**MINI PROJECT 3:**

**Exploring Unsupervised Learning Techniques**

**Objective**

The primary objective of this mini-project is to delve into unsupervised learning techniques, including clustering and dimensionality reduction, and understand their applications.

**Clustering Algorithms**

1. **K-Means Clustering**

* **Dataset:** Iris dataset
* **Implementation:**

from sklearn.cluster import KMeans

from sklearn.datasets import load\_iris

import matplotlib.pyplot as plt

iris = load\_iris()

X = iris.data

# Elbow Method to find optimal k

wcss = []

for k in range(1, 11):

kmeans = KMeans(n\_clusters=k, init='k-means++', random\_state=42)

kmeans.fit(X)

wcss.append(kmeans.inertia\_)

plt.plot(range(1, 11), wcss)

plt.title('Elbow Method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS')

plt.show()

# Using the optimal k from the elbow method (e.g., k=3)

kmeans = KMeans(n\_clusters=3, init='k-means++', random\_state=42)

y\_kmeans = kmeans.fit\_predict(X)

1. **Hierarchical Clustering**

* **Dataset:** Iris dataset
* **Implementation:**

from sklearn.cluster import AgglomerativeClustering

import scipy.cluster.hierarchy as shc

# Agglomerative clustering

hc = AgglomerativeClustering(n\_clusters=3, affinity='euclidean', linkage='ward')

y\_hc = hc.fit\_predict(X)

# Dendrogram

plt.figure(figsize=(10, 7))

shc.dendrogram(shc.linkage(X, method='ward'))

plt.title('Dendrogram')

plt.xlabel('Sample')

plt.ylabel('Euclidean Distance')

plt.show()

1. **DBSCAN**

* **Dataset:** A synthetic dataset with noise and varying density clusters
* **Implementation:**

from sklearn.cluster import DBSCAN

# DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

y\_dbscan = dbscan.fit\_predict(X)

**Dimensionality Reduction Techniques**

1. **Principal Component Analysis (PCA)**

* **Dataset:** Iris dataset
* **Implementation:**

from sklearn.decomposition import PCA

pca = PCA(n\_components=2)

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

1. **t-SNE**

* **Dataset:** Iris dataset
* **Implementation:**

from sklearn.manifold import TSNE

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

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

**Advanced Clustering Techniques**

* **Gaussian Mixture Models (GMM):** GMMs are probabilistic models that assume data points are generated from a mixture of Gaussian distributions. They are useful for clustering data with complex distributions.

**Comparison of Dimensionality Reduction Techniques**

* **PCA:** Good for linear relationships and preserving variance.
* **t-SNE:** Better for preserving local structure and visualizing non-linear relationships.

**Applications of Unsupervised Learning**

* **Customer Segmentation:** Clustering customers based on their behavior or demographics.
* **Anomaly Detection:** Identifying unusual data points that might indicate fraud or system failures.

**Note:** This is a basic outline. You can explore more datasets, experiment with different parameters, and visualize the results to gain a deeper understanding of unsupervised learning techniques.