**13**.**Sorting methods**

#sorting

import numpy as np

a=[1,4,3,5,2,5,6]

print(np.sort(a))

**Output:** [1 2 3 4 5 5 6]

#argsort

import numpy as np

a=[1,4,3,5,2,6]

print(np.argsort(a))

**Output:** [0 4 2 1 3 5]

#unique

import numpy as np

a=[1,4,3,5,2,6]

print(np.unique(a))

**Output:** [1 2 3 4 5 6]

**PANDAS**

**14.work with following functions in series and data frames using pandas**

import pandas as pd

# Create a sample DataFrame

data = {

    'Name': ['Alice', 'Bob', 'Charlie', 'David'],

    'Age': [25, 30, 35, 40],

    'Score': [85, 90, 95, 100]}

df = pd.DataFrame(data)

print("Original DataFrame:\n", df)

# 1. Sort by a single column (ascending)

sorted\_by\_age = df.sort\_values('Age', ascending=True)

print("\nSorted by Age:\n", sorted\_by\_age)

# 2. Sort by multiple columns

sorted\_by\_age\_score = df.sort\_values(['Age', 'Score'])

print("\nSorted by Age, then Score:\n", sorted\_by\_age\_score)

# 3. Reset the index after sorting

reset\_df = sorted\_by\_age\_score.reset\_index(drop=True)

print("\nAfter reset\_index:\n", reset\_df)

# 4. Sort by index (first set custom index for better demo)

df\_indexed = df.set\_index('Name')

print("\nDataFrame with 'Name' as index:\n", df\_indexed)

sorted\_by\_index = df\_indexed.sort\_index()

print("\nSorted by index (Name):\n", sorted\_by\_index)

# Series example with sort\_values and sort\_index

scores = pd.Series([85, 90, 95, 100], index=['Alice', 'Bob', 'Charlie', 'David'])

print("\nOriginal Series:\n", scores)

sorted\_series\_values = scores.sort\_values()

print("\nSeries sorted by values:\n", sorted\_series\_values)

sorted\_series\_index = scores.sort\_index()

print("\nSeries sorted by index:\n", sorted\_series\_index)

**Output:** Original DataFrame:

       Name  Age  Score

0    Alice   25     85

1      Bob   30     90

2  Charlie   35     95

3    David   40    100

Sorted by Age:

       Name  Age  Score

0    Alice   25     85

1      Bob   30     90

2  Charlie   35     95

3    David   40    100

Sorted by Age, then Score:

       Name  Age  Score

0    Alice   25     85

1      Bob   30     90

2  Charlie   35     95

3    David   40    100

After reset\_index:

       Name  Age  Score

0    Alice   25     85

1      Bob   30     90

2  Charlie   35     95

3    David   40    100

DataFrame with 'Name' as index:

          Age  Score

Name

Alice     25     85

Bob       30     90

Charlie   35     95

David     40    100

Sorted by index (Name):

          Age  Score

Name

Alice     25     85

Bob       30     90

Charlie   35     95

David     40    100