Basic Array Operations:

1. Create a 3x3 array filled with random integers between 0 and 10. Calculate the sum, mean, and standard deviation of the array.

2. Create a 1D array of 10 elements and compute the cumulative sum of the elements.

3. Generate two 2x3 arrays with random integers and perform element-wise addition, subtraction, multiplication, and division.

4. Create a 4x4 identity matrix.

5. Given an array a = np.array([5, 10, 15, 20, 25]), divide each element by 5 using broadcasting.

Array Manipulation:

6. Reshape a 1D array of 12 elements into a 3x4 matrix.

7. Flatten a 3x3 matrix into a 1D array.

8. Stack two 3x3 matrices horizontally and vertically.

9. Concatenate two arrays of different sizes along a new axis.

10. Transpose a 3x2 matrix and then reshape it to have 3 rows and 2 columns.

Indexing and Slicing:

11. Given a 1D array of 15 elements, extract elements at positions 2 to 10 with a step of 2.

12. Create a 5x5 matrix and extract the sub-matrix containing elements from rows 1 to 3 and columns 2 to 4.

13. Replace all elements in a 1D array greater than 10 with the value 10.

14. Use fancy indexing to select elements from a 1D array at positions [0, 2, 4, 6].

15. Create a 1D array of 10 elements and reverse it using slicing.

Broadcasting:

16. Create a 3x3 matrix and add a 1x3 array to each row using broadcasting.

17. Multiply a 1D array of 5 elements by a scalar value using broadcasting.

18. Subtract a 3x1 column vector from a 3x3 matrix using broadcasting.

19. Add a scalar to a 3D array and demonstrate how broadcasting works across all dimensions.

20. Create two arrays of shapes (4, 1) and (1, 5) and add them using broadcasting.

Vectorized Operations:

21. Compute the square root of each element in a 2D array using vectorized operations.

22. Calculate the dot product of two 1D arrays of size 5.

23. Perform element-wise comparison of two 1D arrays and return an array of boolean values indicating where the first array has larger elements.

24. Create a 2D array and apply a vectorized operation to double the value of each element.

25. Create a 1D array of 100 random integers and compute the sum of all even numbers using vectorized operations.

Linear Algebra:

26. Create a 3x3 matrix and find its determinant.

27. Given a 2x2 matrix, compute its inverse and verify by multiplying with the original matrix to obtain the identity matrix.

28. Calculate the eigenvalues and eigenvectors of a 2x2 matrix.

29. Solve the system of linear equations 2x + 3y = 5 and 4x + 6y = 10 using NumPy.

30. Perform Singular Value Decomposition (SVD) on a 3x3 matrix and reconstruct the original matrix using the SVD components.

_____

1. Data Import and Cleaning

  - Import a CSV file into a Pandas DataFrame. Identify and drop rows with missing values.

  - Load a dataset and replace missing numerical values with the mean of the column.

  - Replace missing categorical values with the mode of the column in a DataFrame.

2. Data Transformation

  - Create a new column in a DataFrame that is the sum of two existing columns using NumPy vectorized operations.

  - Apply a mathematical function (e.g., square root) to all elements of a numerical column using NumPy.

  - Normalize a numerical column in a DataFrame using MinMaxScaler from sklearn.preprocessing and explain the transformation process.

3. Merging and Joining Datasets

  - Merge two DataFrames based on a common key and fill any missing values in the resulting DataFrame.

  - Perform a left join on two DataFrames with different keys and handle missing data in the result.

  - Concatenate two DataFrames along the columns and handle any duplicate column names.

4. Grouping and Aggregation

  - Group a DataFrame by a categorical column and calculate the mean and standard deviation of a numerical column for each group.

_____

- Use groupby() to calculate the sum of a column for each group, then apply a NumPy function to the grouped results.

- Create a pivot table from a DataFrame that groups data by two categorical columns and summarizes a numerical column using NumPy operations.

5. Array Operations and Manipulation

- Create a NumPy array from a DataFrame column and perform element-wise operations on the array.

- Reshape a NumPy array and assign it back to a new DataFrame column.

- Use NumPy to filter a DataFrame for rows where a numerical column's values are above a certain threshold.

6. Broadcasting and Vectorized Operations

- Broadcast a NumPy array across a DataFrame column to perform a vectorized operation.

- Create a new column in a DataFrame that results from a vectorized operation on multiple columns using NumPy.

- Demonstrate broadcasting by subtracting the mean of each row from the row's elements in a DataFrame.

7. Linear Algebra with NumPy

- Solve a system of linear equations using NumPy, and store the solution in a DataFrame.

- Compute the dot product of two columns from a DataFrame using NumPy.

- Perform matrix multiplication on two DataFrames treated as matrices and store the result in a new DataFrame.

---

8. Handling Missing Data

  - Interpolate missing values in a DataFrame using a linear method with NumPy and Pandas.

  - Use a mask created with NumPy to fill missing values in a DataFrame with a specific value.

  - Identify and replace outliers in a DataFrame column with the median value using a NumPy mask.

9. Advanced Data Analysis

  - Use a combination of groupby() and NumPy operations to analyze trends in a multi-level categorical dataset.

  - Create a summary DataFrame that includes the correlation matrix of numerical columns using both Pandas and NumPy.

  - Perform a rolling mean calculation on a time series dataset and visualize the result using Matplotlib.

10. DataFrame and Array Manipulation

  - Convert a DataFrame into a NumPy array, perform an operation, and convert it back to a DataFrame.

  - Use NumPy to generate a DataFrame with random values, then apply a condition to filter rows based on multiple criteria.

  - Create a DataFrame where each element is the result of applying a custom NumPy function to corresponding elements in two NumPy arrays.

11. Data Reshaping and Analysis

- Use NumPy's reshape to change the shape of an array extracted from a DataFrame and analyze the reshaped data.

- Stack two DataFrames vertically and use NumPy to analyze the combined dataset.

- Create a 3D array using NumPy and convert it into a MultiIndex DataFrame, then perform a group operation.

12. Time Series Analysis

- Convert a column in a DataFrame to a time series using Pandas and perform a NumPy-based operation on the time series data.

- Calculate the moving average of a time series in a DataFrame using a combination of Pandas and NumPy functions.

- Use pd.to_datetime to convert a string column to datetime format and use NumPy to compute the time difference between rows.

NumPy

1. Create a NumPy array of random numbers and normalize the data using min-max scaling.

2. Generate a 5x5 matrix with random integers and replace all occurrences of a specific value with another.

3. Create two NumPy arrays and perform element-wise addition, subtraction, multiplication, and division.

4. Solve the system of linear equations: $(3x + 4y = 7)$ and $(5x + 2y = 8)$ using NumPy.

5. Implement broadcasting to perform an operation between a 3x3 matrix and a 1D array.

6. Create a 3x3 identity matrix using NumPy.

---

7. Perform matrix multiplication between two 2D arrays.

8. Calculate the dot product and cross product of two vectors using NumPy.

9. Generate a NumPy array of 20 random integers and find the unique elements in the array.

10. Implement a function using NumPy that returns the inverse of a given matrix.

Pandas

11. Load a CSV file into a Pandas DataFrame and display the first 10 rows.

12. Perform groupby operations on a dataset to find the mean and sum of numerical columns based on a categorical column.

13. Handle missing data in a DataFrame by replacing NaN values with the column mean.

14. Merge two DataFrames on a common key and perform an inner, outer, left, and right join.

15. Convert a column of object type into a float type and handle any errors that occur during conversion.

16. Filter a DataFrame to select rows where a specific column's values fall within a given range.

17. Create a pivot table from a DataFrame and analyze the data based on multiple aggregations.

18. Use the apply() function to apply a custom function to each element in a Pandas Series.

19. Create a new column in a DataFrame that categorizes a numerical column into bins.

20. Replace all instances of a specific value in a DataFrame column with another value.

Data Cleaning & Preprocessing

21. Load a dataset with mixed data types and clean the data by removing or correcting any non-numeric values.

22. Identify and remove duplicate rows from a DataFrame.

_____

23. Normalize a dataset using StandardScaler from scikit-learn and explain the difference between min-max scaling and standard scaling.

24. Convert a categorical column to numeric using label encoding and explain its impact on the dataset.

25. Split a dataset into training and testing sets using an 80-20 split and shuffle the data before splitting.

Exploratory Data Analysis (EDA)

26. Perform an exploratory data analysis on a dataset by calculating summary statistics (mean, median, mode, standard deviation) for numerical columns.

27. Visualize the distribution of a numerical column using a histogram and a box plot. Identify and explain any outliers.

28. Create a correlation matrix of a DataFrame and visualize it using a heatmap. Interpret the results.

29. Use Pandas to create a scatter plot matrix of multiple columns in a dataset and analyze the relationships between them.

30. Perform feature engineering on a dataset by creating new features and then visualize their importance using a bar chart.