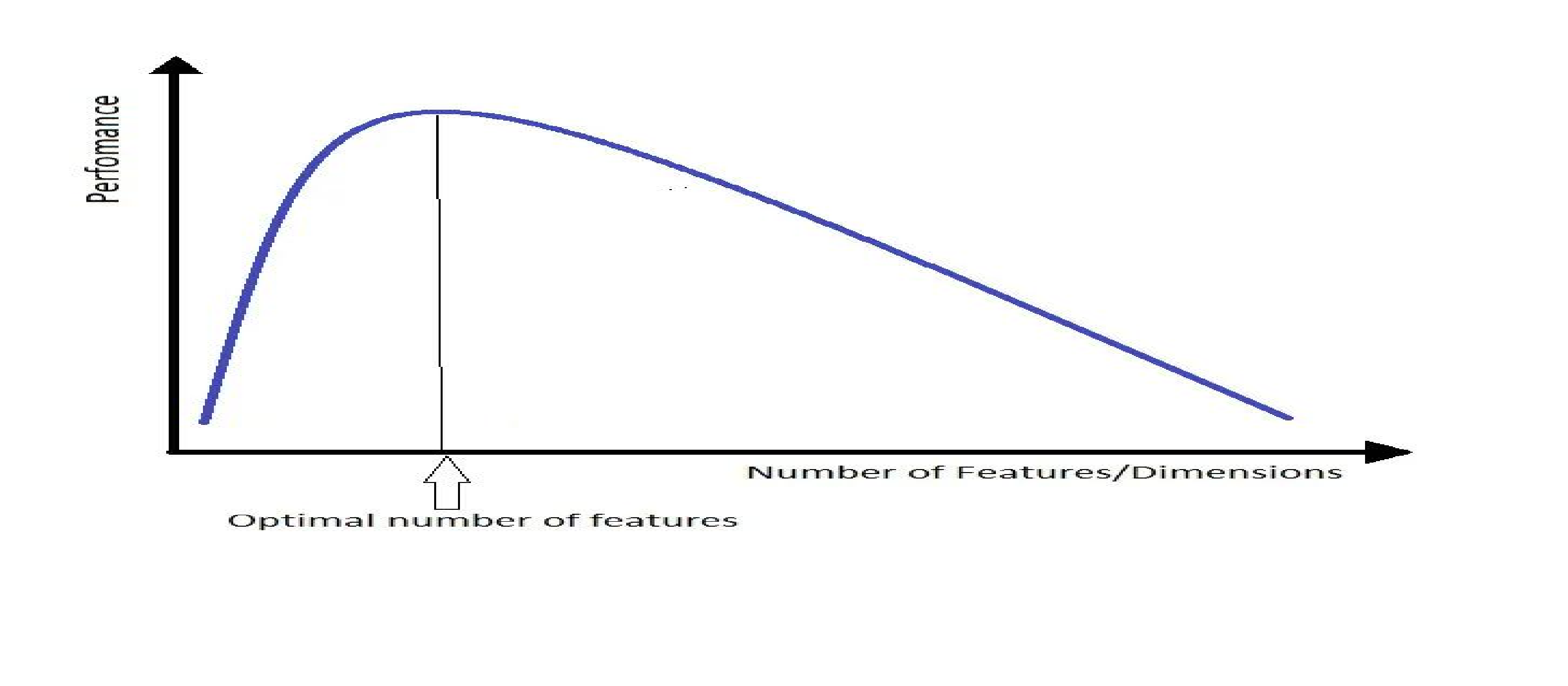
ML
LECTURE-20

BY
Dr. Ramesh Kumar Thakur
Assistant Professor (II)
School Of Computer Engineering

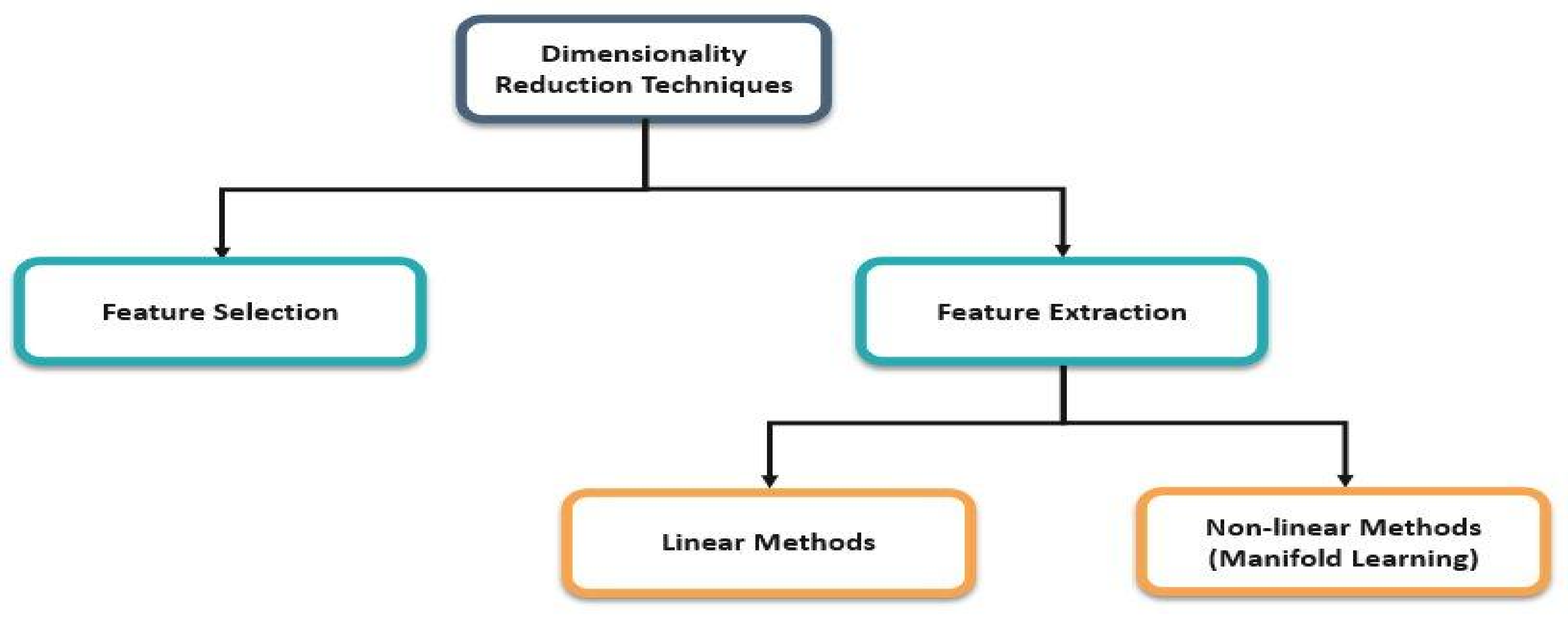
Curse of Dimensionality and Need of Dimension Reduction

- As the **dimensionality increases, the number of data points required for good performance of any machine learning algorithm increases exponentially.**
- The reason is that, we would need more number of data points for any given combination of features, for any machine learning model to be valid.
- Hughes (1968) in his study concluded that **with a fixed number of training samples, the predictive power of any classifier first increases as the number of dimensions increase, but after a certain value of number of dimensions, the performance deteriorates.**
- Thus, the phenomenon of **curse of dimensionality is also known as Hughes phenomenon.**



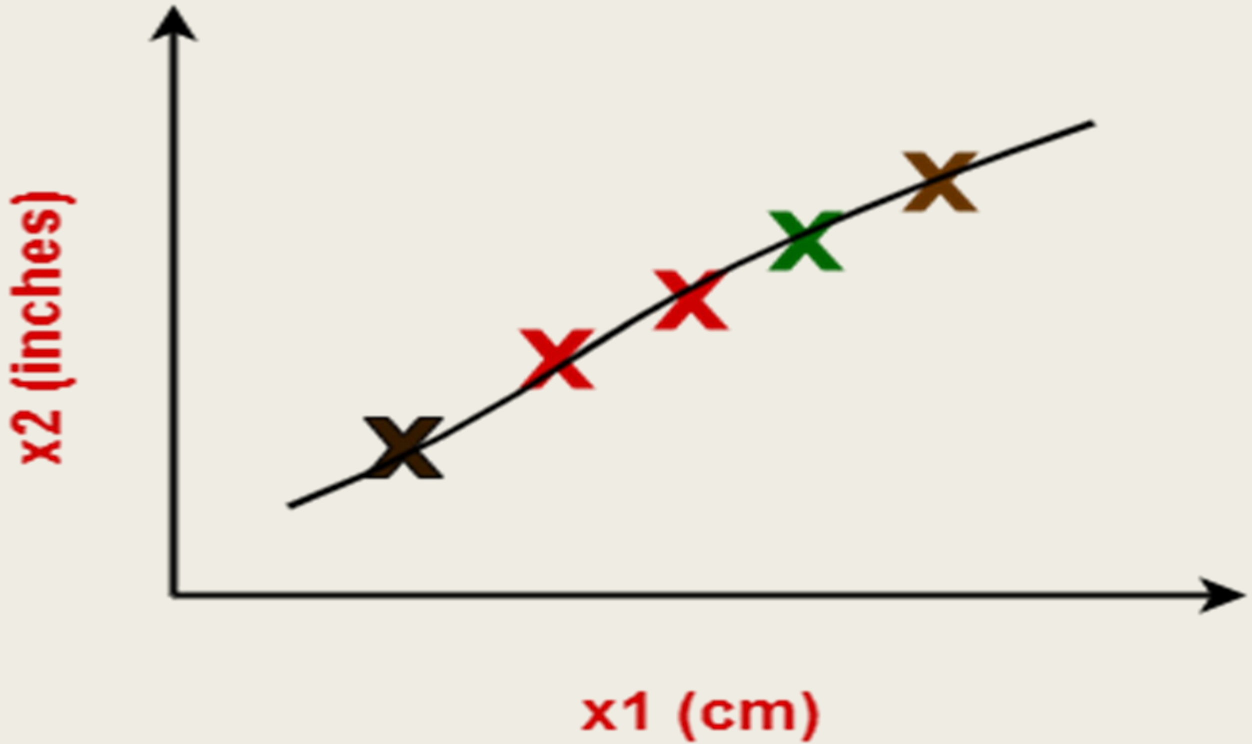
Solution to Curse of Dimensionality

- Several techniques can be employed for dimensionality reduction depending on the problem and the data.
- These techniques are divided into two broad categories:
- **Feature Selection: Choosing the most important features from the data**
- **Feature Extraction: Combining features to create new superfeatures.**



Dimension Reduction

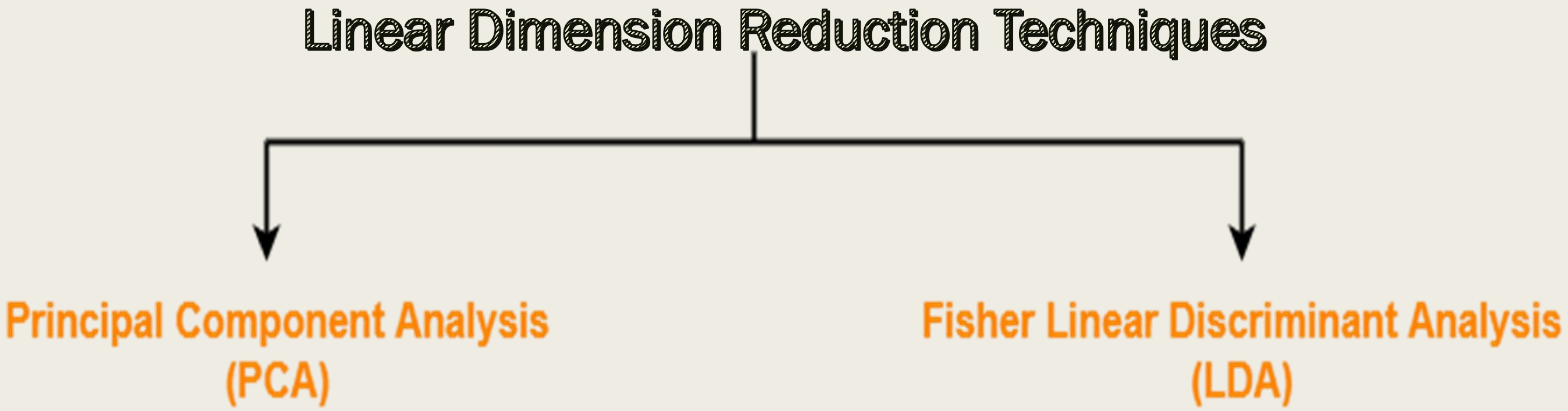
- **It is a process of converting a data set having vast dimensions into a data set with lesser dimensions.**
- It ensures that the converted data set conveys similar information concisely.
- Example-
- The following graph shows two dimensions x1 and x2.
- x1 represents the measurement of several objects in cm.
- x2 represents the measurement of several objects in inches.



Dimension Reduction

- Using both these dimensions convey similar information.
- Also, they introduce a lot of noise in the system.
- So, it is better to use just one dimension.
- Using dimension reduction techniques-
- We convert the dimensions of data from 2 dimensions (x1 and x2) to 1 dimension (z1).
- It makes the data relatively easier to explain.

- **<u>Benefits-</u>**
- It **compresses the data** and thus **reduces the storage space requirements.**
- It **reduces the time required for computation** since less dimensions require less computation.
- It **eliminates the redundant features.**
- It **improves the model performance.**



- **<u>Principal Component Analysis-</u>**
- Principal Component Analysis is a well-known **dimension reduction technique.**
- It **transforms the variables into a new set of variables called as principal components.**
- These principal components are **linear combination of original variables and are orthogonal.**
- The **first principal component accounts for most of the possible variation of original data.**
- The **second principal component does its best to capture the variance in the data.**
- There **can be only two principal components for a two-dimensional data set.**

PCA Algorithm

- **The steps involved in PCA Algorithm are as follows-**
- Step-01: Get data.
- Step-02: Compute the mean vector (μ).
- Step-03: Subtract mean from the given data.
- Step-04: Calculate the covariance matrix.
- Step-05: Calculate the eigen vectors and eigen values of the covariance matrix.
- Step-06: Choosing components and forming a feature vector.
- Step-07: Deriving the new data set.

Example on PCA Algorithm

- **Problem-01:** Given data = { 2, 3, 4, 5, 6, 7; 1, 5, 3, 6, 7, 8 }.
- F1={ 2, 3, 4, 5, 6, 7}
- F2= {1, 5, 3, 6, 7, 8 }.
- Compute the principal component using PCA Algorithm.

OR

- Consider the two dimensional patterns (2, 1), (3, 5), (4, 3), (5, 6), (6, 7), (7, 8).
- Compute the principal component using PCA Algorithm.

OR

- Compute the principal component of following data-

CLASS 1	CLASS 2
X = 2 , 3 , 4	X = 5 , 6 , 7
Y = 1 , 5 , 3	Y = 6 , 7 , 8

OR

- Reduce the following dataset step by step from 2 dimension to 1 using PCA.

Feature	Example1	Example2	Example3	Example4	Example5	Example6
X	2	3	4	5	6	7
Y	1	5	3	6	7	8

Example on PCA Algorithm

- **Solution-**
- We use the above discussed PCA Algorithm-
- **Step-01:**
- **Get data.**
- The given feature vectors are-
- x1 = (2, 1)
- x2 = (3, 5)
- x3 = (4, 3)
- x4 = (5, 6)
- x5 = (6, 7)
- x6 = (7, 8)

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} \begin{bmatrix} 6 \\ 7 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

Example on PCA Algorithm

- **Step-02:**
- **Calculate the mean vector (μ).**
- Mean vector (μ) = ((2 + 3 + 4 + 5 + 6 + 7) / 6, (1 + 5 + 3 + 6 + 7 + 8) / 6) = (4.5, 5)
- Thus, **Mean vector (μ) =** $\begin{bmatrix} 4.5 \\ 5 \end{bmatrix}$

- **Step-03:**
- **Subtract mean vector (μ) from the given feature vectors.**
- x1 – μ = (2 – 4.5, 1 – 5) = (-2.5, -4)
- x2 – μ = (3 – 4.5, 5 – 5) = (-1.5, 0)
- x3 – μ = (4 – 4.5, 3 – 5) = (-0.5, -2)
- x4 – μ = (5 – 4.5, 6 – 5) = (0.5, 1)
- x5 – μ = (6 – 4.5, 7 – 5) = (1.5, 2)
- x6 – μ = (7 – 4.5, 8 – 5) = (2.5, 3)
- Feature vectors (xi) after subtracting mean vector (μ) are- $\begin{bmatrix} -2.5 \\ -4 \end{bmatrix} \begin{bmatrix} -1.5 \\ 0 \end{bmatrix} \begin{bmatrix} -0.5 \\ -2 \end{bmatrix} \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ 2 \end{bmatrix} \begin{bmatrix} 2.5 \\ 3 \end{bmatrix}$

Example on PCA Algorithm

- **Step-04:**
- **Calculate the covariance matrix.**
- Covariance matrix is given by-

$$\text{Covariance Matrix} = \frac{\Sigma(x_i - \mu)(x_i - \mu)^t}{n}$$

$$m_1 = (x_1 - \mu)(x_1 - \mu)^t = \begin{bmatrix} -2.5 \\ -4 \end{bmatrix} \begin{bmatrix} -2.5 & -4 \end{bmatrix} = \begin{bmatrix} 6.25 & 10 \\ 10 & 16 \end{bmatrix}$$

$$m_2 = (x_2 - \mu)(x_2 - \mu)^t = \begin{bmatrix} -1.5 \\ 0 \end{bmatrix} \begin{bmatrix} -1.5 & 0 \end{bmatrix} = \begin{bmatrix} 2.25 & 0 \\ 0 & 0 \end{bmatrix}$$

$$m_3 = (x_3 - \mu)(x_3 - \mu)^t = \begin{bmatrix} -0.5 \\ -2 \end{bmatrix} \begin{bmatrix} -0.5 & -2 \end{bmatrix} = \begin{bmatrix} 0.25 & 1 \\ 1 & 4 \end{bmatrix}$$

$$m_4 = (x_4 - \mu)(x_4 - \mu)^t = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \begin{bmatrix} 0.5 & 1 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$m_5 = (x_5 - \mu)(x_5 - \mu)^t = \begin{bmatrix} 1.5 \\ 2 \end{bmatrix} \begin{bmatrix} 1.5 & 2 \end{bmatrix} = \begin{bmatrix} 2.25 & 3 \\ 3 & 4 \end{bmatrix}$$

$$m_6 = (x_6 - \mu)(x_6 - \mu)^t = \begin{bmatrix} 2.5 \\ 3 \end{bmatrix} \begin{bmatrix} 2.5 & 3 \end{bmatrix} = \begin{bmatrix} 6.25 & 7.5 \\ 7.5 & 9 \end{bmatrix}$$

Example on PCA Algorithm

- Covariance matrix = (m1 + m2 + m3 + m4 + m5 + m6) / 6

$$\text{Covariance Matrix} = \frac{1}{6}\begin{bmatrix} 17.5 & 22 \\ 22 & 34 \end{bmatrix}$$

$$\text{Covariance Matrix} = \begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix}$$

- **Step-05:**
- **Calculate the eigen values and eigen vectors of the covariance matrix.**
- **λ is an eigen value for a matrix M if it is a solution of the characteristic equation $|M - \lambda I| = 0$.**

$$\begin{vmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{vmatrix} - \begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix} = 0$$

$$\begin{vmatrix} 2.92 - \lambda & 3.67 \\ 3.67 & 5.67 - \lambda \end{vmatrix} = 0$$

Example on PCA Algorithm

- $(2.92 - \lambda)(5.67 - \lambda) - (3.67 \times 3.67) = 0$
- $16.56 - 2.92\lambda - 5.67\lambda + \lambda2 - 13.47 = 0$
- $\lambda^2 - 8.59\lambda + 3.09 = 0$
- Solving this quadratic equation, we get $\lambda = 8.22, 0.38$
- Thus, **two eigen values are $\lambda 1 = 8.22$ and $\lambda 2 = 0.38$.**
- Clearly, the **second eigen value is very small compared to the first eigen value.**
- So, the second eigen vector can be left out.
- **Eigen vector corresponds to the greatest eigen value is principal component for the given data set.**
- **So. we find the eigen vector corresponding to eigen value $\lambda 1$.**
- We use the following equation to find the eigen vector-
- **$MX = \lambda X$**
- where-
- M = Covariance Matrix
- X = Eigen vector
- λ = Eigen value

Example on PCA Algorithm

- Substituting the values in the above equation, we get-

$$\begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix} \begin{bmatrix} X1 \\ X2 \end{bmatrix} = 8.22 \begin{bmatrix} X1 \\ X2 \end{bmatrix}$$

- Solving these, we get-
- $2.92X1 + 3.67X2 = 8.22X1$
- $3.67X1 + 5.67X2 = 8.22X2$
- On simplification, we get-
- $5.3X1 = 3.67X2$ ....(1)
- $3.67X1 = 2.55X2$ ....(2)
- $X1/2.55=X2/3.67$
- From (1) and (2), $X1 = 0.69X2$
- From (2), the eigen vector is-

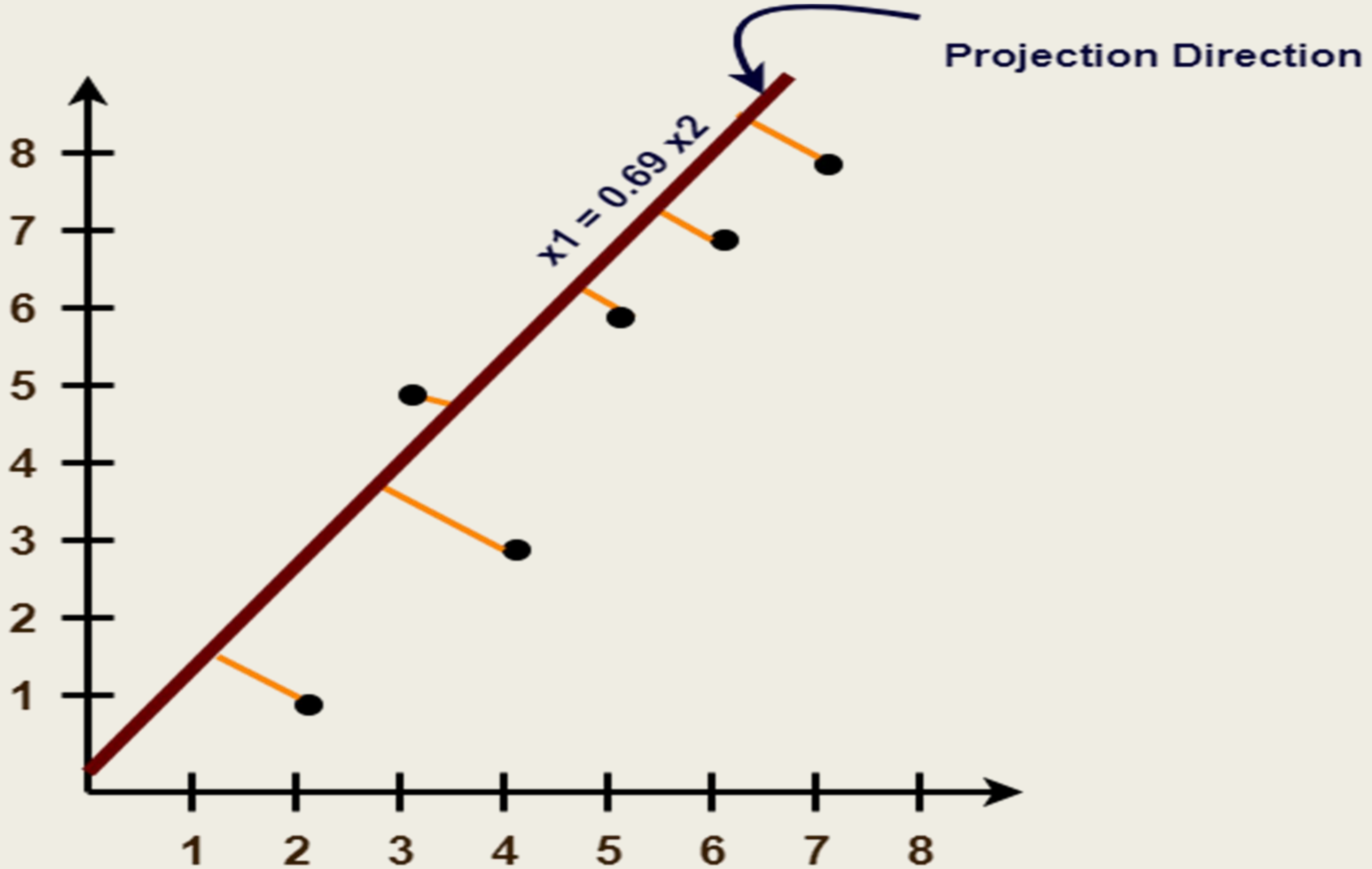
Eigen Vector : $\begin{bmatrix} X1 \\ X2 \end{bmatrix} = \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}$

- Thus, principal component for the given data set is-

Principal Component : $\begin{bmatrix} X1 \\ X2 \end{bmatrix} = \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}$

Example on PCA Algorithm

- Lastly, we project the data points onto the new subspace as-



Example on PCA Algorithm

- **Problem-02:**
- Use PCA Algorithm to transform the pattern (2, 1) onto the eigen vector in the previous question.
- **Solution-**
- The given feature vector is (2, 1).

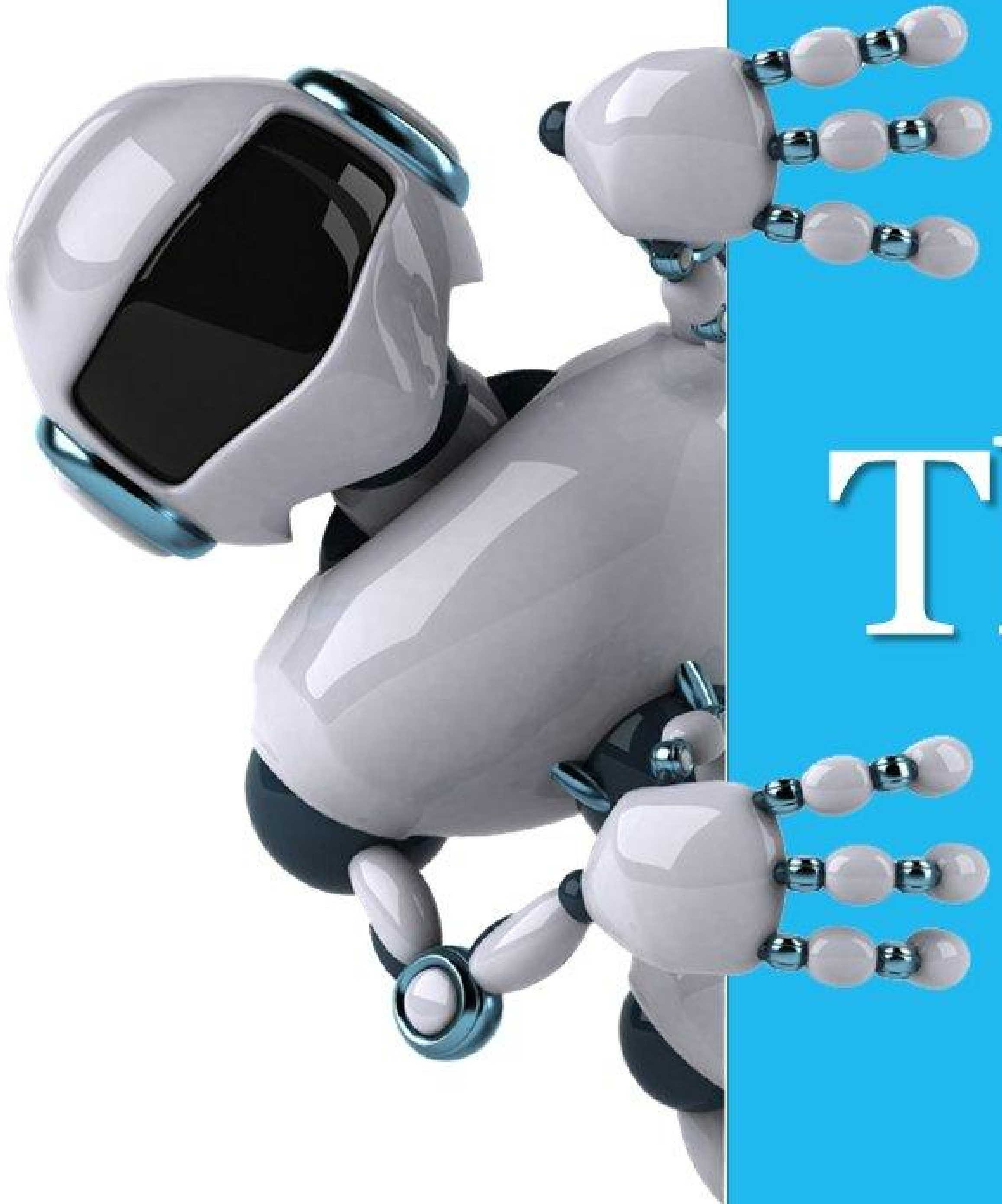
$$\text{Given Feature Vector : } \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

- The feature vector gets transformed to = Transpose of Eigen vector x (Feature Vector – Mean Vector)

$$= \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}^T \times \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 4.5 \\ 5 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 2.55 & 3.67 \end{bmatrix} \times \begin{bmatrix} -2.5 \\ -4 \end{bmatrix}$$

$$= -21.055$$



Thank you