Course: Pattern Recognition and ML Lab

Assignment 1: Library Exploration

Deadline – 1 day Before Next Lab

Instruction- Create a ipython Notebook for completing the assignment.

Notebook must contain- Your Name, Rollno, first copy past question in text block then write your code in code block.

Save and submit file - <name><rollno.>Assignment 1.ipynb

Part 1: Exploration of NumPy

1. **Array Creation**
   Create a 1D array, 2D array, and 3D array using NumPy. Print the shape, size, and dimensions of each array.
2. **Data Types in Arrays**
   Create a NumPy array with floating-point numbers. Convert it to an integer array and a boolean array.
3. **Indexing and Slicing**
   Create a 1D array of integers from 10 to 50 with a step of 5. Extract:
   - The first 3 elements
   - Every alternate element
   - Elements greater than 25
4. **Reshaping and Resizing**
   Create a 1D array of 12 elements and reshape it into:
   - A 2x6 array
   - A 3x4 array
     Verify the shapes of the resulting arrays.
5. **Arithmetic Operations**
   Create two arrays of size 3x3 with random integers. Perform element-wise addition, subtraction, multiplication, and division.
6. **Statistical Functions**
   Generate a 1D array of 50 random numbers between 1 and 100. Compute:
   - Mean
   - Median

- Standard deviation

7. **Dot Product**
   Create two 2x2 matrices. Compute their dot product and verify the result.

8. **Conditional Selection**
   Create an array of 20 random integers between 1 and 50. Extract all elements:
   - Greater than 25
   - Divisible by 5

9. **Masking**
   Create a 5x5 matrix of random integers between 10 and 50. Replace all values greater than 30 with -1.

10. **Broadcasting**
    Create a 4x4 array and add a 1D array of size 4 to it using broadcasting.

11. **Stacking and Splitting**
    Create two 1D arrays of size 10 each.
    - Stack them vertically and horizontally.
    - Split the resulting vertical stack into two equal parts.

12. **Sorting**
    Create a 1D array of 15 random integers between 1 and 100. Sort it:
    - In ascending order
    - In descending order

13. **Unique Elements**
    Create a 1D array of 20 random integers with possible duplicate values. Find all unique elements and their counts.

14. **Special Arrays**
    Create the following arrays using NumPy:
    - A 3x3 identity matrix
    - A diagonal matrix with [5, 10, 15] on the diagonal
    - A 4x4 matrix filled with zeros except for a border of ones

15. **Use Case: Linear Algebra**
    Solve the system of linear equations:

    $3x+2y+z=10$

$2x+3y+3z=18$

$x+y+2z=8$

Use NumPy's linalg.solve to find x,y,z.

Part 2: Exploration of Pandas

1. **Series Creation**
   Create a Pandas Series using a Python list, a NumPy array, and a dictionary. Print the index, values, and data type of each Series.
2. **DataFrame Creation**
   Create a DataFrame from:
   - A dictionary of lists
   - A NumPy array
   - A CSV file (use a sample dataset)
3. **Data Inspection**
   Use the following methods on the DataFrame created in Task 2:
   - .head()
   - .tail()
   - .info()
   - .describe()
4. **Data Selection**
   From the DataFrame created in Task 2, extract:
   - A single column
   - Multiple columns
   - Rows using .iloc and .loc
5. **Filtering Data**
   Create a DataFrame of student scores with columns: Name, Subject, and Score. Extract rows where the score is greater than 80 and subject is "Mathematics".
6. **Adding and Removing Columns**
   Create a DataFrame and add a new column based on existing columns. Remove a column using both del and .drop().

7. **Sorting**
   Create a DataFrame with random data. Sort it:
   - By a single column in ascending order
   - By multiple columns in descending order
8. **Renaming Columns and Index**
   Rename the columns and index of a DataFrame. Use both rename() and direct assignment.
9. **Detecting Missing Values**
   Create a DataFrame with some missing values. Use:
   - .isnull() and .notnull()
   - .sum() to count missing values per column
10. **Filling Missing Values**
    Fill missing values in the DataFrame using:
    - A constant value
    - The mean of the column
11. **Dropping Missing Values**
    Drop rows and columns with missing values using .dropna(). Experiment with different axis and how parameters.
12. **GroupBy and Aggregation**
    Create a DataFrame of sales data with columns: Salesperson, Region, and Sales. Use .groupby() to calculate:
    - Total sales per region
    - Average sales per salesperson
13. **Merging and Joining**
    Create two DataFrames:
    - One with employee details (EmployeeID, Name, Department)
    - One with salary details (EmployeeID, Salary)
      Merge them using different types of joins: inner, outer, left, and right.
14. **Pivot Tables**
    Create a DataFrame with sales data and generate a pivot table to analyze sales by Region and Product.
15. **File Operations**

- Read a dataset from a CSV file into a DataFrame.
- Write the DataFrame to an Excel file.
- Read a dataset from an Excel file.

Part 3: Basic Plotting

1. **Line Plot**
   Create a line plot for the following data:
   - X-axis: Numbers from 1 to 10
   - Y-axis: Squares of the numbers (e.g., y=x2y = x^2y=x2)
     Customize the plot by adding:
   - A title
   - Labels for the X and Y axes
   - A grid
2. **Bar Plot**
   Create a bar plot to visualize the following data:
   - Categories: Apples, Bananas, Cherries, Dates
   - Values: 40, 25, 35, 20
     Add:
   - Different colors for each bar
   - A legend
3. **Scatter Plot**
   Generate a scatter plot of random data:
   - X: 50 random numbers between 1 and 100
   - Y: 50 random numbers between 1 and 100
     Customize the scatter plot by:
   - Changing the marker size and color
   - Adding a title and axis labels
4. **Subplots**
   Create a figure with two subplots:
   - Subplot 1: A sine wave plot for values of xxx from 000 to 2π2\pi2π

- Subplot 2: A cosine wave plot for values of xxx from 000 to 2π2\pi2π Ensure both subplots have appropriate titles and axis labels.

5. **Histogram**
   Generate a histogram for a dataset of 1000 random numbers sampled from a normal distribution with mean 0 and standard deviation 1.
   - Add bins to the histogram.
   - Customize the color and edge color.
   - Save the plot as an image file (histogram.png).