**Anupriya(301251645)**

**Assignment 1( Dimensionality Reduction using PCA)**

**Analysis Report**

**Section 1: MNIST Dataset**

* **mnist = fetch\_openml('mnist\_784',version=1, as\_frame=False)**: This line loads the MNIST dataset, a popular dataset for handwritten digit recognition, using the **fetch\_openml** function from scikit-learn.
* **x = mnist["data"]** and **y = mnist["target"]**: These lines store the dataset in the **x** variable and the corresponding labels in the **y** variable.

**Section 2: PCA**

* **pca = PCA(n\_components=2)**: This line creates a PCA object with 2 components.
* **x\_pca = pca.fit\_transform(x)**: This line applies PCA to the dataset, reducing its dimensionality to 2 components.
* **print(f" Explained variance ratio: {pca.explained\_variance\_ratio\_}")**: This line prints the explained variance ratio, which indicates how much of the data's variability is explained by the 2 components.

**Section 3: Incremental PCA**

* **inc\_pca = IncrementalPCA(n\_components=154)**: This line creates an Incremental PCA object with 154 components.
* **for X\_batch in np.array\_split(x, n\_batches): inc\_pca.partial\_fit(X\_batch)**: This loop fits the Incremental PCA model in batches.
* **X\_reduced = inc\_pca.transform(x)**: This line reduces the dimensionality of the dataset using the fitted Incremental PCA model.

**Section 4: Swiss Roll Dataset**

* **X = generate\_swiss\_roll(n\_samples=1000, noise=0.1)**: This line generates the Swiss Roll dataset, a synthetic dataset, using a custom function **generate\_swiss\_roll**.

**Section 5: Kernel PCA**

* **kpca\_linear = KernelPCA(n\_components=2, kernel='linear')**: This line creates a Kernel PCA object with 2 components and a linear kernel.
* **X\_linear = kpca\_linear.fit\_transform(X)**: This line applies Kernel PCA to the Swiss Roll dataset using the linear kernel.
* **kpca\_rbf = KernelPCA(n\_components=2, kernel='rbf')** and **kpca\_sigmoid = KernelPCA(n\_components=2, kernel='sigmoid')**: These lines create Kernel PCA objects with 2 components and rbf and sigmoid kernels, respectively.
* **X\_rbf = kpca\_rbf.fit\_transform(X)** and **X\_sigmoid = kpca\_sigmoid.fit\_transform(X)**: These lines apply Kernel PCA to the Swiss Roll dataset using the rbf and sigmoid kernels, respectively.

**Screenshots of Output**

1. **Loading the datasets Mnist**

**A screen shot of a computer

Description automatically generated**

1. **Display every digit**

A screenshot of a computer

Description automatically generated

1. **Using PCA for Retrieving 1st and 2nd principal component ratio:**

A screen shot of a computer

Description automatically generated

1. **Projection of 1st and 2nd principal component**

A colorful dots on a white background

Description automatically generated

1. **Incremental PCA to reduce the dimensionality to 154 dimensions**

**A screenshot of a computer program

Description automatically generated**

1. **Displaying compressed digits:**

**A close-up of numbers

Description automatically generated**

1. **Generating Swiss Roll Dataset**

**A screen shot of a computer program

Description automatically generated**

1. **Plotting dataset in 3D**

A graph of colored dots

Description automatically generated with medium confidence

1. **Applying Kernal PCA**

**A screen shot of a computer program

Description automatically generated**

1. **Plotting KPCA results:**

A graph of a curve

Description automatically generated

1. **Finding the best kernel and gamma value for best accuracy and printing the best parameters using GridSearchCV**

**A screenshot of a computer program

Description automatically generated**

1. **Plotting GridSearchCV results:**

A graph with a line

Description automatically generated