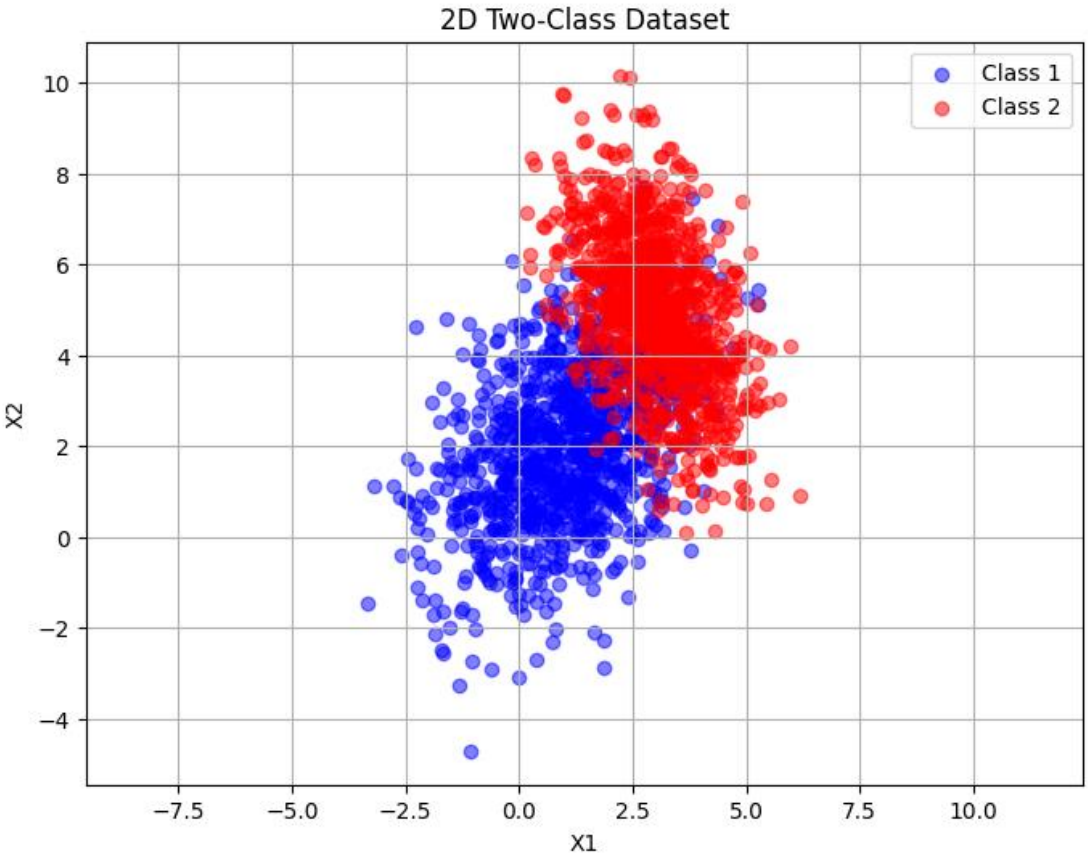
Q1) Generate a 2 dimensional 2 class dataset with 1000 examples for each class where

$\mu_1 = [1\ 2], \mu_2 = [3\ 5]$ and

$\Sigma_1 = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 3 \end{bmatrix}$
plot the dataset.

(You can use built in functions to generate data)

a) Plot the generated datasets.



b) Partition the dataset into training (80%) and test set (20%) randomly. Compute the mean vectors and covariance matrices of the classes for training set. (In order to compute the covariance matrix and mean vectors use the formulas that are given in the book. You are not allowed to use built in functions.

Class 1 and class 2 were split into training and test sets. The mean vector for the training sets was calculated by dividing the sum of the 2 features of the 800 training samples by 800, which is the number of samples. The covariance matrices for both classes were also calculated using the following formula:

$$\mathrm{Cov}(X) = \frac{1}{n-1}(X-\mu)^T(X-\mu)$$

Mean Vector of Class 1:

$$\begin{bmatrix} 1.0243737012162313 \\ 2.003704530694493 \end{bmatrix}$$

Covariance Matrix of Class 1:

$$\begin{bmatrix} 1.98543696 & 0.97002468 \\ 0.97002468 & 3.04867979 \end{bmatrix}$$

Mean Vector of Class 2:

$$\begin{bmatrix} 3.009585326228859 \\ 5.018832540895357 \end{bmatrix}$$

Covariance Matrix of Class 2:

$$\begin{bmatrix} 1.04466239 & -0.82717713 \\ -0.82717713 & 2.89386274 \end{bmatrix}$$

c) Design a quadratic discriminant classifier using different covariance matrices. Obtain the training and test errors. Draw the decision boundary.

Formula used for the quadratic discriminant classifier:

QUADRATIC DISCRIMINANT

which defines a *quadratic discriminant* (see figure 5.3) that can also be written as

(5.20) $$g_i(\boldsymbol{x}) = \boldsymbol{x}^T \mathbf{W}_i \boldsymbol{x} + \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}$$

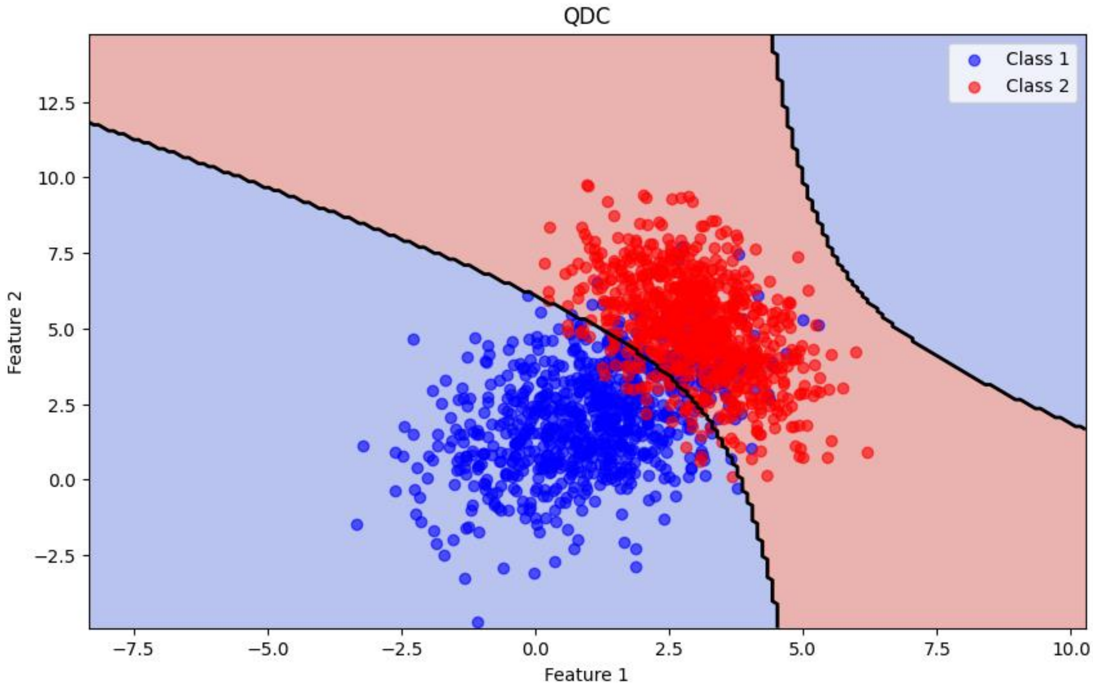
where

$$\begin{aligned} \mathbf{W}_i &= -\frac{1}{2}\mathbf{S}_i^{-1} \\ \boldsymbol{w}_i &= \mathbf{S}_i^{-1}\boldsymbol{m}_i \\ w_{i0} &= -\frac{1}{2}\boldsymbol{m}_i^T\mathbf{S}_i^{-1}\boldsymbol{m}_i - \frac{1}{2}\log|\mathbf{S}_i| + \log\hat{P}(C_i) \end{aligned}$$

(X represents samples, m represents mean vector, S represents covariance matrix. Also, prior probability is assumed to be 0.5 for both classes). For each sample, the g(x) values were calculated for both class 1 and class 2. For a given sample, whichever class had the higher g(x) value determined the class to which that sample was assigned.

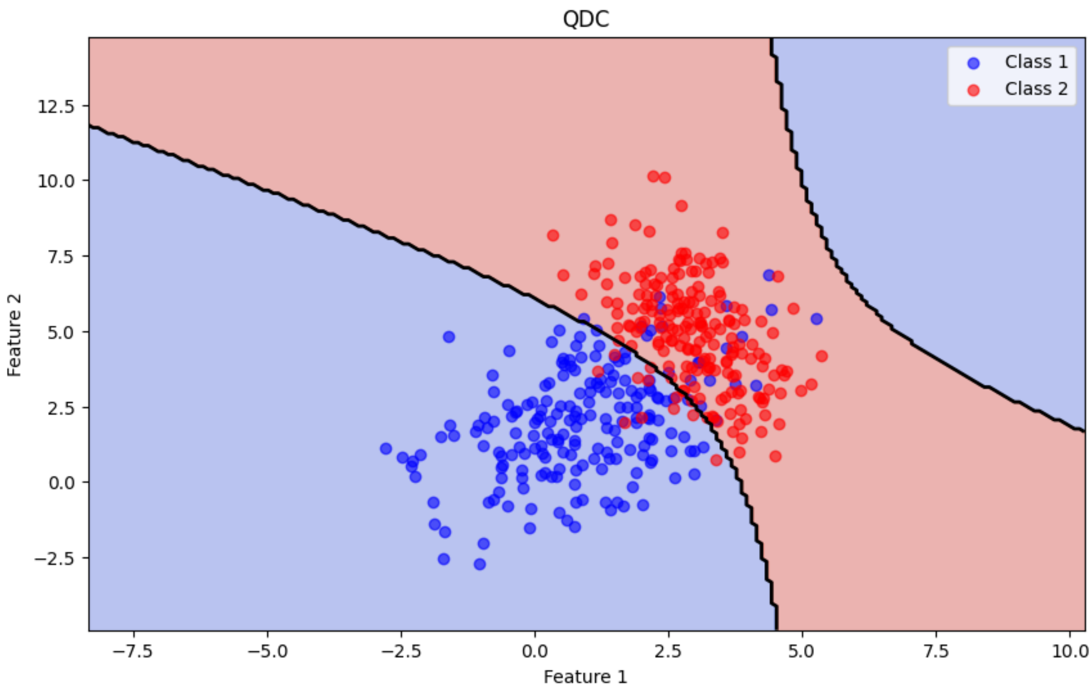
Train Error for Class 1: 0.145
Train Error for Class 2: 0.0625

Plot for Training Set:



Test Error for Class 1: 0.11
Test Error for Class 2: 0.055

Plot for Test Set:



d) Design a linear discriminant classifier using a shared covariance matrix. Obtain the training and test errors. Draw the decision boundary

Since a common covariance matrix is used, the common covariance matrix was calculated by taking the average of the class 1 and class 2 covariance matrices.

Covariance matrix:

$$\begin{bmatrix} 1.51504967 & 0.07142378 \\ 0.07142378 & 2.97127126 \end{bmatrix}$$

Formula used for the quadratic discriminant classifier:

$$(5.23) \quad g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}$$

where

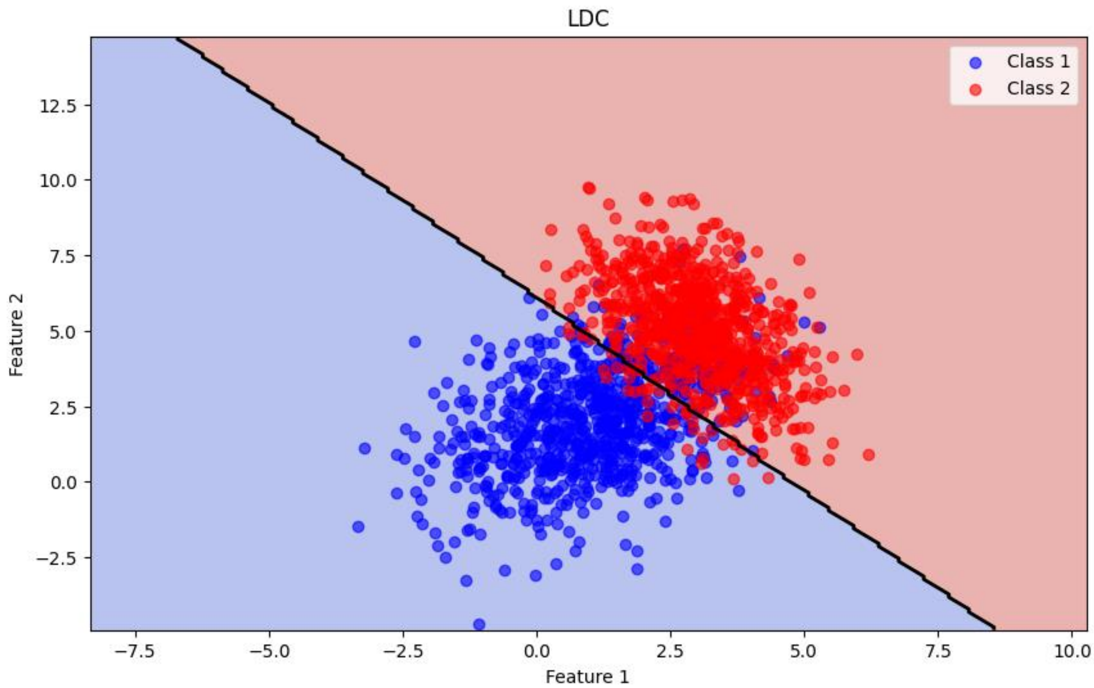
$$\boldsymbol{w}_i = \mathbf{S}^{-1}\boldsymbol{m}_i$$

$$w_{i0} = -\frac{1}{2}\boldsymbol{m}_i^T \mathbf{S}^{-1}\boldsymbol{m}_i + \log \hat{P}(C_i)$$

(X represents samples, m represents mean vector, S represents covariance matrix. Also, prior probability is assumed to be 0.5 for both classes). For each sample, the g(x) values were calculated for both class 1 and class 2. For a given sample, whichever class had the higher g(x) value determined the class to which that sample was assigned.

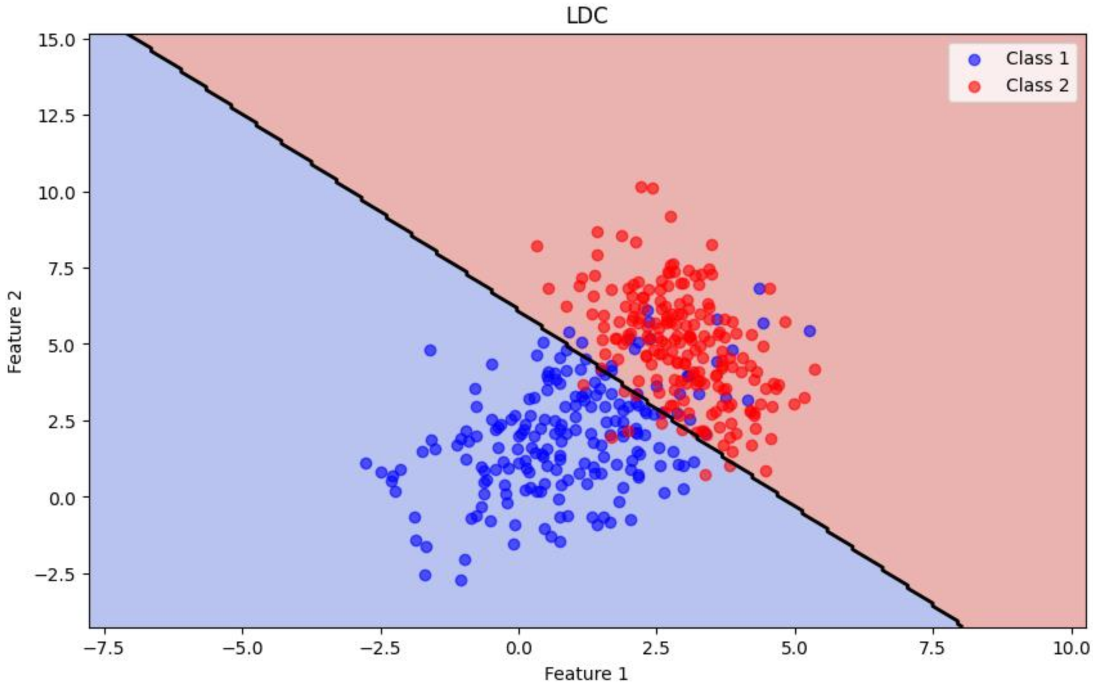
Train Error for Class 1: 0.1825
Train Error for Class 2: 0.05

Plot for Train Set:



Test Error for Class 1: 0.135
Test Error for Class 2: 0.045

Plot for Test Set:



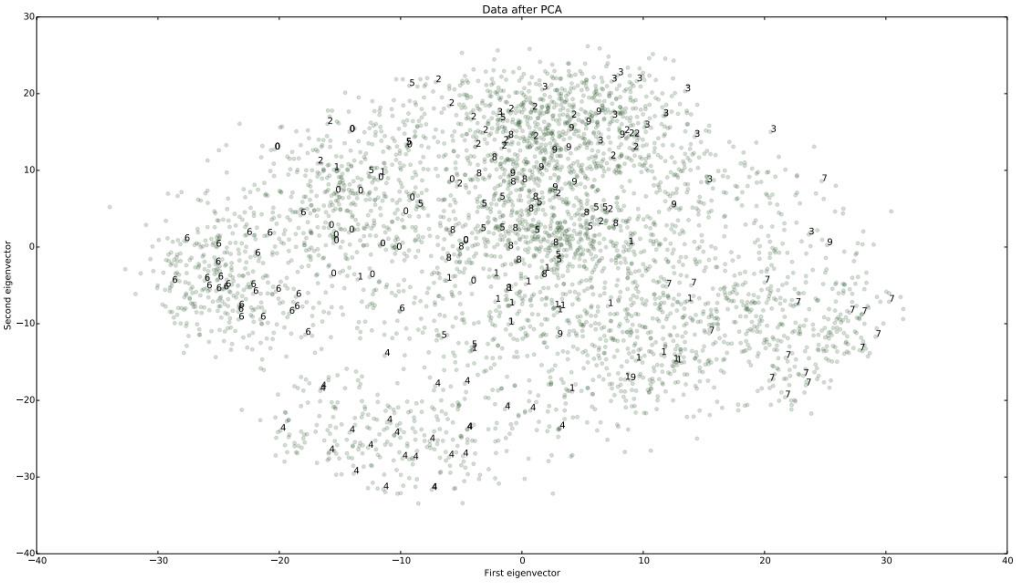
If the source code is run again, the numerical results and graphs may show slight differences because the samples are obtained randomly.

Q2) You will use the attached opdigits dataset for this assignment. The last column of each row in the file represents the class label (an integer from 0 to 9).

Important:
You are required to implement PCA from scratch.
Do not use any built-in pca() functions.
Do not use cov() or mean() functions.
You may use the eig() function to compute eigenvalues and eigenvectors.

Tasks:
Implement PCA on the data provided in data.txt.
The last column (class label) should be excluded during PCA computation.
Compute the covariance matrix Σ to calculate the PCA components.
Project the data into a 2-dimensional space using the top two principal components.
Plot the 2D projections:
Include all data points in the plot.
Randomly select 200 instances, and annotate their corresponding class labels at their projected positions. An example plot is given below.
Use a function like annotate() to display the text (class label) at each of the 200 points.

Include all your plots in your report.



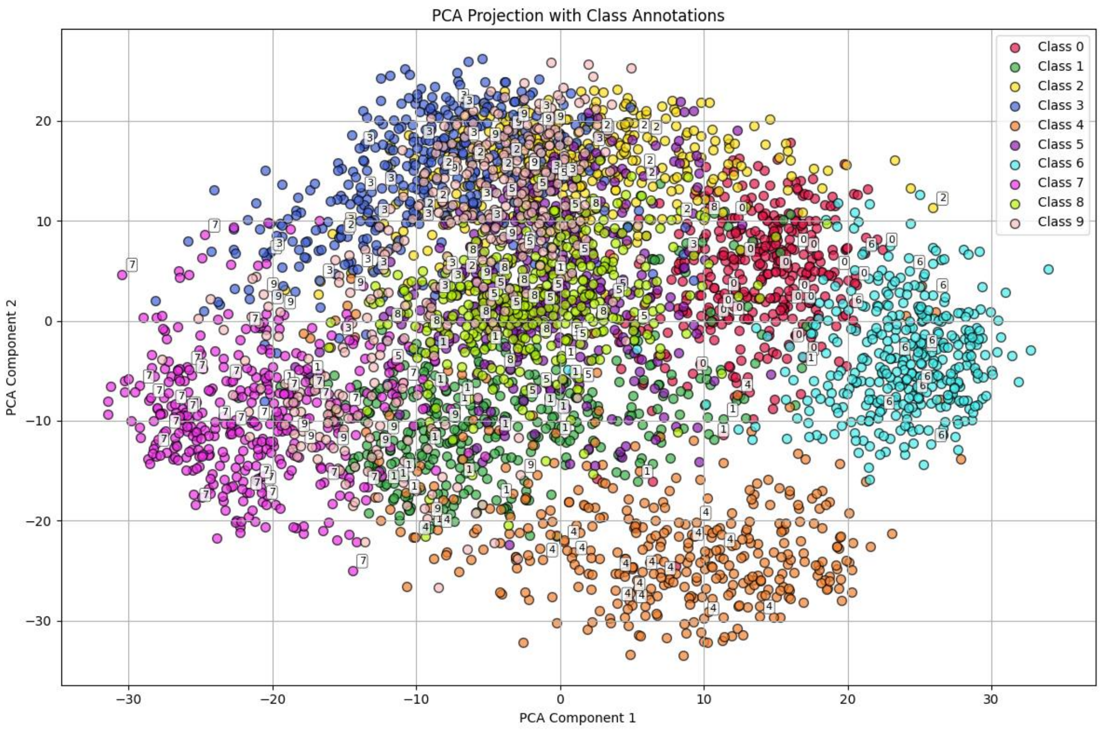
Step 1: The mean vector is computed.

Step 2: The covariance matrix is calculated for all samples using the formula mentioned in the first question.

Step 3: The eigenvectors and eigenvalues of this covariance matrix are computed.

Step 4: The two eigenvectors with the largest eigenvalues are considered the most important two principal components.

Step 5: The data is then projected onto these components.



(Annotated class numbers can be seen in the small boxes.)

Q3)
Apply Linear Discriminant Analysis (LDA) on the opdigits dataset and project the data into a 2-dimensional space, similar to what you did in Question 2.

Instructions:
You must implement the LDA algorithm from scratch.
You may use the eig() function to compute eigenvalues and eigenvectors.
Do not use any built-in LDA functions or libraries.

Step 1: Similar to PCA, the mean vector is calculated.

Step 2: The within-class scatter matrix (Sw) and the between-class scatter matrix (Sb) are computed.

$$S_w = \sum_{k=1}^{c} \sum_{\mathbf{x}_i \in \mathcal{C}_k} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T$$

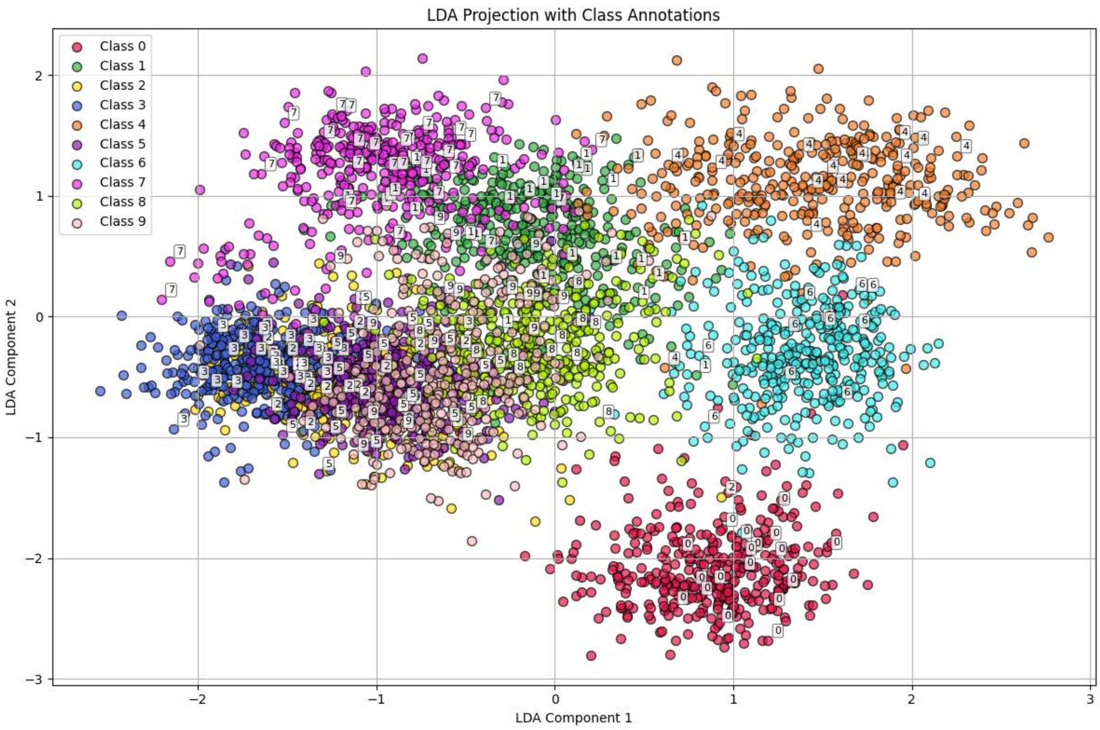
$$S_b = \sum_{k=1}^{c} N_k(\mu_k - \mu)(\mu_k - \mu)^T$$

Step 3: The inverse of Sw is multiplied by Sb.

Step 4: The eigenvalues and eigenvectors of the resulting matrix are calculated.

$$S_w^{-1} S_b \mathbf{v} = \lambda \mathbf{v}$$

Step 5: The dataset is then projected onto the 2 eigenvectors corresponding to the 2 largest eigenvalues.



(Annotated class numbers can be seen in the small boxes.)

Notes:
Use the class labels (last column of the dataset) when computing the within-class and between-class scatter matrices.
Project the data into 2D using the two most discriminative LDA components.
Plot the 2D projections of all data points.
As in Question 2, randomly select 200 instances and annotate each with its corresponding class label using a function like annotate().
Include your implementation, plot(s), and a brief explanation of your results in the report.