**5. Linear algebraic functions**

import numpy as np

# Create matrices

A = np.array([[1, 2], [3, 4]])

B = np.array([[11, 12], [13, 14]])

print("Matrix A:\n", A)

print("Matrix B:\n", B)

# Transpose

print("\nTranspose of A:\n", A.T)

# Inverse

print("\nInverse of A:\n", np.linalg.inv(A))

# Determinant

print("Determinant of A:", np.linalg.det(A))

# Matrix Rank

print("Rank of A:", np.linalg.matrix\_rank(A))

# Dot Product

print("\nDot product of A and B:\n", np.dot(A, B))

# Eigenvalues and Eigenvectors

eigvals, eigvecs = np.linalg.eig(A)

print("\nEigenvalues of A:", eigvals)

print("Eigenvectors of A:\n", eigvecs)

# Singular Value Decomposition

U, S, Vt = np.linalg.svd(A)

print("\nSingular Value Decomposition of A:")

print("U:\n", U)

print("Singular values:", S)

print("V transpose:\n", Vt)

**Output:**

Matrix A:

 [[1 2]

 [3 4]]

Matrix B:

 [[11 12]

 [13 14]]

Transpose of A:

 [[1 3]

 [2 4]]

Inverse of A:

 [[-2.   1. ]

 [ 1.5 -0.5]]

Determinant of A: -2.0000000000000004

Rank of A: 2

Dot product of A and B:

 [[37 40]

 [85 92]]

Eigenvalues of A: [-0.37228132  5.37228132]

Eigenvectors of A:

 [[-0.82456484 -0.41597356]

 [ 0.56576746 -0.90937671]]

Singular Value Decomposition of A:

U:

 [[-0.40455358 -0.9145143 ]

 [-0.9145143   0.40455358]]

Singular values: [5.4649857  0.36596619]

V transpose:

 [[-0.57604844 -0.81741556]

 [ 0.81741556 -0.57604844]]

**6.Stacking and concatination**

import numpy as np

# Create two sample arrays

A = np.array([[11, 22], [33, 44]])

B = np.array([[55, 66], [77, 88]])

print("Matrix A:\n", A)

print("Matrix B:\n", B)

# Vertical stacking: stacks A on top of B

v\_stacked = np.vstack((A, B))

print("\nVertical Stack (vstack):\n", v\_stacked)

# Horizontal stacking: stacks A next to B

h\_stacked = np.hstack((A, B))

print("\nHorizontal Stack (hstack):\n", h\_stacked)

# You can also use np.concatenate for more control

concat\_vertical = np.concatenate((A, B), axis=0)

print("\nConcatenated vertically (axis=0):\n", concat\_vertical)

concat\_horizontal = np.concatenate((A, B), axis=1)

print("\nConcatenated horizontally (axis=1):\n", concat\_horizontal)

**Output:**

Matrix A:

 [[11 22]

 [33 44]]

Matrix B:

 [[55 66]

 [77 88]]

Vertical Stack (vstack):

 [[11 22]

 [33 44]

 [55 66]

 [77 88]]

Horizontal Stack (hstack):

 [[11 22 55 66]

 [33 44 77 88]]

Concatenated vertically (axis=0):

 [[11 22]

 [33 44]

 [55 66]

 [77 88]]

Concatenated horizontally (axis=1):

 [[11 22 55 66]

 [33 44 77 88]]

**7.Broadcasting**

broadcasted\_array = array + np.array([10, 20, 30])

print("Broadcasted array:\n", broadcasted\_array)

**output:** [[11 22 33]

 [14 25 36]

 [17 28 39]]