Lab Assignment: Dimensionality Reduction Using PCA and t-SNE

Objective:

The goal of this lab is to reduce the dimensionality of a dataset using **Principal Component Analysis (PCA)** and **t-SNE**. Students will learn how dimensionality reduction helps in visualizing high-dimensional data and improving model performance.

Tasks:

1. **Load the Dataset**
   - Use the **Wine Dataset** from `sklearn.datasets`. This dataset consists of 13 features related to wine classification.
   - Display the first 5 rows and describe the dataset.
   - Split the dataset into features (X) and target labels (y).
2. **Data Standardization**
   - Standardize the dataset using `StandardScaler` from `sklearn.preprocessing`. Standardization is important for PCA and t-SNE.
3. **Principal Component Analysis (PCA)**
   - Apply **PCA** to reduce the dataset to 2 components respectively.
   - For the 2-component PCA, visualize the data using a 2D scatter plot, with points colored based on their target labels.
4. **t-SNE Visualization**
   - Use **t-SNE** to reduce the dimensionality of the dataset to 2 dimensions.
   - Visualize the t-SNE transformed data using a 2D scatter plot and color the points based on their target labels.
5. **K-Means Clustering on Reduced Data**
   - Apply **K-Means Clustering** on the 2-component PCA reduced data.
   - Compare the clustering performance (using adjusted Rand index or silhouette score) with clustering performed on the original high-dimensional dataset.

- **Silhouette Score**: Measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). It ranges from -1 to 1, where a higher value indicates better-defined clusters.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Where:

- a(i)a(i)a(i) is the average intra-cluster distance (how close a point is to other points in the same cluster).
- b(i)b(i)b(i) is the average nearest-cluster distance (how close a point is to points in the nearest neighboring cluster).
-

```
from sklearn.datasets import make_blobs
 from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
X, y = make_blobs(random_state=42)
kmeans = KMeans(n_clusters=2, random_state=42)
silhouette_score(X, kmeans.fit_predict(X))
```

Example t-SNE	Example PCA
`model = TSNE(n_components = 2, random_state = 0)` `# configuring the parameters` `# the number of components = 2` `# default perplexity = 30` `# default learning rate = 200` `# default Maximum number of iterations` `# for the optimization = 1000` `tsne_data = model.fit_transform(data_1000)` `# creating a new data frame which` `# help us in plotting the result data` `tsne_data = np.vstack((tsne_data.T, labels_1000)).T` `tsne_df = pd.DataFrame(data = tsne_data,` `columns =("Dim_1", "Dim_2", "label"))` `# Plotting the result of tsne` `sn.scatterplot(data=tsne_df, x='Dim_1', y='Dim_2',` `hue='label', palette="bright")` `plt.show()`	`import numpy as np` `>>> from sklearn.decomposition import PCA` `X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])` `pca = PCA(n_components=2)` `pca.fit(X)` *`PCA(n_components=2)`*