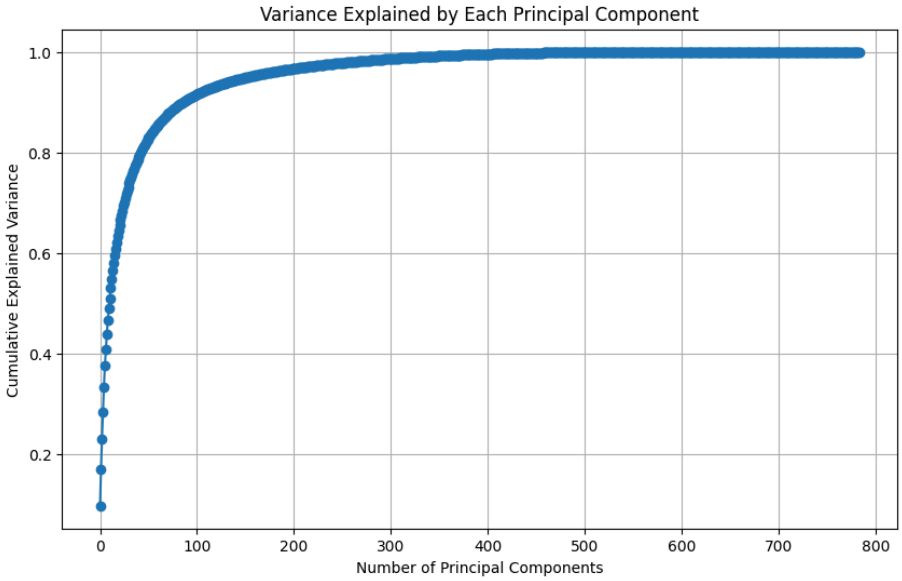
Name:Rajnish Maurya
Roll No.: DA24M015
Assignment: 3

Q(1). Download the MNIST dataset from https://huggingface.co/datasets/mnist. Use a random set of 1000 images (100 from each class 0-9) as your dataset.
i. Write a piece of code to run the PCA algorithm on this data-set. Visualize the images of the principal components that you obtain. How much of the variance in the data-set is explained by each of the principal components?

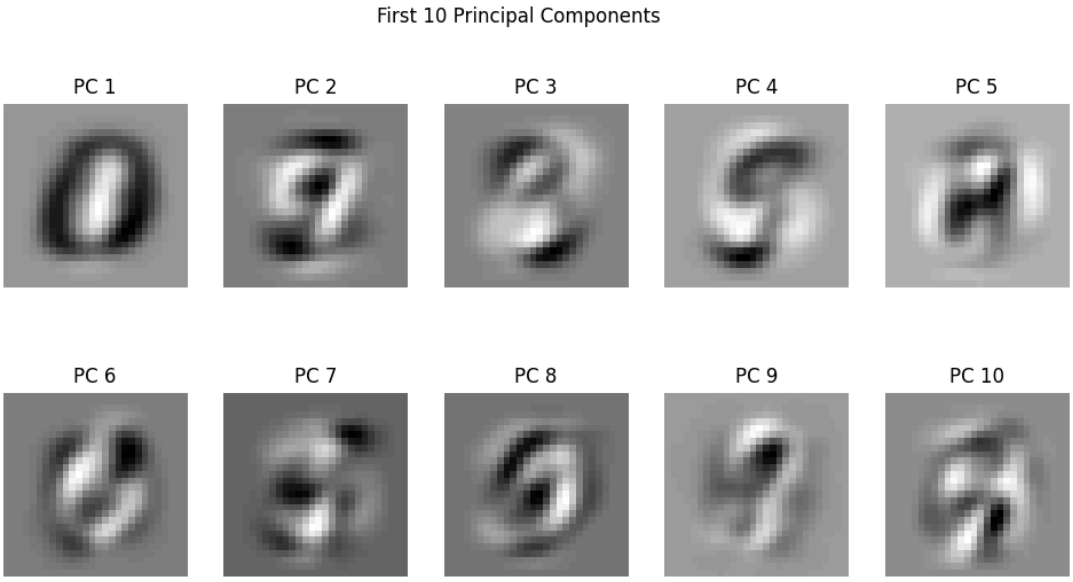
Methodology:
Preparation of the data:
Each 1,000 grayscale image of handwritten digits found in the dataset is a 784-dimensional vector (flattened from 28x28 pixel images). Subtracting the mean of every feature **standardized the dataset**, therefore centering the data.

The principle component analysis:
A covariance matrix was computed to record the interactions among several pixel characteristics.Calculated were eigenvalues and eigenvectors of the covariance matrix. While eigenvalues show the variance each component explains, eigenvectors match the primary components. Eigenvalues and eigenvectors were arranged in declining order of eigenvalues to rank the main components with maximum variance.

Variance for each principal component:
Along with the total variation among components, one computed the percentage of variance explained by every main component.



Principal Component Visualization:
Redressed into 28x28 pixel pictures and displayed, the top 10 principal components were



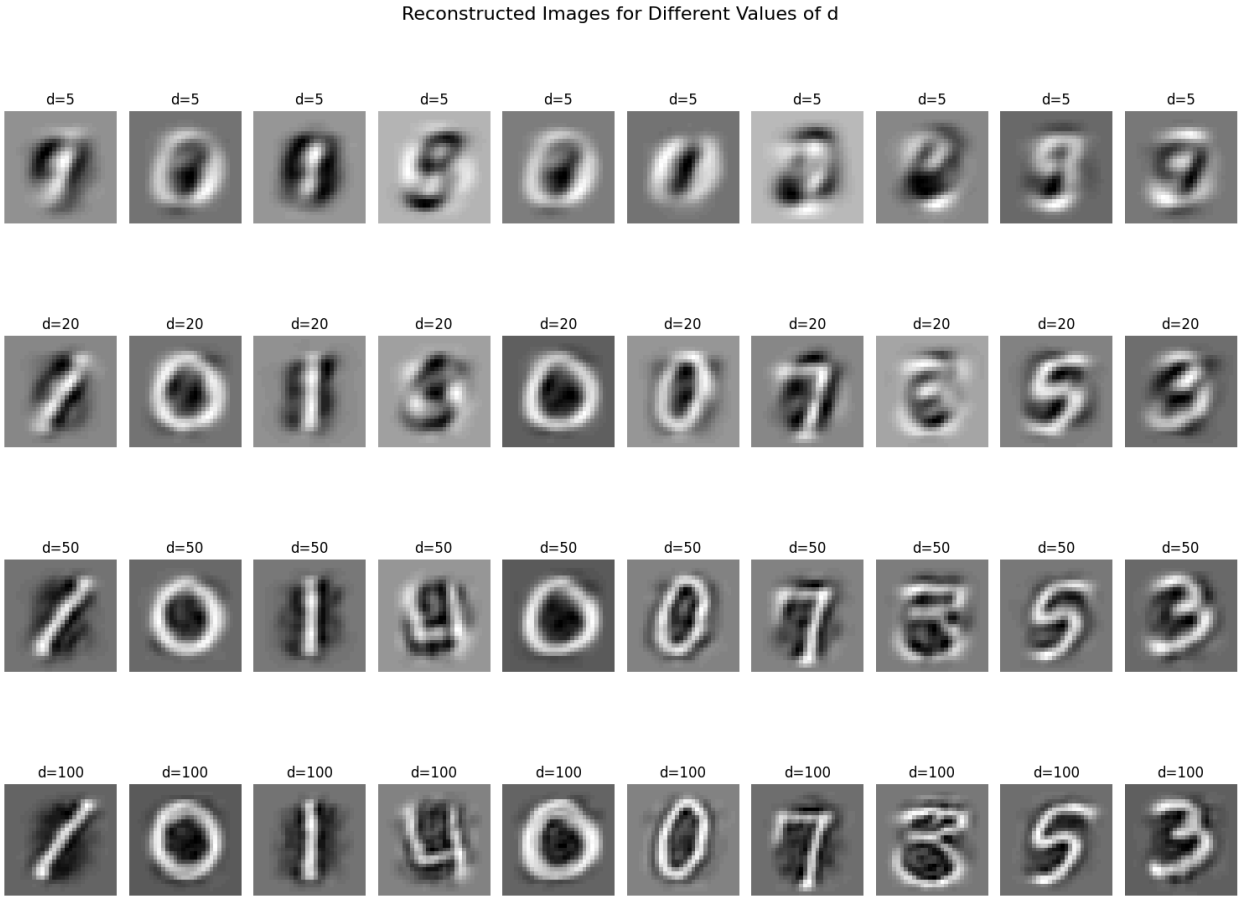
ii. Reconstruct the dataset using different dimensional representations. How do these look like? If you had to pick a dimension d that can be used for a downstream task where you need to classify the digits correctly, what would you pick and why?

Data Preprocessing:
Each 1,000 grayscale image of handwritten digits in the dataset is a 784-dimensional vector (flattened from 28x28 pixel images).Centering the data on the mean helped to ensure zero mean across all features.

PCA Applied Method:

Calculated to record feature relationships is the covariance matrix. Taken eigenvalues and eigenvectors from the covariance matrix.For every selected dimension, the top eigenvectors were projected the data to a lower-dimensional space and subsequently reconstructed back to the original space.

Reconstruction and Visualization: For varying d values 5, 20, 50, and 100 the images were rebuilt. Visualized and compared were rebuilt images for the first ten samples.



Q2). You are given a data-set with 1000 data points each in R2(cm dataset 2.csv).
i. Write a piece of code to implement the Llyod"s algorithm for the K-means problem with k = 2 . Try 5 different random initialization and plot the error function w.r.t iterations in each case. In each case, plot the clusters obtained in different colors.

Objective:
The goal was to implement Lloyd's algorithm for k = 2 and analyze its behavior across 5 different random initializations.

Implementation:
Lloyd's algorithm consists of:

- Randomly choose k starting centroids.
- Give every data point the closest centroid's location.
- Update: Determine the centroids by averaging the allocated points.
- Continue until convergence or a predetermined count of iterations.

The error function sum of squared distances from points to their assigned centroids was kept and visualized throughout iterations for every initialization.

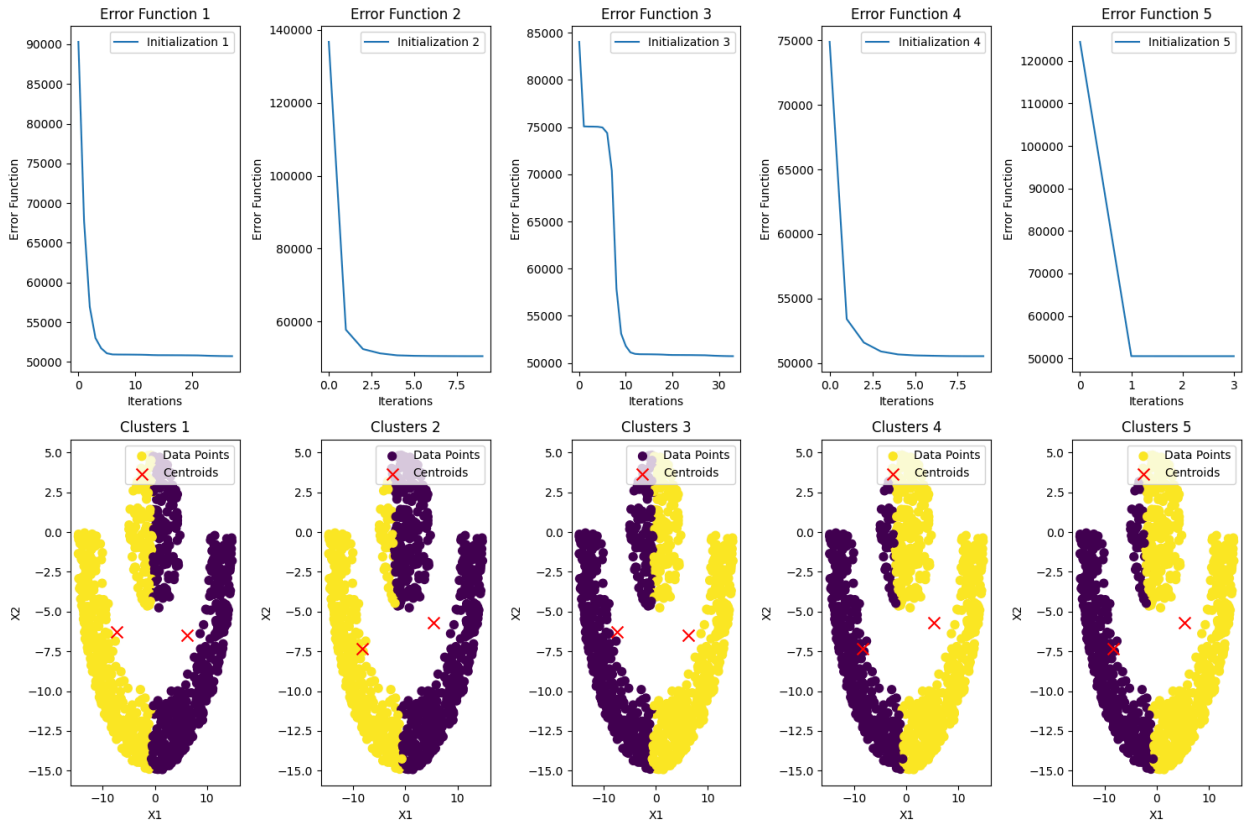
As expected, the error function fell monotonically with iterations since the method maximizes the sum of squared distances.
Lloyd's sensitivity to initialization is shown by varied initializations producing different convergence speeds and maybe different end clusters.

Plots:

- Five graphs displaying the error function declining across iterations for every random initialization show against each other.
- Scatter plots for every starting display the data points in varying colors depending on their cluster assignment with centroids highlighted.

Observations:

- Say initializing 5 found some initializations converged faster than others.
- The method might converge to several local optima depending on startup.



ii. For each K = {2, 3, 4, 5}, Fix an arbitrary initialization and obtain cluster centers according to K-means algorithm using the fixed initialization. For each value of K, plot the Voronoi regions associated to each cluster center. (You can assume the minimum and maximum value in the data-set to be the range for each component of R2).

Goal: Use fixed centroids to execute the K-means algorithm for each K in {2, 3, 4, 5}, and then plot the corresponding Voronoi areas.

Execution:

1. Fixed Initialization: For every value of K, manually set the initial centroids.
2. Execute K-means:
    - To determine the final cluster centers, apply Lloyd's algorithm.

3. Draw Regions of Voronoi:
    - Scipy.spatial was used.Using Voronoi to show diagrams of voronoi.
    - Plotted the cluster centers, data points, and Voronoi regions.

Results:

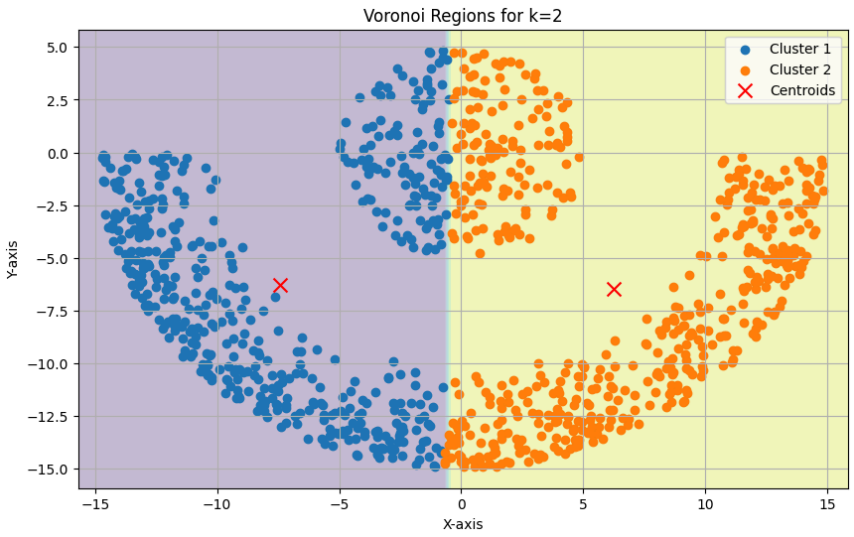
- The dataset was divided into K separate clusters by well-defined voronoi areas.
- The regions grew smaller and more regionally specialized as K rose.

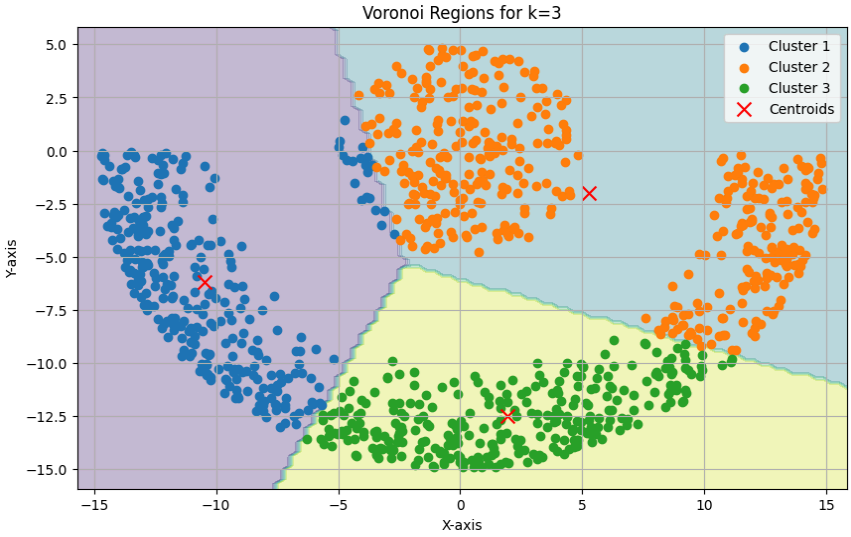
Plots:

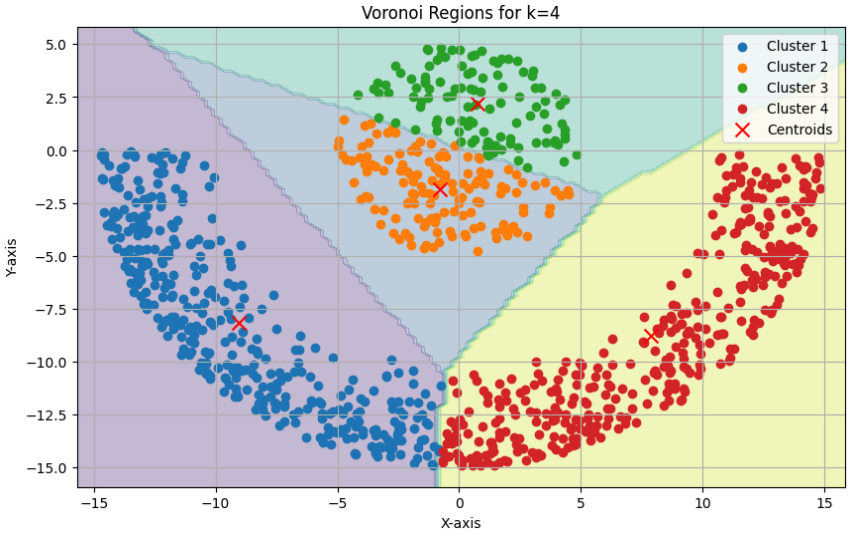
1. Diagrams of voronoi:
    - A diagram displaying the dataset points and Voronoi regions for every K.
2. Assignments to Clusters:
    - Cluster assignments were used to color-code the data points.

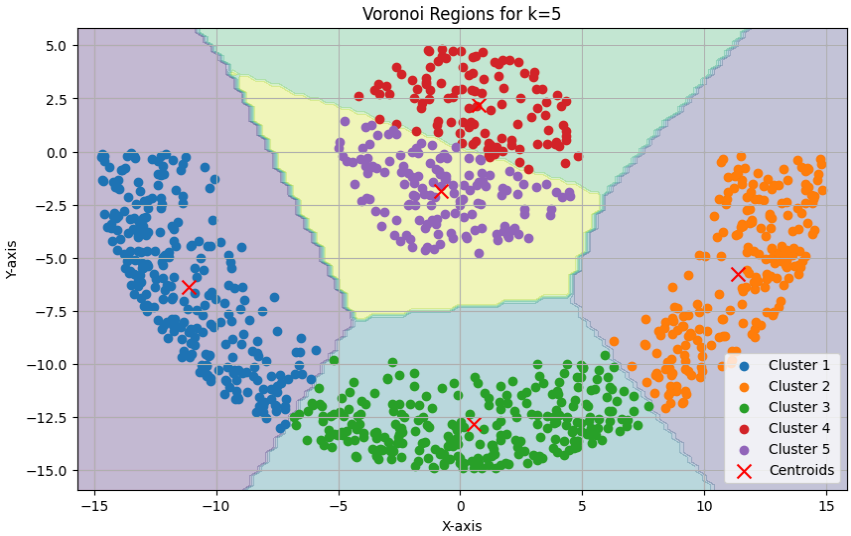
Observations:

- The dataset's division was visually validated by the Voronoi diagrams.
- Partitions became more sophisticated as K increased, but big K can cause overfitting.




Voronoi Regions for k=3
Cluster 1
Cluster 2
Cluster 3
Centroids
Y-axis
X-axis


Voronoi Regions for k=4
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Centroids
Y-axis
X-axis


Voronoi Regions for k=5
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Centroids
Y-axis
X-axis

iii. Is the Llyod's algorithm a good way to cluster this dataset? If yes, justify your answer. If not, give your thoughts on what other procedure would you recommend to cluster this dataset?
Solution:
No, this dataset cannot be clustered using Lloyd's Algorithm (K-means).

Explanation:

- Lloyd's algorithm (K-means) assigns data points to clusters depending on **Euclidean distance** and **linear decision limits**.
- The data points in the provided set clearly show a **non-linear structure** similar to two **"half-moons".**
- K-means neglects the curved limits separating the two clusters. Though they technically belong to another cluster, points close to the boundary could be misclassified depending on closeness to centroids.
- K-means makes the assumption that clusters are spherical and homogeneous in size. Datasets having non-linear or complicated geometric forms such as the present one do not fit this assumption.
- Though K-means reduces the sum of squared distances to cluster centroids, this goal is insufficient for datasets including **non-linearly separable clusters**.

Several Clustering Techniques:

The following approaches can be used to solve Lloyd's algorithm's restrictions in this particular situation:

Spectral Clustering:
Spectral clustering projects data into a higher-dimensional space where non-linear clusters become linearly separable.It then uses this altered space clustering techniques including K-means.
Why it's better: For datasets like the present one, spectral clustering is well-suited and manages the non-linear limits between clusters.

Kernel K-Means:
Kernel K-means maps a dataset into a higher-dimensional space where non-linear clusters can be better distinguished using a kernel function e.g. **RBF** kernel.
Why is that better: While maintaining the advantages of the K-means algorithm, this method lets one handle intricate cluster forms.

Gaussian Mixture Models:
GMM, or Gaussian Mixture Models, adapts the model to maximize the likelihood by assuming that data points arise from a mix of Gaussian distributions.
Why is it improved: By modeling elliptical or non-spherical clusters, GMM offers additional freedom for datasets with intricate boundaries.