

```

import numpy as np

import matplotlib.pyplot as plt

X1 = np.array([[4, 2], [2, 4], [2, 3], [3, 6], [4, 4]])
X2 = np.array([[9, 10], [6, 8], [9, 5], [8, 7], [10, 8]])

mean1 = np.mean(X1, axis=0)
mean2 = np.mean(X2, axis=0)

print("Mean vector for class 1:", mean1)
print("Mean vector for class 2:", mean2)

S1 = np.cov(X1.T)
S2 = np.cov(X2.T)

Sw = S1 + S2

print("\nS1 (Covariance matrix for class 1):\n", S1)
print("\nS2 (Covariance matrix for class 2):\n", S2)
print("\nSw (Within-class scatter matrix):\n", Sw)

mean_diff = (mean1 - mean2).reshape(2, 1)
Sb = mean_diff @ mean_diff.T

print("\nSb (Between-class scatter matrix):\n", Sb)

eigenvalues, eigenvectors = np.linalg.eig(np.linalg.inv(Sw) @ Sb)

print("\nEigenvalues:", eigenvalues)
print("\nEigenvectors:\n", eigenvectors)

W = eigenvectors[:, np.argmax(eigenvalues)]

W_normalized = W / W[0]

print("\nNormalized projection vector W:", W_normalized)

Y1 = X1 @ W_normalized
Y2 = X2 @ W_normalized

print("\nProjected data for class 1:", Y1)
print("Projected data for class 2:", Y2)

plt.figure(figsize=(12, 5))

```

```
plt.subplot(121)

plt.scatter(X1[:, 0], X1[:, 1], label='Class 1')
plt.scatter(X2[:, 0], X2[:, 1], label='Class 2')

plt.title('Before applying LDA')

plt.xlabel('x1')
plt.ylabel('x2')

plt.legend()

plt.subplot(122)

plt.scatter(Y1, np.zeros_like(Y1), label='Class 1')
plt.scatter(Y2, np.zeros_like(Y2), label='Class 2')

plt.title('After applying LDA')

plt.xlabel('Projection axis')

plt.yticks([])

plt.legend()

plt.tight_layout()

plt.show()
```