SET A

1. **Write a python program to demonstrate the use of PCA on the Iris dataset and visualizes the results in 2D.**

```
# Import necessary libraries
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA

# Load the Iris dataset
iris = load_iris()
X = iris.data  # Features
y = iris.target  # Target (species)
target_names = iris.target_names  # Species names

# Apply PCA to reduce the data to 2 dimensions
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

# Create a scatter plot of the 2D projection
plt.figure(figsize=(8, 6))

# Plot each class in a different color
for i, target_name in enumerate(target_names):
    plt.scatter(X_pca[y == i, 0], X_pca[y == i, 1], label=target_name)

# Add title and legend
plt.title("PCA of Iris Dataset")
plt.legend(title="Species")

# Show the plot
plt.show()
```

SET B

1. Write a python program to implement LDA use any synthetic dataset.

```
# Import necessary libraries
```

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.preprocessing import StandardScaler

# Step 1: Create a synthetic dataset
X, y = make_classification(n_samples=500, n_features=5, n_classes=3, random_state=42)

# Step 2: Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Step 3: Apply LDA to reduce the data to 2 dimensions
lda = LinearDiscriminantAnalysis(n_components=2)
X_lda = lda.fit_transform(X_scaled, y)

# Step 4: Create a scatter plot of the 2D projection
plt.figure(figsize=(8, 6))

# Plot each class in a different color
for i in range(3):
    plt.scatter(X_lda[y == i, 0], X_lda[y == i, 1], label=f'Class {i}')

# Add title and legend
plt.title("LDA of Synthetic Dataset")
plt.legend(title="Classes")

# Show the plot
plt.show()
```

SET C

1. **Write a python program applies t-SNE for dimensionality reduction and visualization of the Iris dataset.**

```
# Import necessary libraries
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.manifold import TSNE

# Step 1: Load the Iris dataset
iris = load_iris()
X = iris.data  # Features
y = iris.target  # Target (species)
target_names = iris.target_names  # Species names

# Step 2: Apply t-SNE to reduce the data to 2 dimensions
tsne = TSNE(n_components=2, random_state=42)
X_tsne = tsne.fit_transform(X)

# Step 3: Create a scatter plot of the 2D projection
plt.figure(figsize=(8, 6))

# Plot each class in a different color
for i, target_name in enumerate(target_names):
    plt.scatter(X_tsne[y == i, 0], X_tsne[y == i, 1], label=target_name)

# Add title and legend
plt.title("t-SNE of Iris Dataset")
plt.legend(title="Species")
```

```
# Show the plot

plt.show()
```