Assignment 01

Jayesh Kirtane
Student ID: M24DS008

Q1)

Given the following data, use PCA (Principal Component Analysis) to reduce the dimension from 2D to 1D

Feature	Example 01	Example 02	Example 03	Example 04
x	2	6	10	14
y	5	4	11	14

Table 1: Data for PCA

Answer 1

To Perform dimensionality reduction using PCA, We will follow the following steps:

STEP 1: COMPUTING THE MEAN VECTOR

The mean vector µ is given as:

$$\mu = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}$$

Where $\bar{x}$ and $\bar{y}$ is given as:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

therefore,

$$\bar{x} = \frac{2+6+10+14}{4}$$

$$\bar{x} = \frac{32}{4}$$

$$\bar{x} = 8$$

$$\bar{y} = \frac{5+4+11+14}{4}$$

$$\bar{Y} = \frac{34}{4}$$

$$\bar{Y} = 8.5$$

$$\mu = \begin{bmatrix} 8 \\ 8.5 \end{bmatrix}$$

STEP 2: NORMALISING THE DATA

To normalise the data we will do:

$$x_i = x_i - \bar{x}$$

$$y_i = y_i - \bar{y}$$

So after normalization the data looks like:

x	-6	-2	2	6
y	-3.5	-4.5	2.5	5.5

Let **X** be the matrix form of the data:

$$\mathbf{X} = \begin{bmatrix} -6 & -2 & 2 & 6 \\ -3.5 & -4.5 & 2.5 & 5.5 \end{bmatrix} \tag{1}$$

STEP 3: CALCULATING THE COVARIANCE MATRIX:

Let the Covariance Matrix Be **M** then it is given as:

$$\mathbf{M} = \frac{X \cdot X^T}{n-1}$$

where n is the number of datapoints

$$\mathbf{M} = \frac{1}{4-1} \begin{bmatrix} -6 & -2 & 2 & 6 \\ -3.5 & -4.5 & 2.5 & 5.5 \end{bmatrix} \cdot \begin{bmatrix} -6 & -2 & 2 & 6 \\ -3.5 & -4.5 & 2.5 & 5.5 \end{bmatrix}^T$$

$$\mathbf{M} = \frac{1}{3}\begin{bmatrix} 80 & 68 \\ 68 & 69 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} 26.67 & 22.67 \\ 22.67 & 23 \end{bmatrix}$$

STEP 4: CALCULATING EIGEN VALUES AND EIGEN VECTORS OF COVARIANCE MATRIX:

our covariance matrix is $\mathbf{M}$;

$$\mathbf{M} = \begin{bmatrix} 26.67 & 22.67 \\ 22.67 & 23 \end{bmatrix}$$

The eigenvalues λ are found by solving the characteristic equation:

$$\det(\mathbf{M} - \lambda\mathbf{I}) = 0$$

Expanding $\mathbf{M} - \lambda\mathbf{I}$, we have:

$$\mathbf{M} - \lambda\mathbf{I} = \begin{bmatrix} 26.67 - \lambda & 22.67 \\ 22.67 & 23 - \lambda \end{bmatrix}$$

The determinant is:

$$|\mathbf{M} - \lambda\mathbf{I}| = (26.67 - \lambda)(23 - \lambda) - (22.67)^2 = 0$$

Expanding and simplifying the determinant equation:

$$(26.67 \cdot 23) - (26.67\lambda + 23\lambda) + \lambda^2 - 22.67^2 = 0$$

$$\lambda^2 - (26.67 + 23)\lambda + (26.67 \cdot 23 - 22.67^2) = 0$$

Solving the above equation, we get the eigenvalues:

$$\lambda_1 = 47.57, \quad \lambda_2 = 2.09$$

The larger eigen values corresponds to a direction with greater variance. Which ensures the capturing of the most significant features from the dataset. There we will be taking λ_1 for getting the principal component:

Eigen vector for $\boldsymbol{\lambda_1 = 47.57}$ is given as:

We solve:

$$[M - \lambda I] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -20.90 & 22.67 \\ 22.67 & -24.57 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

To find $\mathbf{v}$, solve the system of linear equations:

$$-20.90v_1 + 22.67v_2 = 0$$

$$22.67v_1 - 24.57v_2 = 0$$

Solving this system, we get:

$$\mathbf{v} = \begin{bmatrix} 24.57 \\ 22.67 \end{bmatrix}$$

To normalize $\mathbf{v}$, use the formula:

$$\text{Normalized } \mathbf{v} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$$

Calculate the norm $\|\mathbf{v}\|$:

$$\|\mathbf{v}\| = \sqrt{24.57^2 + 22.67^2} = \sqrt{603.88 + 513.21} = \sqrt{1117.09} \approx 33.45$$

Thus, the normalized eigenvector is:

$$\mathbf{v} = \frac{1}{33.45} \begin{bmatrix} 24.57 \\ 22.67 \end{bmatrix} = \begin{bmatrix} 0.73 \\ 0.67 \end{bmatrix}$$

$$PC1 = \mathbf{v} = \begin{bmatrix} 0.73 \\ 0.67 \end{bmatrix}$$

This is our principal component.

STEP 5: DERIVING THE 1D DATASET USING PRINCIPAL COMPONENT:

Given the principal component:

$$\mathbf{v} = \begin{bmatrix} 0.73 \\ 0.67 \end{bmatrix}$$

The data matrix $\mathbf{X}$ is:

$$\mathbf{X} = \begin{bmatrix} -6 & -2 & 2 & 6 \\ -3.5 & -4.5 & 2.5 & 5.5 \end{bmatrix}$$

To project the data points onto the principal component, calculate the new features as:

$$\mathbf{v}^T \cdot \mathbf{X}$$

Compute the transpose of $\mathbf{v}$:

$$\mathbf{v}^T = \begin{bmatrix} 0.73 & 0.67 \end{bmatrix}$$

Then perform the matrix multiplication:

$$\mathbf{v}^T \cdot \mathbf{X} = \begin{bmatrix} 0.73 & 0.67 \end{bmatrix} \begin{bmatrix} -6 & -2 & 2 & 6 \\ -3.5 & -4.5 & 2.5 & 5.5 \end{bmatrix}$$

Calculating the result:

$$\mathbf{v}^T \cdot \mathbf{X} = \begin{bmatrix} -6.725 & -4.475 & 3.135 & 8.065 \end{bmatrix}$$

Thus, the new features are:

$$\begin{bmatrix} -6.725 & -4.475 & 3.135 & 8.065 \end{bmatrix}$$

So after reducing the dimensions to 1D

Feature new	-6.725	-4.475	3.135	8.065

Table 2: Data After Dimensionality Reduction

Comparing the eigen values from the Data and given from the question

Variance Explained by Each Principal Component from Eigen values given in the question.

Given the eigenvalues $\lambda_1 = 30.3849$ and $\lambda_2 = 6.6151$, we can compare their significance.

The variance explained by each principal component can be calculated as:

$$\text{Variance explained by PC}_1 = \frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{30.3849}{30.3849 + 6.6151} \approx 0.821 \quad --(i)$$

$$\text{Variance explained by PC}_2 = \frac{\lambda_2}{\lambda_1 + \lambda_2} = \frac{6.6151}{30.3849 + 6.6151} \approx 0.179 \quad --(ii)$$

Since λ_1 is significantly larger than λ_2, the first principal component PC_1 explains approximately 82.1% of the total variance, on the other hand PC_2 only explains about 17.9% of the variance.

Variance Explained by Each Principal Component from Eigen values obtained from the data.
Given the eigenvalues $\lambda_1 = 47.57$ and $\lambda_2 = 2.09$, we can compare their significance.
The variance explained by each principal component can be calculated as:

$$\text{Variance explained by PC}_1 = \frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{47.57}{47.57 + 2.09} \approx 0.957 \quad --(iii)$$

$$\text{Variance explained by PC}_2 = \frac{\lambda_2}{\lambda_1 + \lambda_2} = \frac{2.09}{47.57 + 2.09} \approx 0.04 \quad --(iv)$$

Since λ_1 is significantly larger than λ_2, the first principal component PC_1 explains approximately 95.7% of the total variance, on the other hand PC_2 only explains about 4% of the variance.

Conclusion

From the Two sets of Eigen values(given and obtained from the data) the principal component PC_1 obtained from the data should be selected is more significant as it captures the majority of the variance from the dataset as compared to the given values (comparing equation 'i' and 'iii').
By focusing on this principal component, we can reduce the dimensionality of the data while retaining the most important information from the dataset.

Q2)

In this problem, you will perform K-means clustering manually, with K = 2, on a small example with n = 6 observations and p = 2 features. The observations are as (14),(13),(04),(51),(62),(40).

1. Plot the observations.
2. Randomly assign a cluster label to each observation. Report the cluster labels for each observation.
3. Compute the centroid for each cluster.
4. Assign each observation to the centroid to which it is closest, in terms of Euclidean distance. Report the cluster labels for each observation.
5. Repeat (c) and (d) until the answers obtained stop changing
6. In your plot from (a), color the observations according to the cluster labels obtained

Answer 2

Given:

- Number of clusters: $k = 2$
- Number of data points: $n = 6$
- Number of features: $p = 2$

The given points are:

$$\mathbf{A} = (1, 4), \quad \mathbf{B} = (1, 3), \quad \mathbf{C} = (0, 4), \quad \mathbf{D} = (5, 1), \quad \mathbf{E} = (6, 2), \quad \mathbf{F} = (4, 0)$$

Let the two clusters be $\mathbf{C}_1$ and $\mathbf{C}_2$, and their centroids be represented by k_1 and k_2.

The data points after plotting:

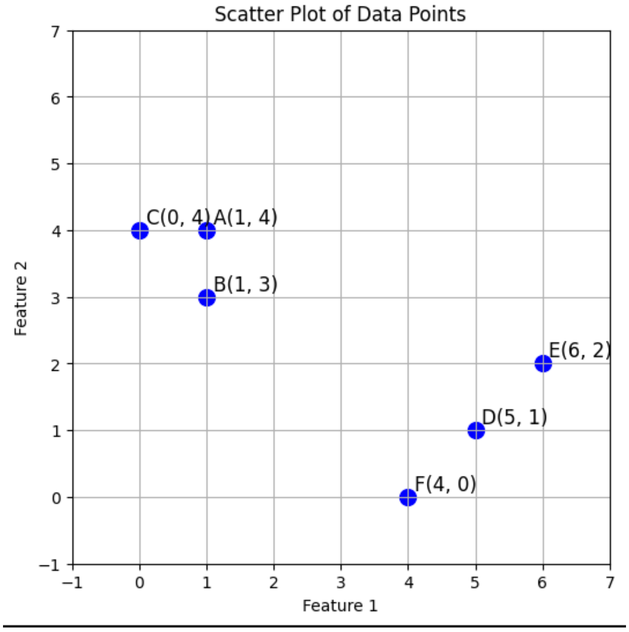

Figure 1: Scatter Plot Of Given Data Points

Applying the K-means Algorithm

Step 1: Randomly Assigning Datapoints to clusters:
Let the two clusters be:

$$C_1 = \{A, C\}$$
$$C_2 = \{B, D, E, F\}$$

Step 2: Calculating Centroids:

$$\text{Calculate the centroid } k_1 \text{ for } C_1 :$$
$$k_1 = \left(\frac{1+0}{2}, \frac{4+4}{2}\right) = \left(\frac{1}{2}, 4\right) = (0.5, 4)$$

$$\text{Calculate the centroid } k_2 \text{ for } C_2 :$$
$$k_2 = \left(\frac{1+5+6+4}{4}, \frac{3+1+2+0}{4}\right)$$
$$k_2 = \left(\frac{16}{4}, \frac{6}{4}\right) = (4, 1.5)$$

Thus, the new centroids are:

$$k_1 = (0.5, 4), \quad k_2 = (4, 1.5)$$

STEP 3: Assigning Points to the nearest centroids:

To find which cluster is closest to the given point. we will calculate the Euclidean Distance. The Distance between Two points $\mathbf{X} = (x_1, y_1)$ and $\mathbf{Y} = (x_2, y_2)$

$$d(\mathbf{X}, \mathbf{Y}) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The Distance of Points from the centroids (k1,k2) using the euclidean distance is

$$\textbf{Distances from the new centroid } \mathbf{k_1} = (\mathbf{0.5}, \mathbf{4}):$$

$$d_{A,k_1} = \sqrt{(1 - 0.5)^2 + (4 - 4)^2} = \sqrt{(0.5)^2 + 0^2} = 0.5$$

$$d_{B,k_1} = \sqrt{(1 - 0.5)^2 + (3 - 4)^2} = \sqrt{(0.5)^2 + (-1)^2} = \sqrt{0.25 + 1} = \sqrt{1.25} \approx 1.118$$

$$d_{C,k_1} = \sqrt{(0 - 0.5)^2 + (4 - 4)^2} = \sqrt{(-0.5)^2 + 0^2} = 0.5$$

$$d_{D,k_1} = \sqrt{(5 - 0.5)^2 + (1 - 4)^2} = \sqrt{(4.5)^2 + (-3)^2} = \sqrt{20.25 + 9} = \sqrt{29.25} \approx 5.408$$

$$d_{E,k_1} = \sqrt{(6 - 0.5)^2 + (2 - 4)^2} = \sqrt{(5.5)^2 + (-2)^2} = \sqrt{30.25 + 4} = \sqrt{34.25} \approx 5.852$$

$$d_{F,k_1} = \sqrt{(4 - 0.5)^2 + (0 - 4)^2} = \sqrt{(3.5)^2 + (-4)^2} = \sqrt{12.25 + 16} = \sqrt{28.25} \approx 5.317$$

$$\textbf{Distances from the new centroid } \mathbf{k_2} = (\mathbf{4}, \mathbf{1.5}):$$

$$d_{A,k_2} = \sqrt{(1 - 4)^2 + (4 - 1.5)^2} = \sqrt{(-3)^2 + 2.5^2} = \sqrt{9 + 6.25} = \sqrt{15.25} \approx 3.905$$

$$d_{B,k_2} = \sqrt{(1 - 4)^2 + (3 - 1.5)^2} = \sqrt{(-3)^2 + 1.5^2} = \sqrt{9 + 2.25} = \sqrt{11.25} \approx 3.354$$

$$d_{C,k_2} = \sqrt{(0-4)^2 + (4-1.5)^2} = \sqrt{(-4)^2 + 2.5^2} = \sqrt{16+6.25} = \sqrt{22.25} \approx 4.717$$

$$d_{D,k_2} = \sqrt{(5-4)^2 + (1-1.5)^2} = \sqrt{(1)^2 + (-0.5)^2} = \sqrt{1+0.25} = \sqrt{1.25} \approx 1.118$$

$$d_{E,k_2} = \sqrt{(6-4)^2 + (2-1.5)^2} = \sqrt{(2)^2 + 0.5^2} = \sqrt{4+0.25} = \sqrt{4.25} \approx 2.061$$

$$d_{F,k_2} = \sqrt{(4-4)^2 + (0-1.5)^2} = \sqrt{0^2 + (-1.5)^2} = \sqrt{2.25} = 1.5$$

points	k1	k2
A	0.5	3.9
B	1.11	3.35
C	0.5	4.7
D	5.4	1.11
E	5.8	2.06
F	5.3	1.5

Table 3: Distances of points from Centroids

Updating Cluster and Centroids

Based on the distances of the points from the centroids the updated clusters are:

$$C_1 = \{A, B, C\}$$
$$C_2 = \{D, E, F\}$$

This was our **Iteration 1**.

Iteration 2:

Calculating the new centroids:

Calculate the new centroid $\mathbf{k_1}$ for $\mathbf{C_1}$:

$$k_1 = \left(\frac{1+1+0}{3}, \frac{4+3+4}{3}\right) = \left(\frac{2}{3}, \frac{11}{3}\right)$$

Calculate the new centroid $\mathbf{k_2}$ for $\mathbf{C_2}$:

$$k_2 = \left(\frac{5+6+4}{3}, \frac{1+2+0}{3}\right)$$

$$k_2 = \left(\frac{15}{3}, \frac{3}{3}\right) = (5, 1)$$

Thus, the new centroids are:

$$k_1 = (0.67, 3.67), \quad k_2 = (5, 1)$$

Calculate the distance of points from new cluster centroid

Distances from the new centroid $\mathbf{k_1} = (\mathbf{0.67}, \mathbf{3.67})$:

$$\begin{aligned}
d_{A,k_1} &= \sqrt{(1-0.67)^2 + (4-3.67)^2} = \sqrt{(0.33)^2 + (0.33)^2}\sqrt{0.2178} \approx 0.467 \\
d_{B,k_1} &= \sqrt{(1-0.67)^2 + (3-3.67)^2} = \sqrt{(0.33)^2 + (-0.67)^2} = \sqrt{0.5578} \approx 0.746 \\
d_{C,k_1} &= \sqrt{(0-0.67)^2 + (4-3.67)^2} = \sqrt{(-0.67)^2 + (0.33)^2} = \sqrt{0.5578} \approx 0.746 \\
d_{D,k_1} &= \sqrt{(5-0.67)^2 + (1-3.67)^2} = \sqrt{(4.33)^2 + (-2.67)^2} = \sqrt{25.8778} \approx 5.088 \\
d_{E,k_1} &= \sqrt{(6-0.67)^2 + (2-3.67)^2} = \sqrt{(5.33)^2 + (-1.67)^2} = \sqrt{31.1978} \approx 5.585 \\
d_{F,k_1} &= \sqrt{(4-0.67)^2 + (0-3.67)^2} = \sqrt{(3.33)^2 + (-3.67)^2} = \sqrt{24.5578} \approx 4.955
\end{aligned}$$

Distances from the new centroid $\mathbf{k_2} = (\mathbf{5}, \mathbf{1})$:

$$\begin{aligned}
d_{A,k_2} &= \sqrt{(1-5)^2 + (4-1)^2} = \sqrt{(-4)^2 + 3^2} = \sqrt{16+9} = \sqrt{25} = 5 \\
d_{B,k_2} &= \sqrt{(1-5)^2 + (3-1)^2} = \sqrt{(-4)^2 + 2^2} = \sqrt{16+4} = \sqrt{20} \approx 4.472 \\
d_{C,k_2} &= \sqrt{(0-5)^2 + (4-1)^2} = \sqrt{(-5)^2 + 3^2} = \sqrt{25+9} = \sqrt{34} \approx 5.831 \\
d_{D,k_2} &= \sqrt{(5-5)^2 + (1-1)^2} = \sqrt{0^2 + 0^2} = 0 \\
d_{E,k_2} &= \sqrt{(6-5)^2 + (2-1)^2} = \sqrt{(1)^2 + 1^2} = \sqrt{1+1} = \sqrt{2} \approx 1.414 \\
d_{F,k_2} &= \sqrt{(4-5)^2 + (0-1)^2} = \sqrt{(-1)^2 + (-1)^2} = \sqrt{1+1} = \sqrt{2} \approx 1.414
\end{aligned}$$

points	k1	k2
A	0.46	5
B	0.74	4.47
C	0.74	5.83
D	5.08	0
E	5.59	1.41
F	4.95	1.41

Table 4: Distances of points from Centroids

Updating Cluster and Centroids

Based on the distances of the points from the centroids the updated clusters are:

$$C_1 = \{A, B, C\}$$

$$C_2 = \{D, E, F\}$$

This was our **Iteration 2**.

Since the clusters have **not changed**, i.e. there is no reassignment of points to different cluster. we can conclude that our clustering process has come to an end. Therfore, we will stop the process.

The Final Clusters Are:

$$\mathbf{C_1 = \{A, B, C\}}$$

$$\mathbf{C_2 = \{D, E, F\}}$$

Final Plot with Clusters:

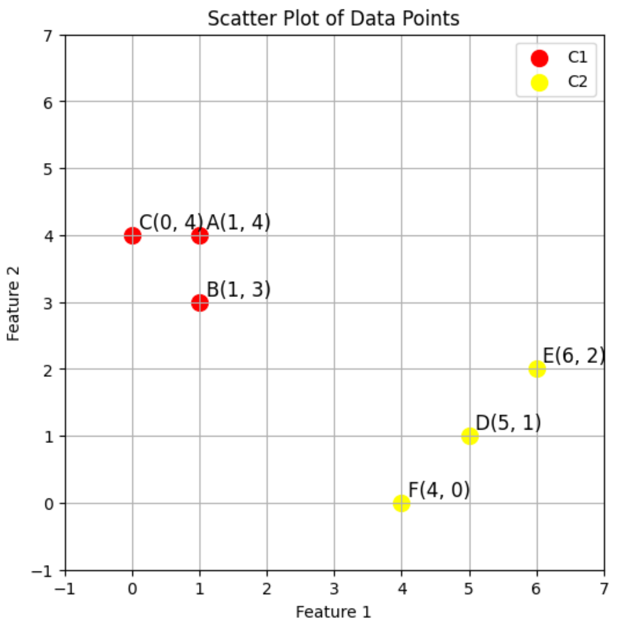

Figure 2: Data Points in cluster

Q3)

Consider a 2-D dataset having two types of classes of data points namely X1 and X2. Given, X1 = (x1, x2) = (4, 1),(2, 4),(2, 3),(3, 6),(4, 4) and X2 = (x1, x2) = (9, 10),(6, 8),(9, 5),(8, 7),(10, 8).

1. Apply Linear Discriminant Analysis in the view of dimensionality reduction.
2. Plot the graphs if required
3. Write advantages, disadvantages and applications of Linear Discriminant Analysis(LDA).

Answer 3

The given Data is:

Class	x1	x2
X1	4	1
X1	2	4
X1	2	3
X1	3	6
X1	4	4
X2	9	10
X2	6	8
X2	9	5
X2	8	7
X2	10	8

Table 5: Caption

For a better visualization of the Data:

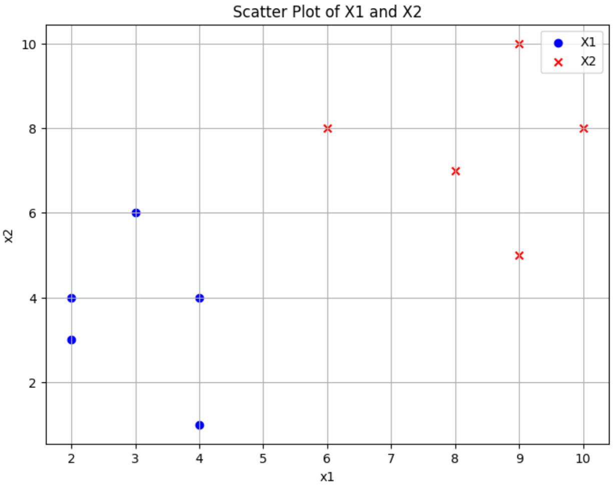

Figure 3: Data Points in X1 and X2

PERFORMING LDA FOR DIMENSIONALITY REDUCTION:

STEP 1: Calculation the mean vectors:

For Class $X1$

Given the data points:

$$X1 = \{(4,1),(2,4),(2,3),(3,6),(4,4)\}$$

Compute the mean vector μ_1:

$$\mu_1 = \left(\frac{1}{n_1}\sum_{i=1}^{n_1} x_{1i}, \frac{1}{n_1}\sum_{i=1}^{n_1} x_{2i} \right)$$

where $n_1 = 5$.
Calculate μ_{1x_1}:

$$\mu_{1x_1} = \frac{4+2+2+3+4}{5} = \frac{15}{5} = 3$$

Calculate μ_{1x_2}:

$$\mu_{1x_2} = \frac{1+4+3+6+4}{5} = \frac{18}{5} = 3.6$$

Thus, the mean vector for $X1$ is:

$$\mu_1 = \begin{pmatrix} 3 \\ 3.6 \end{pmatrix}$$

For Class $X2$

Given the data points:

$$X2 = \{(9, 10), (6, 8), (9, 5), (8, 7), (10, 8)\}$$

Compute the mean vector μ_2:

$$\mu_2 = \left(\frac{1}{n_2} \sum_{i=1}^{n_2} x_{1i}, \frac{1}{n_2} \sum_{i=1}^{n_2} x_{2i} \right)$$

where $n_2 = 5$.
Calculate μ_{2x}:

$$\mu_{2x} = \frac{9 + 6 + 9 + 8 + 10}{5} = \frac{42}{5} = 8.4$$

Calculate μ_{2x_2}:

$$\mu_{2x_2} = \frac{10 + 8 + 5 + 7 + 8}{5} = \frac{38}{5} = 7.6$$

Thus, the mean vector for $X2$ is:

$$\mu_2 = \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}$$

STEP 2: CALCULATE THE WITHIN-ClASS SCATTER MATRIX

The scatter Matrices are given by the formula:

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

The within-class scatter matrix S_w is given by:

$$S_w = \sum_{i \in no_classes} S_i$$

Calculations for $\mathbf{S_1}$:

$$X_1 = \begin{bmatrix} 4 & 2 & 2 & 3 & 6 \\ 1 & 4 & 3 & 6 & 4 \end{bmatrix}$$

Therefore;

$$X_1 - \mu_1 = \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4 \end{bmatrix}$$

To find the covariance matrix S1,

$$S_1 = \frac{1}{n}.\left[X_1 - \mu_1\right]\left[X_1 - \mu_1\right]^T$$

here n=5 (for number of datapoints)

$$S_1 = \frac{1}{5}\begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4 \end{bmatrix} * \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4 \end{bmatrix}^T$$

$$S_1 = \frac{1}{5}\begin{bmatrix} 4 & -2 \\ -2 & 13.2 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.6 \end{bmatrix}$$

Calculations for $\mathbf{S_2}$:

$$X_2 = \begin{bmatrix} 9 & 6 & 9 & 8 & 10 \\ 10 & 8 & 5 & 7 & 8 \end{bmatrix}$$

Therefore;

$$X_2 - \mu_2 = \begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix}$$

To find the covariance matrix S2,

$$S_2 = \frac{1}{n}.\left[X_2 - \mu_2\right]\left[X_2 - \mu_2\right]^T$$

here n=5 (for number of datapoints)

$$S_2 = \frac{1}{5}\begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix}\begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix}^T$$

$$S_2 = \frac{1}{5}\begin{bmatrix} 9.2 & -0.2 \\ -0.2 & 13.2 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.6 \end{bmatrix}$$

$\mathbf{S_w}$ is given as

$$S_w = S_1 + S_2$$

so,

$$S_w = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.2 \end{bmatrix}$$

The between-class scatter matrix S_B is given by:

$$S_B = (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T$$

where μ_1 and μ_2 are the mean vectors of the two classes.

$$\mu_1 = \begin{bmatrix} 3 \\ 3.6 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix}$$

$$\mu_2 - \mu_1 = \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix} - \begin{bmatrix} 3 \\ 3.6 \end{bmatrix} = \begin{bmatrix} 8.4 - 3 \\ 7.6 - 3.6 \end{bmatrix} = \begin{bmatrix} 5.4 \\ 4 \end{bmatrix}$$

$$S_B = \begin{bmatrix} 5.4 \\ 4 \end{bmatrix} \begin{bmatrix} 5.4 & 4 \end{bmatrix}$$

$$S_B = \begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16 \end{bmatrix}$$

Finding the best LDA projection.

We find this using Eigen Vectors having largest Eigen Value

$$S_w^{-1} S_B v = \lambda V$$

Inverse of S_w is:

$$S_w^{-1} = \begin{bmatrix} 0.38 & 0.032 \\ 0.03 & 0.19 \end{bmatrix}$$

and

$$S_w^{-1} * S_B = \begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 2.67 \end{bmatrix}$$

finding eigen values:

$$det(S_w^- 1 S_B v - \lambda I) = 0$$

$$\left| \begin{bmatrix} 11.89 - \lambda & 8.81 \\ 5.08 & 2.76 - \lambda \end{bmatrix} \right| = 0$$

the characteristic equation is

$$\lambda^2 - 14.65\lambda - 11.9384 = 0$$

Solving for λ, $\quad \lambda_1 = 15.65, \lambda_2 = -0.77$
we will use λ_1 as it will capture variance better.
Substituting λ_1 to get eigen vectors:

$$\begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.76 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = 16.65 \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

We get

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 0.91 \\ 0.39 \end{bmatrix}$$

Reducing 2-D data to 1D: We combine X_1 and X_2 into one matrix D:

$$D = \begin{bmatrix} 4 & 1 \\ 2 & 4 \\ 2 & 3 \\ 3 & 6 \\ 4 & 4 \\ 9 & 10 \\ 6 & 8 \\ 9 & 5 \\ 8 & 7 \\ 10 & 8 \end{bmatrix}$$

Next, we perform the projection by calculating DV:

$$V = \begin{bmatrix} 0.91 \\ 0.39 \end{bmatrix}$$

Now, calculate DV:

$$DV = \begin{bmatrix} 4 & 1 \\ 2 & 4 \\ 2 & 3 \\ 3 & 6 \\ 4 & 4 \\ 9 & 10 \\ 6 & 8 \\ 9 & 5 \\ 8 & 7 \\ 10 & 8 \end{bmatrix} \begin{bmatrix} 0.91 \\ 0.39 \end{bmatrix}$$

Simplifying each row:

$$DV = \begin{bmatrix} 4.03 \\ 3.38 \\ 2.99 \\ 5.07 \\ 5.2 \\ 12.09 \\ 8.58 \\ 10.14 \\ 10.01 \\ 12.22 \end{bmatrix}$$

Class	New Feature
X1	4.03
X1	3.38
X1	2.99
X1	5.07
X1	5.2
X2	12.09
X2	8.58
X2	10.14
X2	10.01
X2	12.22

Table 6: Reduced Data

Finally:

Therefore the data is now reduced to 1 dimension

visualising projection of this data onto the vector:

Since the Points are now in 1-D we can represent them on a line, I am plotting the new points and their projections:

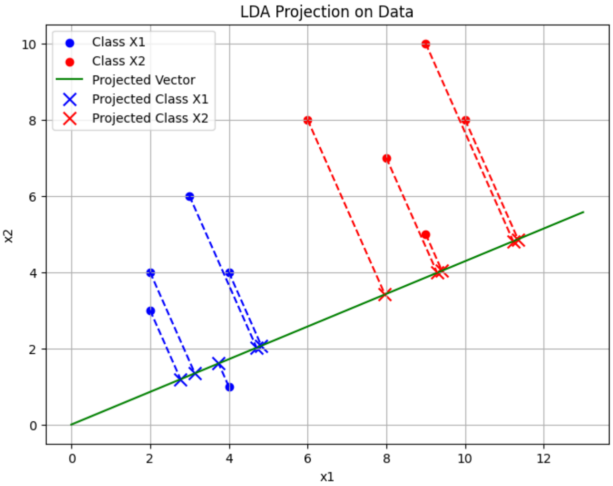

Figure 4: LDA Projection of X1 and X2

The same points can be viewed in 1-D as along the X-axis, and will look like:

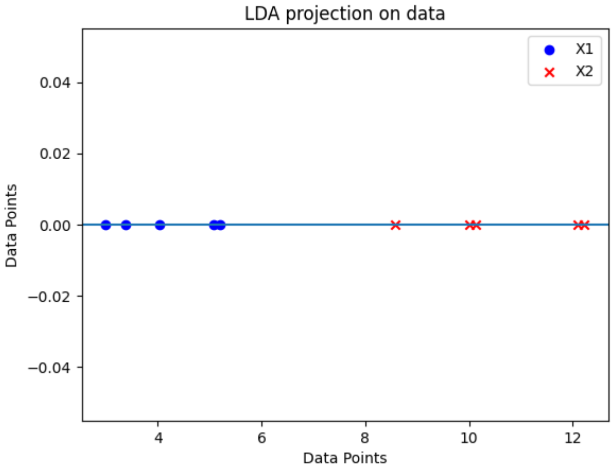

Figure 5: New Points along X-axis

As you can see we can still seperate the two cluster distinctly. So the class separation is maintained even after reduction of dimensions.

advantages, disadvantages and applications of Linear Discriminant Analysis(LDA).

Advantages:

1. **Dimensionality Reduction:** Lda is effective for reducing dimensions of the data while preserving class information.
2. **Class Sepearation:** LDA ensures that the distances between the means of different classes is maximum, and variance is minimum. which help in classification algorithm.
3. **Efficiency:** For smaller datasets, LDA is computationally efficients and it uses matrix operations.
4. **Robust ot noise:** It is robust to noise and irrelevant features as it tries to find the directions that maximizes class separabilty.

Disadvantages:

1. **Linearity Assumption:**LDA assumes a linear relationship between features, which if not the case, results in poor performance.
2. **Not Suitable for High-dimensional data:** LDA performance might degrade, in high-dimensional spaces.

3. **Equal covariance matrices assumption:** LDA assumes that all classes must share the same covariance matrix. If this assumption is violated, the performance of LDA might degrade

Applications:

1. **Face Recognition:** It widely used in face recognition for reducing dimensions while preserving class separbility
2. **Medical Diagnosis:** LDA can be used to classigy patients based on some diagnostic festures.
3. **Text Classification:** LDA can be applied to text classification tasks such as topic modelling.
4. **Market and Customer Segmentation:** LDA can be used to segment the customers into different groups.

Q4)

For the Wine Quality Data Set, convert all the values in the quality attribute to 0 (bad) if the value is less than or equal to 6 and to 1 (good) otherwise. Normalize all the other attributes between 0 and 1 by min-max scaling. Mention why we use min-max scaling

Answer 4

The coding problem is solved, to view please click the below link.

Wine Quality Dataset Problem.

Min-Max Scaling is a pre-processing technique which is used to change an attribute's values in a dataset tare set within a specified range, typically between [0,1] or [-1,1]. the scaled value is given by the formula:

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

where:

- **X** is the original value.
- $\mathbf{X_{min}}$ is the smallest value in the dataset for the feature
- $\mathbf{X_{max}}$ is the largest value in the dataset for the feature

Some key reasons for using min-max scaling are:

1. **Feature Standardization:** It is possible for different features in the datasets to have varying magnitudes and unit scales. We can bring all features to a common scale without changing the distribution by utilizing min-max scaling. This stops larger magnitude characteristics from controlling the learning process.
2. **Improved Model Performance:** Many learning algorithms are sensitive to the scale of the data. Example:- KNN,SVM,Neural networks e.t.c. For such algorithms, features with larger magnitude and scale can dominate the cost function and eventually affect the model's performance. Scaling ensures, the features equally contribute to the result

3. **Handling Non-Gaussian Distributed Data:** For a skewed dataset, or one which does not follow a normal distribution. min-max scaling is a straightforward and efficient method for normalizing a feature range without the need for the assumption of a specific data distribution.

4. **Facilitates Comparisons Across Datasets:** When we are combining or comparing data from different sources, with varying scales, min-max scaling allows us to standardize these features, enabling consistent comparison.

5. **Improves Model Convergence:** Scaled features allows a accelerated convergence of gradient descent. which in turn helps in faster training and better accuracy of the models.

Q5)

What is the difference between model parameters and model hyperparameters? What is meant by hyperparameter tuning? Name some common hyperparameters used in clustering algorithms.

Answer 5

Model Parameters: model parameters are those variables that the model learns from the training process. They are responsible to make predictions, thereby influences the error and the loss functions.
During the learning process the model updates these parameters on its own and no intervention is needed. The model accuracy is based on these parameters.
During the learning process the model updates these parameters and no intervention is needed. The model accuracy is based on these parameters.
Model HyperParameters: model hyperparameters are those variables and setting that control the learning process. They are initialised before the learning process starts.
These values do not get updated automatically and have to be set either manually or through some optimisation technique to get the best values of hyperparameters

In conclusion, **Model Parameters** are those values learned by a learning algorithm during a training process, which it uses to makes predictions, and are **automatically adjusted** during the training.
on the other hand, **Model Hyperparameters** are those values that control the behaviour of the **learning process** in itself, and are set before the training process starts.

HyperParameter Tuning:
Hyperparameters are those that let you control the learning process. Hyperparameters tuning is the process of finding the best set of hyperparameters that will give the best performance. This process enhances the model's performance and learning speed. It also ensures that the model is generalised and is not sensitive to the training data.

Some common hyperparameters used in clustering algorithm:

- **No. of Cluster :** The variable represents the number of clusters that will be formed from the dataset.

- **Distance metric:** Used to define, what method to use to calculate the distance between datapoint(e.g. Euclidean, cosine e.t.c).
- **Maximum Iterations:** defines the maximum number of iterations that the algorithm is allowed to have
- **convergence threshold:** The algorithm uses a threshold for convergence to stop the process when the loss is below the value.
- **Epsilon:** used in DBscan, it is the maximum distance between two points to consider them as neighbours

Q6)

You are given three observed signals x$_1$(t), x$_2$(t), and x$_3$(t) which are linear mixtures of three independent source signals s$_1$(t), s$_2$(t), and s$_3$(t). The mixing process is represented by the following matrix equation:
$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix}, \quad s(t) = \begin{pmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

where A is an unknown 3 × 3 mixing matrix, and s(t) represents the independent source signals.

-)Explain the steps involved in estimating the mixing matrix A and the independent source signals s(t) using ICA.
- Suppose the mixing matrix A is not full rank (i.e., it is singular or nearly singular). How would this affect the ICA process? Discuss the potential challenges and the implications for recovering the original source signals.

Answer 6

The mixing matrix A, Source signals s(t) can be estimated from the observed signals x(t),Independent Component Analysis (ICA). ICA is a computational technique for separating a multivariate signal into additive, independent non-Gaussian components. In our case, we want to find the original source signals from their observed mixtures. Below are the steps involved in the ICA process:

1. Centering the Data

First, we need to center the observed data $x(t)$. This means subtracting the mean of each observed signal to ensure that the data has zero mean.

Given the observed signals $x_1(t), x_2(t), x_3(t)$, calculate the mean vector μ:

$$\mu = \frac{1}{T} \sum_{t=1}^{T} x(t)$$

where T is the total number of observations (or time points).
Center the data:

$$\tilde{x}(t) = x(t) - \mu$$

2. Whitening (Sphering) the Data

Whitening transforms the observed signal vector $\tilde{x}(t)$ such that its components are uncorrelated and have unit variance. This step simplifies the ICA process.

- The covariance matrix of the centered data is calculated by:

$$\Sigma = E[\tilde{x}(t)\tilde{x}(t)^T] = \frac{1}{T}\sum_{t=1}^{T}\tilde{x}(t)\tilde{x}(t)^T$$

- Eigenvalue Decomposition on Covariance Matrix :

$$\Sigma = EDE^T$$

  The matrix E represents the eigenvectors, while D is the diagonal matrix of eigenvalues.

- Compute the whitening matrix W:

$$W = D^{-1/2}E^T$$

- Whiten the data:

$$z(t) = W\tilde{x}(t)$$

  Here, $z(t)$ is the whitened signal vector.

3. Estimating the Mixing Matrix (Finding Independent Components)

The next step is to estimate the unmixing matrix W_{ica} that will separate the independent components. We denote the estimated sources by:

$$s(t) = W_{\text{ica}} z(t)$$

some of the algorithms can be used for estimating Wica, such as the Infomax algorithm, FastICA, or the Maximum Likelihood approach. We'll describe the general steps used in a popular algorithm, FastICA:

- **a. Initialize $W_{\textbf{ica}}$:**

  At start a matrix is initialized randomly W_{ica}.

- **b. Choose a Non-Gaussianity Measure:**

  A key principle of ICA is that the source signals are assumed to be non-Gaussian and independent. Non-Gaussianity is often measured using metrics such as kurtosis or negentropy. The goal is to maximize non-Gaussianity of the projected data $y(t) = w^T z(t)$.

  For example, one could use the approximation of negentropy:

  $$J(y) \approx [E\{G(y)\} - E\{G(v)\}]^2$$

  where G is a non-quadratic function (such as $G(y) = y^4$ for kurtosis) and v represents a Gaussian.

- **c. Update the Weight Vector:**

  Iteratively update the weight vector w using the following fixed-point iteration scheme (assuming FastICA):

  $$w^{new} = E\{z(t)g(w^T z(t))\} - E\{g'(w^T z(t))\}w$$

  where g is the derivative of a chosen non-quadratic function (e.g., $g(y) = \tanh(y)$).

- Normalize w^{new}:

  $$w^{\text{new}} = \frac{w^{\text{new}}}{\|w^{\text{new}}\|}$$

- Repeat the update until convergence (i.e., w does not change significantly between iterations).

- **d. Deflationary Orthogonalization:**

  To ensure that the extracted components are independent, apply a deflationary orthogonalization process after each component is extracted:

$$w_k^{\text{new}} = w_k^{\text{new}} - \sum_{i=1}^{k-1} (w_k^{\text{new}} \cdot w_i) w_i$$

  Then, normalize w_k^{new} again.

4. Estimating the Mixing Matrix A

The mixing matrix A is approximated by:

$$A \approx (W_{\text{ica}} W)^{-1}$$

where W is the whitening matrix from Step 2.

5. Estimating the Source Signals

Finally, the estimated source signals $\hat{s}(t)$ are given by:

$$\hat{s}(t) = W_{\text{ica}} z(t)$$

Since $z(t) = W\tilde{x}(t)$, we can substitute to find:

$$\hat{s}(t) = W_{\text{ica}} W \tilde{x}(t)$$

This method allows us to separate the mixed signals $x(t)$ into their independent components $s(t)$ without having the knowledge of the mixing matrix $\mathbf{A}$ in advance, provided that the source signals are statistically independent and non-Gaussian.

Implications of a Non-Full-Rank Mixing Matrix in ICA

When the mixing matrix A is not full rank, meaning $\text{rank}(A) < 3$ (for the case of three sources), several challenges arise for the Independent Component Analysis (ICA) process:

1. **Loss of Information:**
    - A non-full-rank A indicates that the observed signals $x(t)$ do not span the full space of the original source signals $s(t)$. This results in information loss because some of the source signal information is or projected on a lower-dimensional space.
    - Consequently, it becomes impossible to recover all original source signals. Only a lower-dimensional representation of the sources can be reconstructed.
2. **Non-Invertibility of the Mixing Matrix:**
    - ICA requires the calculation $W_{\text{ica}} \approx A^{-1}$ to recover the independent components. If A is singular, A^{-1} does not exist, preventing full recovery of $s(t)$.
    - The absence of an inverse results in ambiguities in separating the source signals, as there could be infinitely many possible source signal sets that generate the observed mixtures.
3. **Challenges in Whitening:**
    - Whitening involves transforming observed data to have uncorrelated components with unit variance by computing the inverse square root of the covariance matrix. A nearly singular A leads to a nearly singular covariance matrix, causing numerical instability during whitening.
    - Instability in whitening can lead to inaccurate transformations, complicating subsequent ICA steps.
4. **Limitation on the Number of Separable Components:**
    - If A has rank r, ICA can only separate r independent components. For example, if $\text{rank}(A) = 4$, only four independent components can be separated, even with more source signals present.
    - This limitation means that not all original signals can be retrieved, and the algorithm may fail to distinguish the full set of independent sources.
5. **Practical Considerations:**
    - A nearly singular A might arise due to physical constraints, sensor failures, or inherent correlations in the data collection process.

It may indicate that fewer independent sources exist than anticipated or that some sources are dependent.

- It is important to verify the rank of the mixing matrix (e.g., using Principal Component Analysis, PCA) and consider preprocessing steps to manage rank deficiencies before applying ICA.

6. **Alternative Approaches:**

   - **Subspace ICA:** This method attempts to find independent components within a lower-dimensional subspace matching the rank of A.
   - **Regularization Techniques:** Applying regularization to handle nearly singular matrices can enhance numerical stability.
   - **Dimensionality Reduction:** Using PCA to reduce data dimensionality before ICA can help focus on the most significant components.