Experiment Number : 5

Problem Statement: **Write Python code to apply different types of Dimensionality Reduction techniques on given data set.**

NAME: Harshwardhan PatilROLLNO: 50

CLASS: IT-B BATCH: B1

**Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.decomposition import PCA

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis as LDA

from sklearn.manifold import TSNE

from sklearn.datasets import load\_iris

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

# Load dataset (Iris dataset as an example)

data = load\_iris()

X = data.data

y = data.target

# Standardize the dataset

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Applying PCA (Principal Component Analysis)

pca = PCA(n\_components=2)

X\_pca = pca.fit\_transform(X\_scaled)

# Applying LDA (Linear Discriminant Analysis)

lda = LDA(n\_components=2)

X\_lda = lda.fit\_transform(X\_scaled, y)

# Applying t-SNE (t-Distributed Stochastic Neighbor Embedding)

tsne = TSNE(n\_components=2, perplexity=30, random\_state=42)

X\_tsne = tsne.fit\_transform(X\_scaled)

# Function to plot results

def plot\_results(X\_transformed, labels, title):

    plt.figure(figsize=(8, 6))

    plt.scatter(X\_transformed[:, 0], X\_transformed[:, 1], c=labels, cmap='viridis', edgecolors='k', alpha=0.7)

    plt.colorbar(label='Classes')

    plt.title(title)

    plt.xlabel('Component 1')

    plt.ylabel('Component 2')

    plt.show()

# Plot results

plot\_results(X\_pca, y, 'PCA - Principal Component Analysis')

plot\_results(X\_lda, y, 'LDA - Linear Discriminant Analysis')

plot\_results(X\_tsne, y, 't-SNE - t-Distributed Stochastic Neighbor Embedding')

Output:





