays of size (100, 100).
    - Performs matrix multiplication on the two arrays.
    - Finds the determinant and inverse of the resulting matrix (if possible).
    - Measures the time taken to complete these operations.

Part-II Pandas Questions

Assignment: Introduction to Pandas in Python
Objective:
To understand the basics of Pandas for data manipulation, analysis, and visualization, focusing on Series, DataFrames, and essential operations.

Assignment Tasks
Task 1: Working with Series

1. Create a Pandas Series from the following list:
   data = [25, 30, 35, 40, 45]
2. Write a Python program to:
   - Assign custom indices ['A', 'B', 'C', 'D', 'E'] to the Series.
   - Display the first three elements of the Series.
   - Calculate the mean, median, and standard deviation of the Series.

Task 2: Creating and Inspecting DataFrames

1. Create a Pandas DataFrame with the following data:

Name	Age	Gender	Marks
Alice	20	Female	85
Bob	22	Male	78
Carol	19	Female	92
David	21	Male	74
Eve	20	Female	88

2. Write a Python program to:
   - Display the first two rows of the DataFrame.
   - Display the column names, data types, and summary statistics.
   - Add a new column Passed that contains True if Marks >= 80 and False otherwise.

Task 3: Data Selection and Filtering

1. Using the DataFrame from Task 2, write a Python program to:
   - Select and display the Name and Marks columns.
   - Filter and display records where Marks > 80.
   - Display the record of the student with the highest marks.

Task 4: Handling Missing Data

1. Modify the DataFrame from Task 2 by introducing missing values:
2. df.loc[1, 'Marks'] = None
3. df.loc[4, 'Age'] = None
4. Write a Python program to:
    - Identify missing values in the DataFrame.
    - Fill missing values in the Marks column with the column's mean.
    - Drop rows where the Age column has missing values.

Task 5: Grouping and Aggregation

1. Using the DataFrame from Task 2, write a Python program to:
    - Group the data by Gender and calculate the mean age and marks for each gender.
    - Count the number of students in each gender group.

Task 6: Reading and Writing Data

1. Write a Python program to:
    - Save the modified DataFrame from Task 4 to a CSV file named students_data.csv.
    - Read the CSV file into a new DataFrame.
    - Display the first five rows of the newly loaded DataFrame.

Task 7: General

1. Download a sample dataset from Kaggle or use any public dataset.
2. Write a Python program to:
    - Load the dataset using Pandas.
    - Perform exploratory data analysis (EDA) by summarizing key statistics, checking for missing values, and visualizing data trends using Matplotlib or Seaborn.
    - Document your findings in Markdown or comments.

Coding Blocks | Online