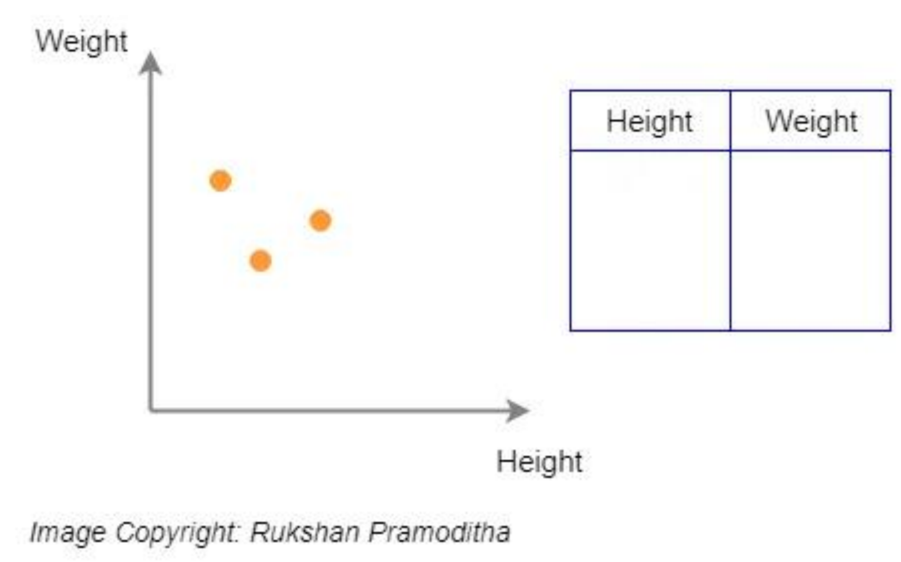
DIMENSIONALITY REDUCTION

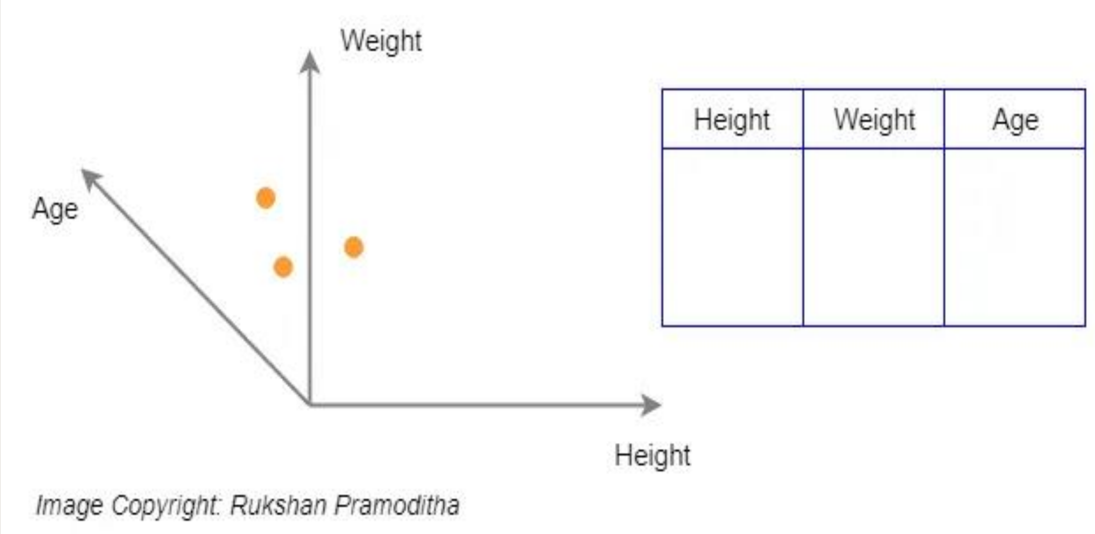
2 DIMENSIONALITY

- the number of attributes, features or input variables of a dataset is referred to as its dimensionality



Height	Weight

Image Copyright: Rukshan Pramoditha



Height	Weight	Age

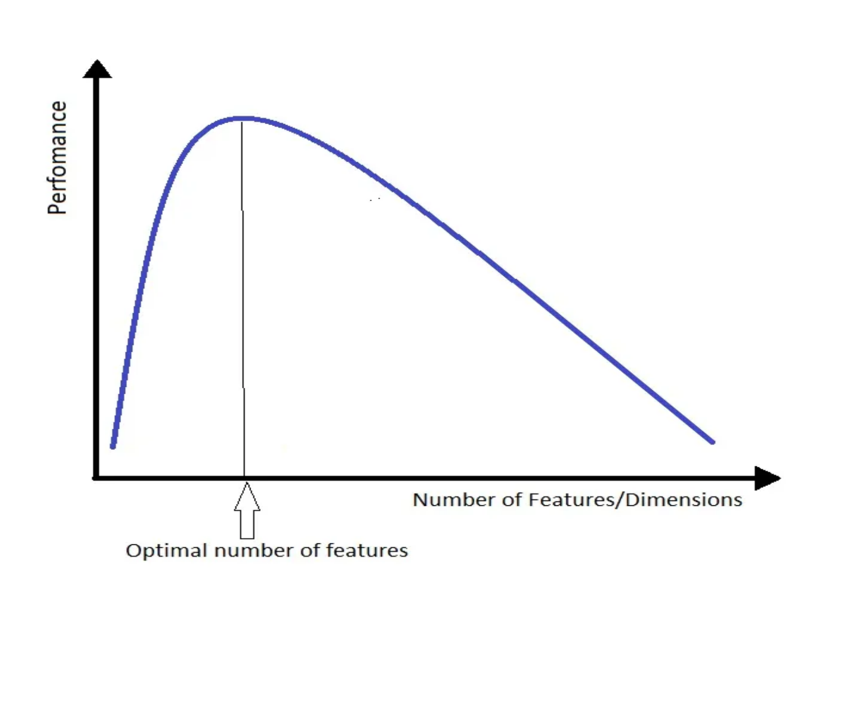
Image Copyright: Rukshan Pramoditha

WHAT IS THE CURSE OF DIMENSIONALITY?

- i) Model_1 consists of only two features say the circuit name and the country name.
- ii) Model_2 consists of 4 features say weather and max speed of the car including the above two.
- iii) Model_3 consists of 8 features say driver's experience, number of wins, car condition, and driver's physical fitness including all the above features.
- iv) Model_4 consists of 16 features say driver's age, latitude, longitude, driver's height, hair color, car color, the car company, and driver's marital status including all the above features.
- v) Model_5 consists of 32 features.
- vi) Model_6 consists of 64 features.
- vii) Model_7 consists of 128 features.

WHAT IS THE CURSE OF DIMENSIONALITY?

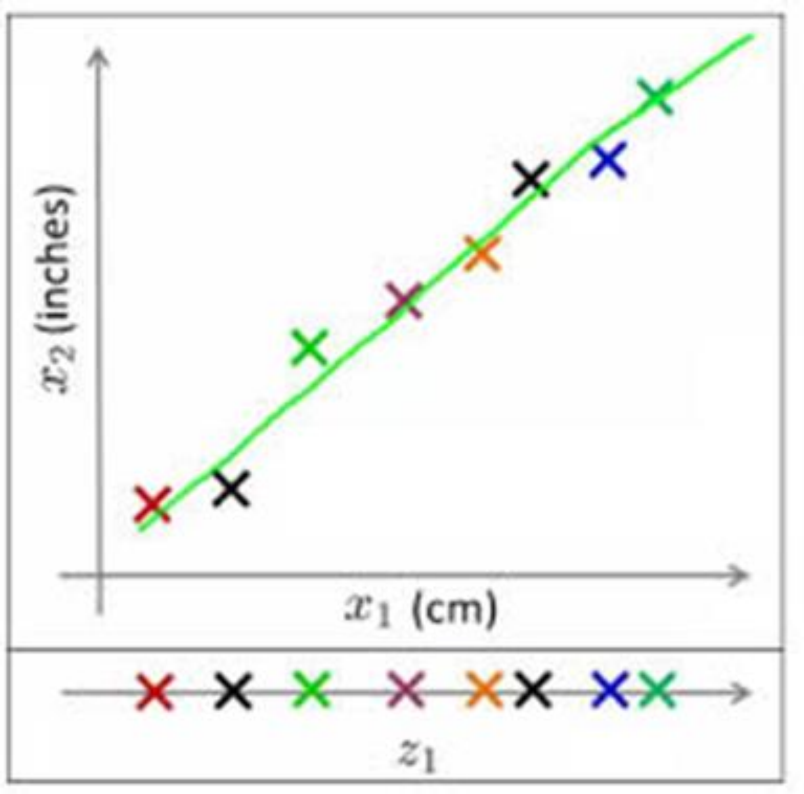
with a fixed number of training samples, the predictive power of any classifier first increases as the number of dimensions increase, but after a certain value of number of dimensions, the **performance deteriorates**. Thus, the phenomenon is known as curse of dimensionality

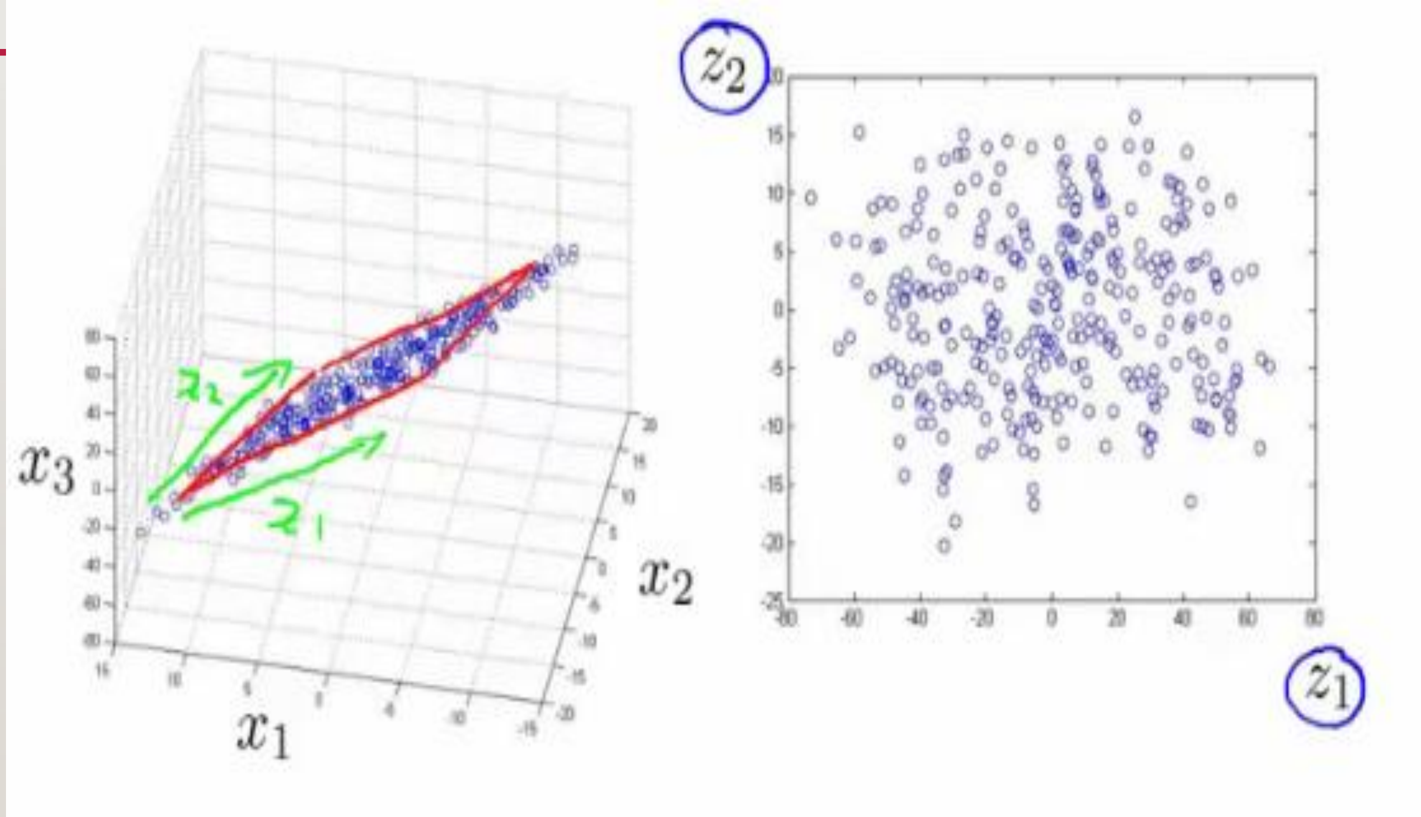


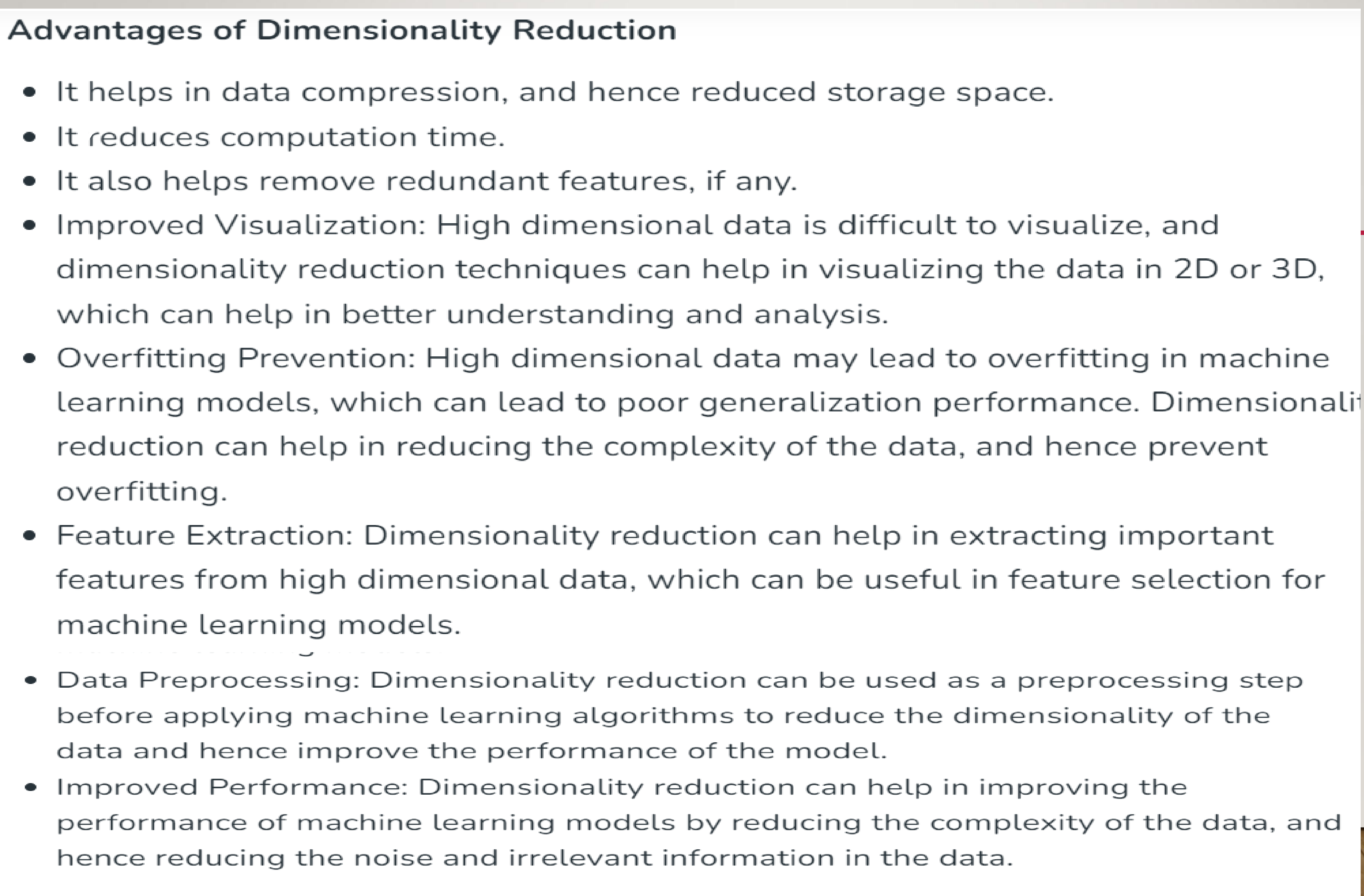
DIMENSIONALITY REDUCTION

- Dimensionality reduction is a process and technique to reduce the number of dimensions -- or features -- in a data set
- The goal of dimensionality reduction is to **decrease the data set's complexity** by reducing the number of features while keeping the most important properties of the original data.

we can reduce n dimensions of data set to k dimensions (k < n) .These k dimensions can be directly identified (filtered) or can be a combination of dimensions (weighted averages of dimensions) or new dimension(s) that represent existing multiple dimensions well.

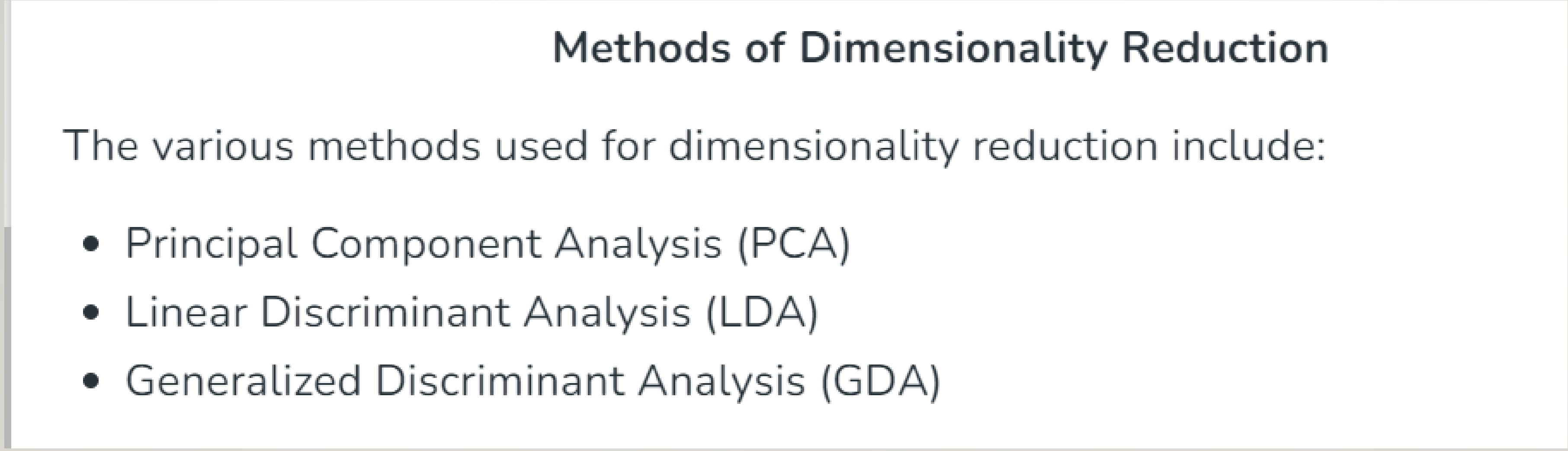

x_2 (inches)
x_1 (cm)
z_1


x_3
x_1
x_2
z_2
z_1
z_2
z_1
15
10
5
0
-5
-10
-15
-20
-25
-80
-60
-40
-20
0
20
40
60
80



Advantages of Dimensionality Reduction

- It helps in data compression, and hence reduced storage space.
- It reduces computation time.
- It also helps remove redundant features, if any.
- Improved Visualization: High dimensional data is difficult to visualize, and dimensionality reduction techniques can help in visualizing the data in 2D or 3D, which can help in better understanding and analysis.
- Overfitting Prevention: High dimensional data may lead to overfitting in machine learning models, which can lead to poor generalization performance. Dimensionalit reduction can help in reducing the complexity of the data, and hence prevent overfitting.
- Feature Extraction: Dimensionality reduction can help in extracting important features from high dimensional data, which can be useful in feature selection for machine learning models.
- Data Preprocessing: Dimensionality reduction can be used as a preprocessing step before applying machine learning algorithms to reduce the dimensionality of the data and hence improve the performance of the model.
- Improved Performance: Dimensionality reduction can help in improving the performance of machine learning models by reducing the complexity of the data, and hence reducing the noise and irrelevant information in the data.



Methods of Dimensionality Reduction

The various methods used for dimensionality reduction include:

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Generalized Discriminant Analysis (GDA)

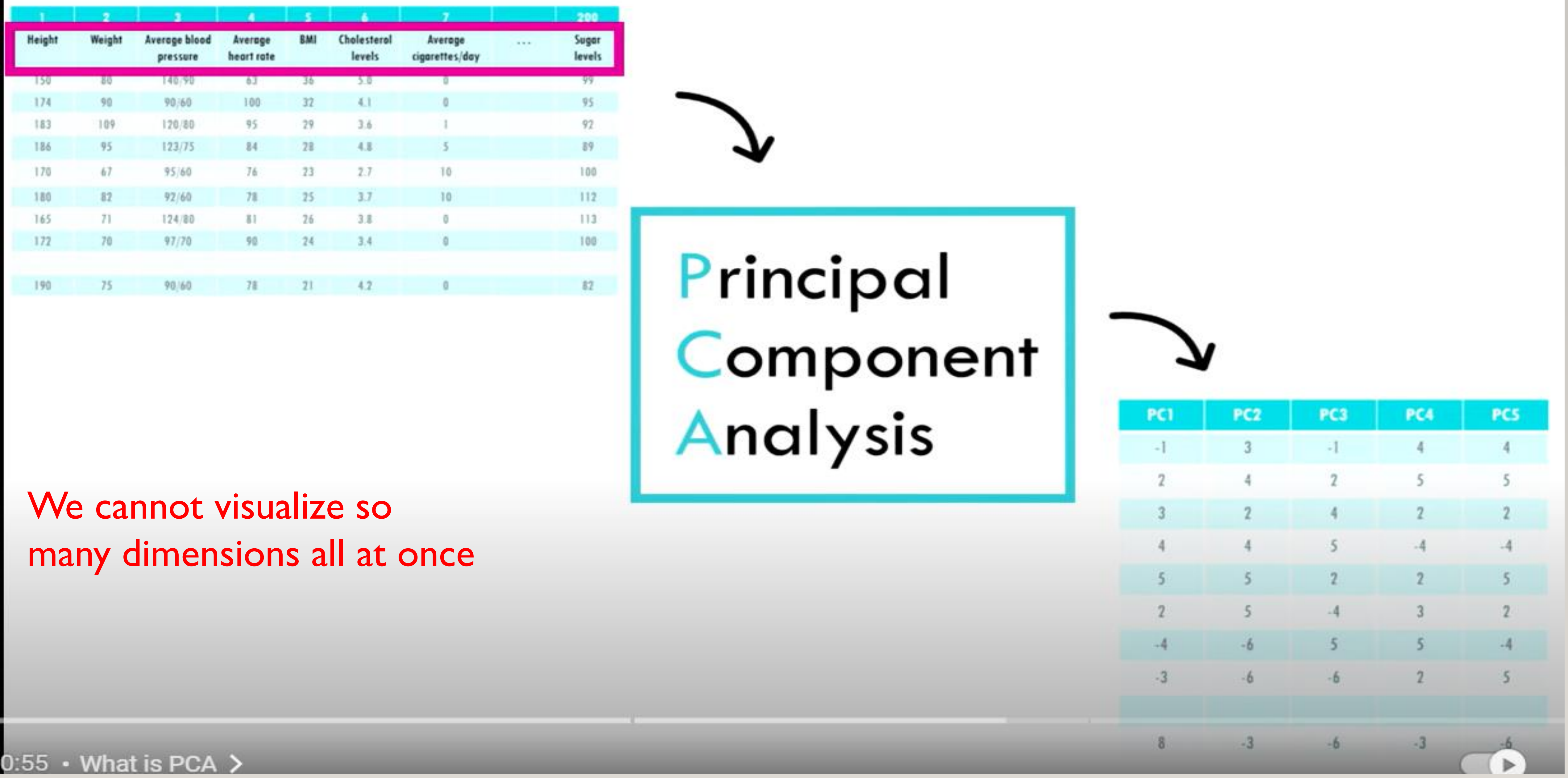
PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA

- Principal Component Analysis, or PCA, is a **dimensionality-reduction method**

- Unsupervised method.

- used to **reduce the dimensio**nality of large data sets, by transforming **a large set of variables into a smaller one** that still contains most of the information in the large set.
- Distribution of data set is linear (PCA assumption)

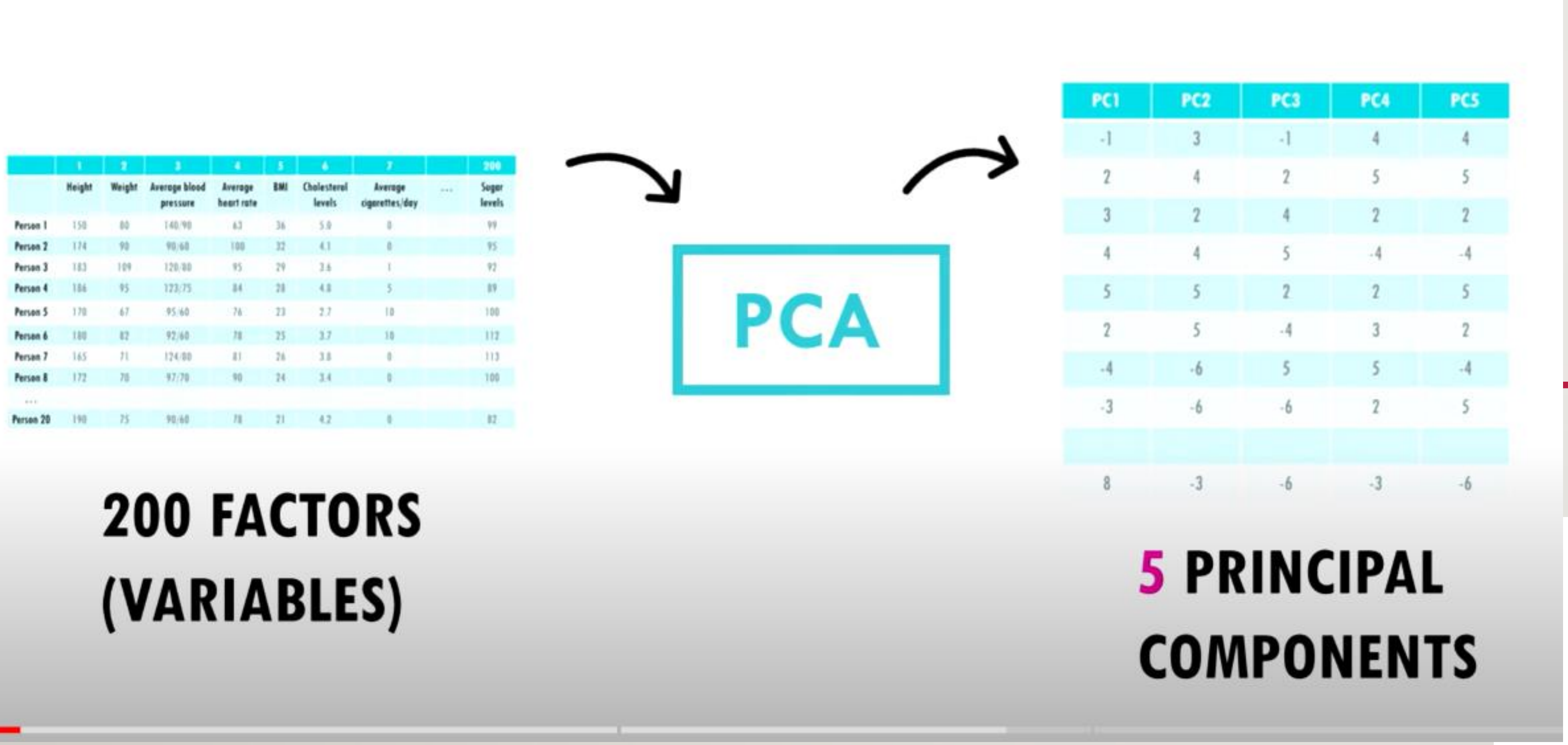


1	2	3	4	5	6	7		200
Height	Weight	Average blood pressure	Average heart rate	BMI	Cholesterol levels	Average cigarettes/day	...	Sugar levels
150	80	140/90	63	36	5.0	0		99
174	90	90/60	100	32	4.1	0		95
183	109	120/80	95	29	3.6	1		92
186	95	123/75	84	28	4.8	5		89
170	67	95/60	76	23	2.7	10		100
180	82	92/60	78	25	3.7	10		112
165	71	124/80	81	26	3.8	0		113
172	70	97/70	90	24	3.4	0		100
190	75	90/60	78	21	4.2	0		82

Principal
Component
Analysis

PC1	PC2	PC3	PC4	PC5
-1	3	-1	4	4
2	4	2	5	5
3	2	4	2	2
4	4	5	-4	-4
5	5	2	2	5
2	5	-4	3	2
-4	-6	5	5	-4
-3	-6	-6	2	5
8	-3	-6	-3	-6

We cannot visualize so many dimensions all at once



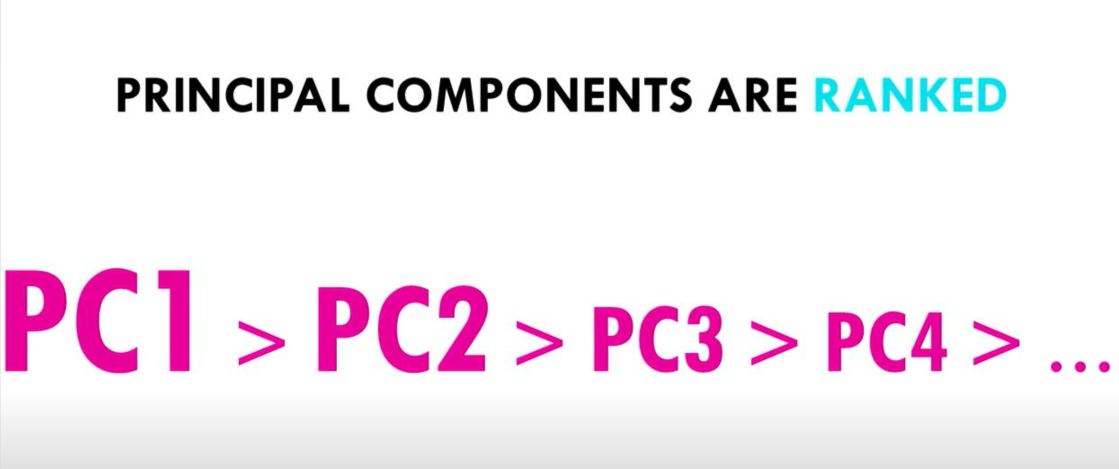
	1	2	3	4	5	6	7		200
	Height	Weight	Average blood pressure	Average heart rate	BMI	Cholesterol levels	Average cigarettes/day	...	Sugar levels
Person 1	150	80	140/90	63	36	5.0	0		99
Person 2	174	90	90/60	100	32	4.1	0		95
Person 3	183	109	120/80	95	29	3.6	1		92
Person 4	186	95	123/75	84	28	4.8	5		89
Person 5	170	67	95/60	76	23	2.7	10		100
Person 6	180	82	92/60	78	25	3.7	10		112
Person 7	165	71	124/80	81	26	3.8	0		113
Person 8	172	70	97/70	90	24	3.4	0		100
...									
Person 20	190	75	90/60	78	21	4.2	0		82

200 FACTORS (VARIABLES)

PC1	PC2	PC3	PC4	PC5
-1	3	-1	4	4
2	4	2	5	5
3	2	4	2	2
4	4	5	-4	-4
5	5	2	2	5
2	5	-4	3	2
-4	-6	5	5	-4
-3	-6	-6	2	5
8	-3	-6	-3	-6

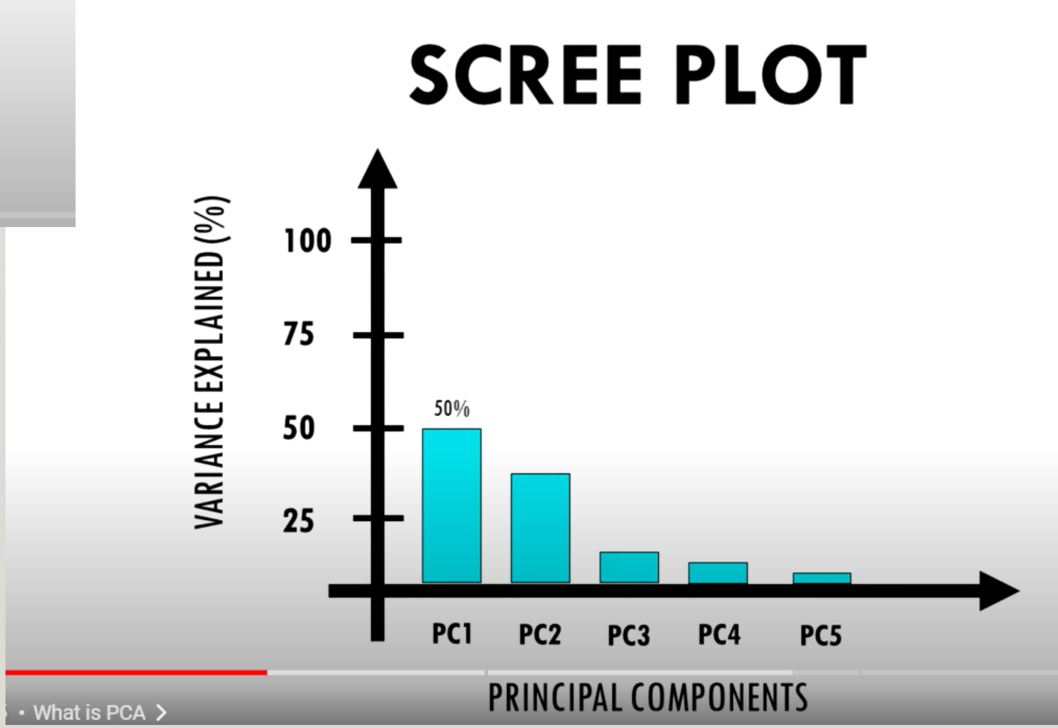
5 PRINCIPAL COMPONENTS

Scree plot means how much variance (information) each PCA plots



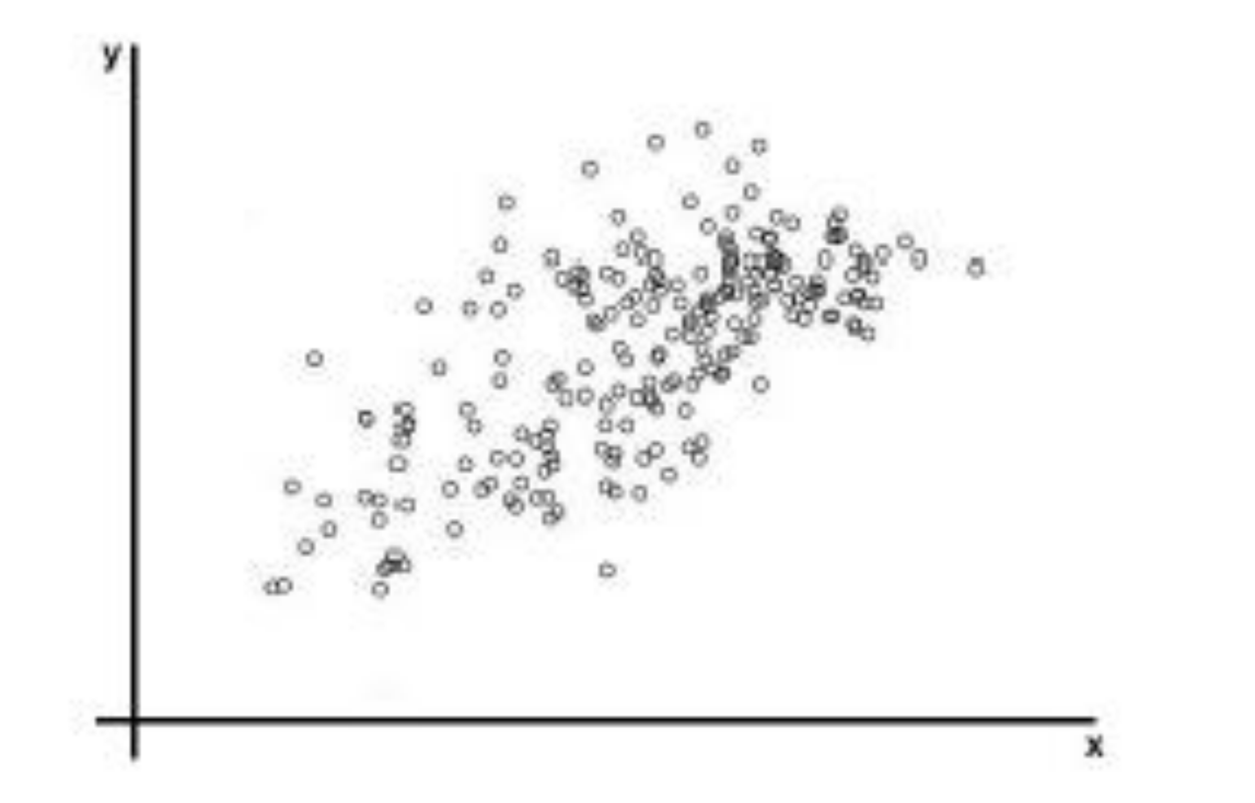
PRINCIPAL COMPONENTS ARE RANKED

PC1 > PC2 > PC3 > PC4 > ...

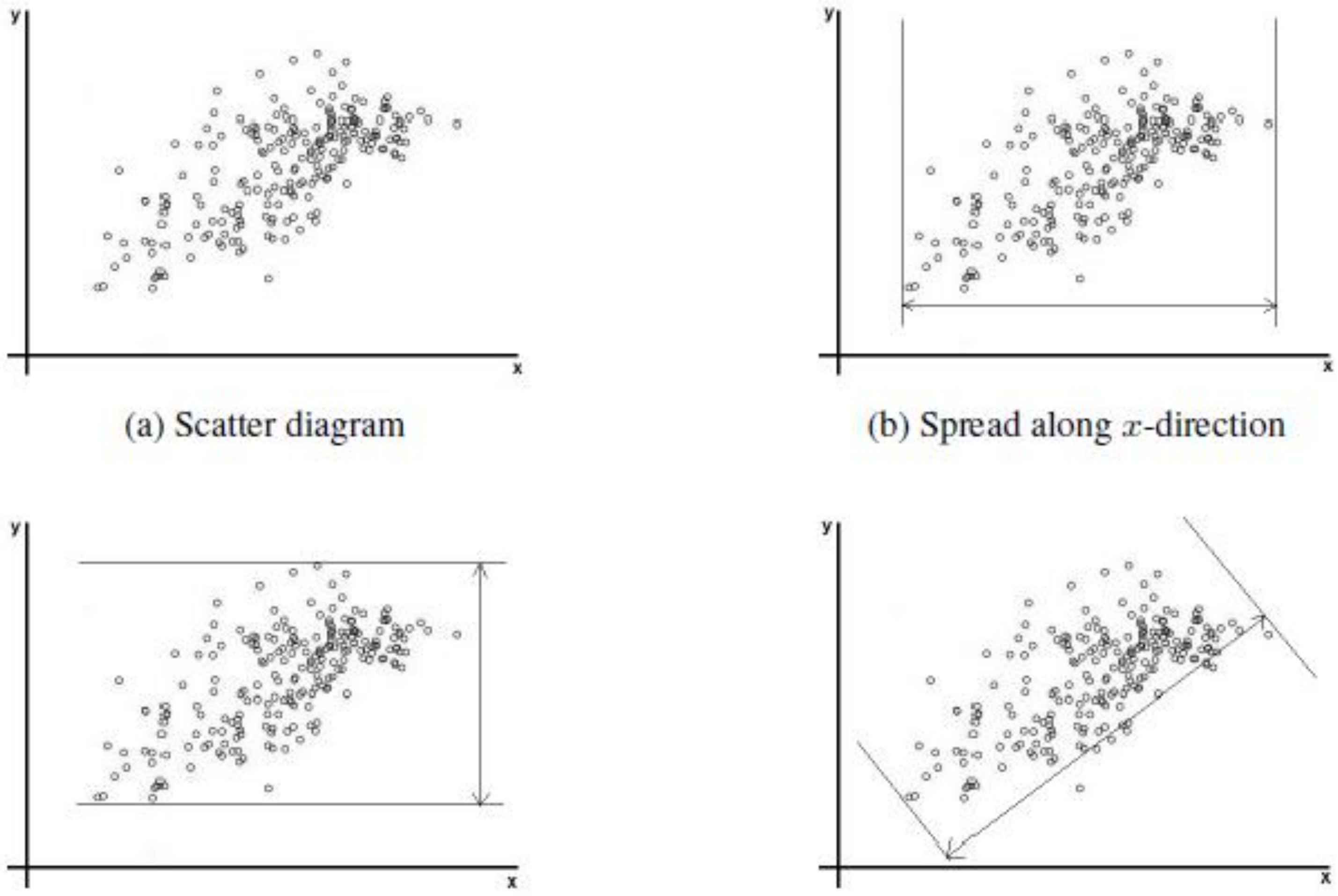


SCREE PLOT

PCA



(a) Scatter diagram

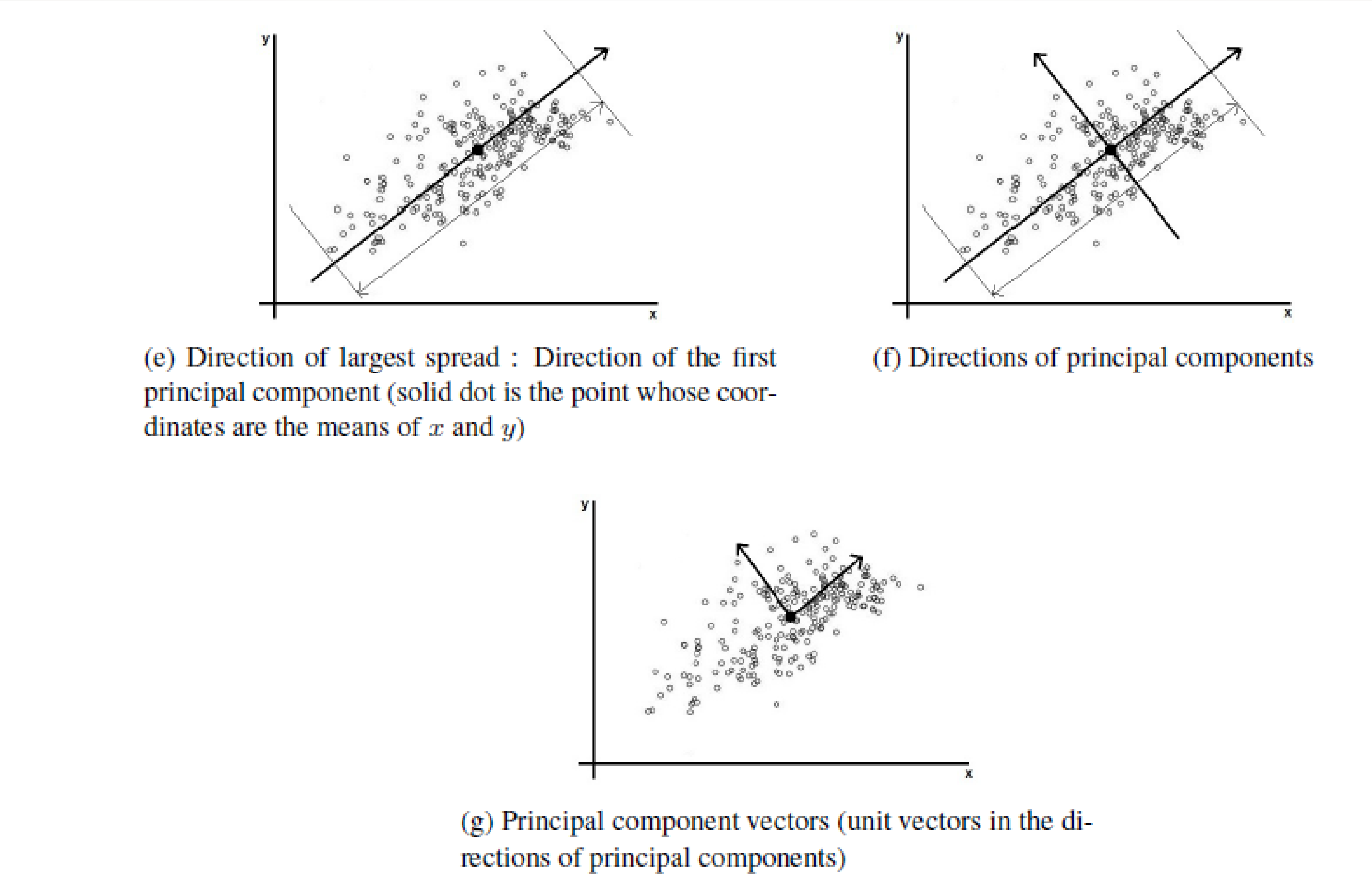


(a) Scatter diagram

(b) Spread along x-direction

(c) Spread along y-direction

(d) Largest spread

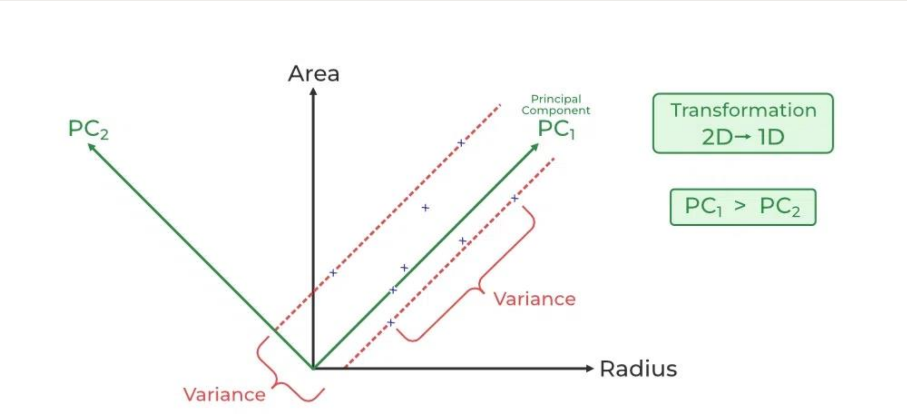


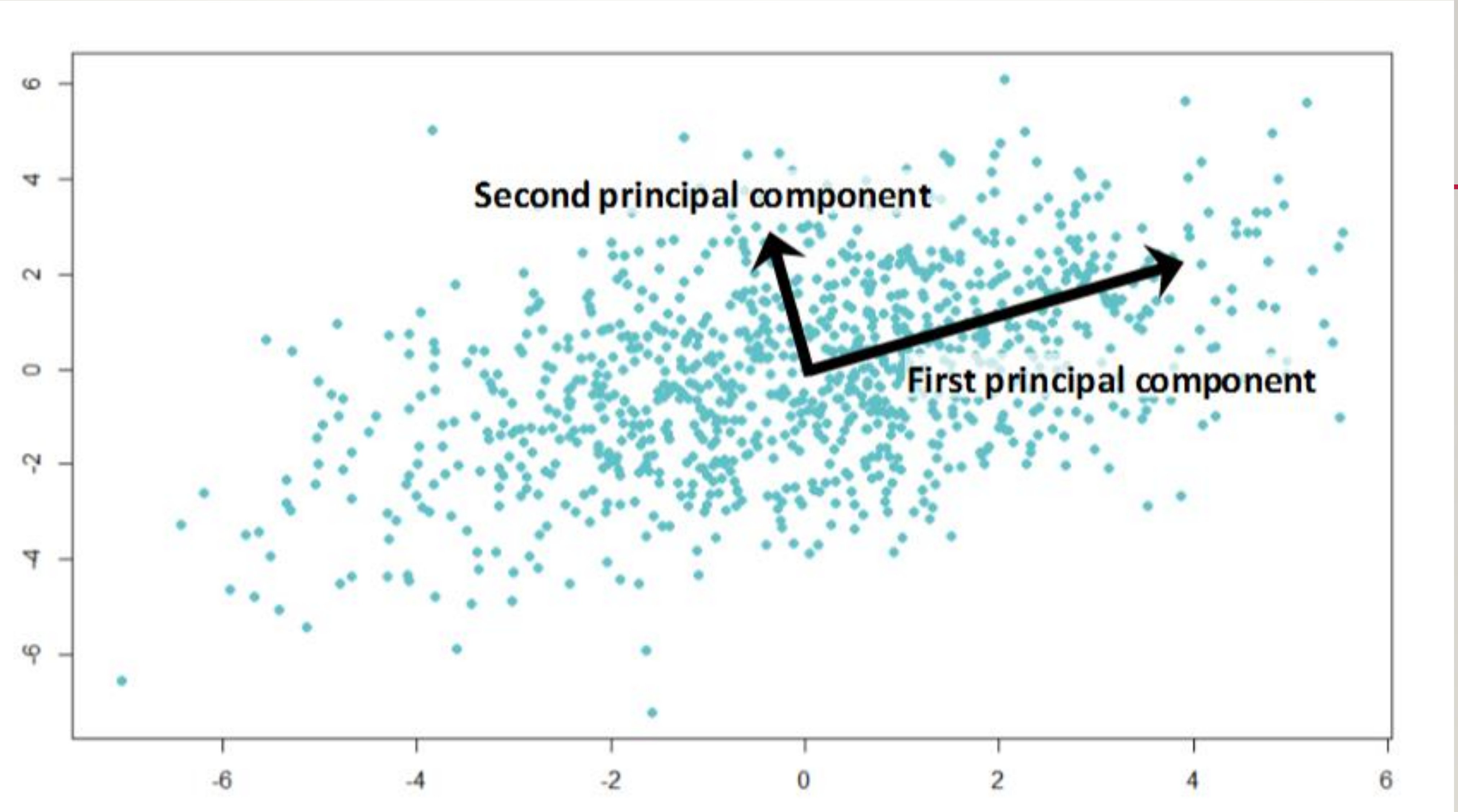
(e) Direction of largest spread : Direction of the first principal component (solid dot is the point whose coordinates are the means of x and y)

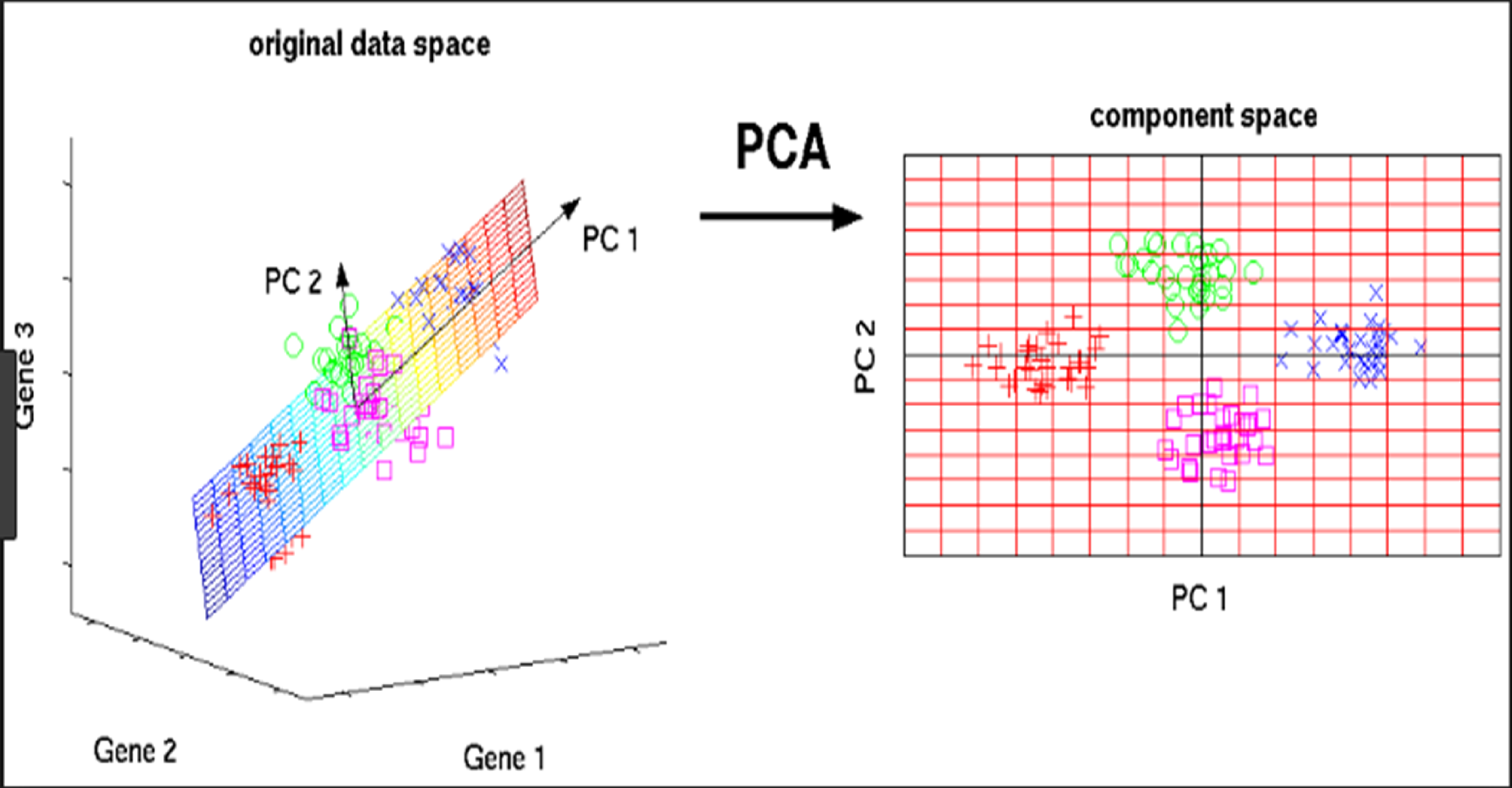
(f) Directions of principal components

(g) Principal component vectors (unit vectors in the directions of principal components)

- **principal components analysis (PCA)** is a technique that can be used to simplify a dataset
- It is a linear transformation that **CHOOSES A NEW COORDINATE SYSTEM** for the data set such that
  - **greatest variance** by any projection of the data set comes to lie on the first axis (then called the first principal component),
  - the **second greatest variance** on the second axis (second principal component), and so on.




Second principal component
First principal component
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6



original data space

component space

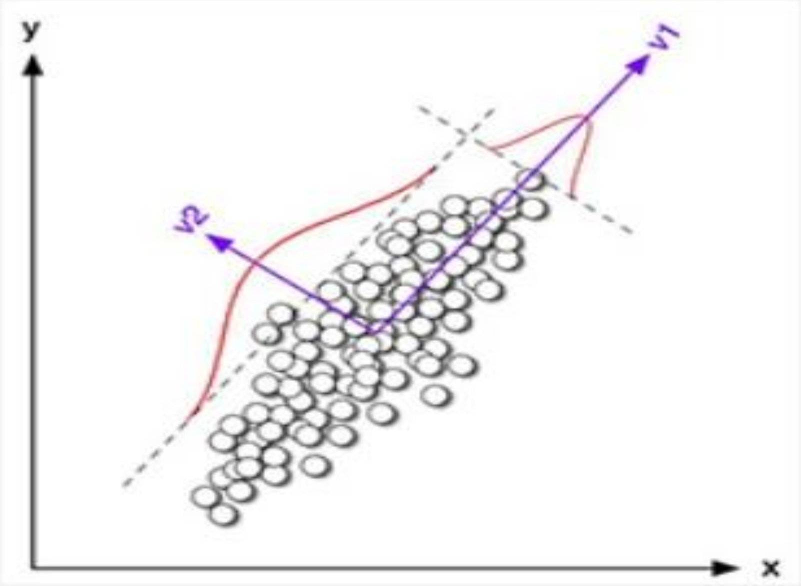
Steps for PCA

1. Standardize the data

2. Calculate the covariance matrix

3. Find the eigenvalues and eingenvectors of the covariance matrix

4. Plot the eigenvectors / principal components over the scaled data



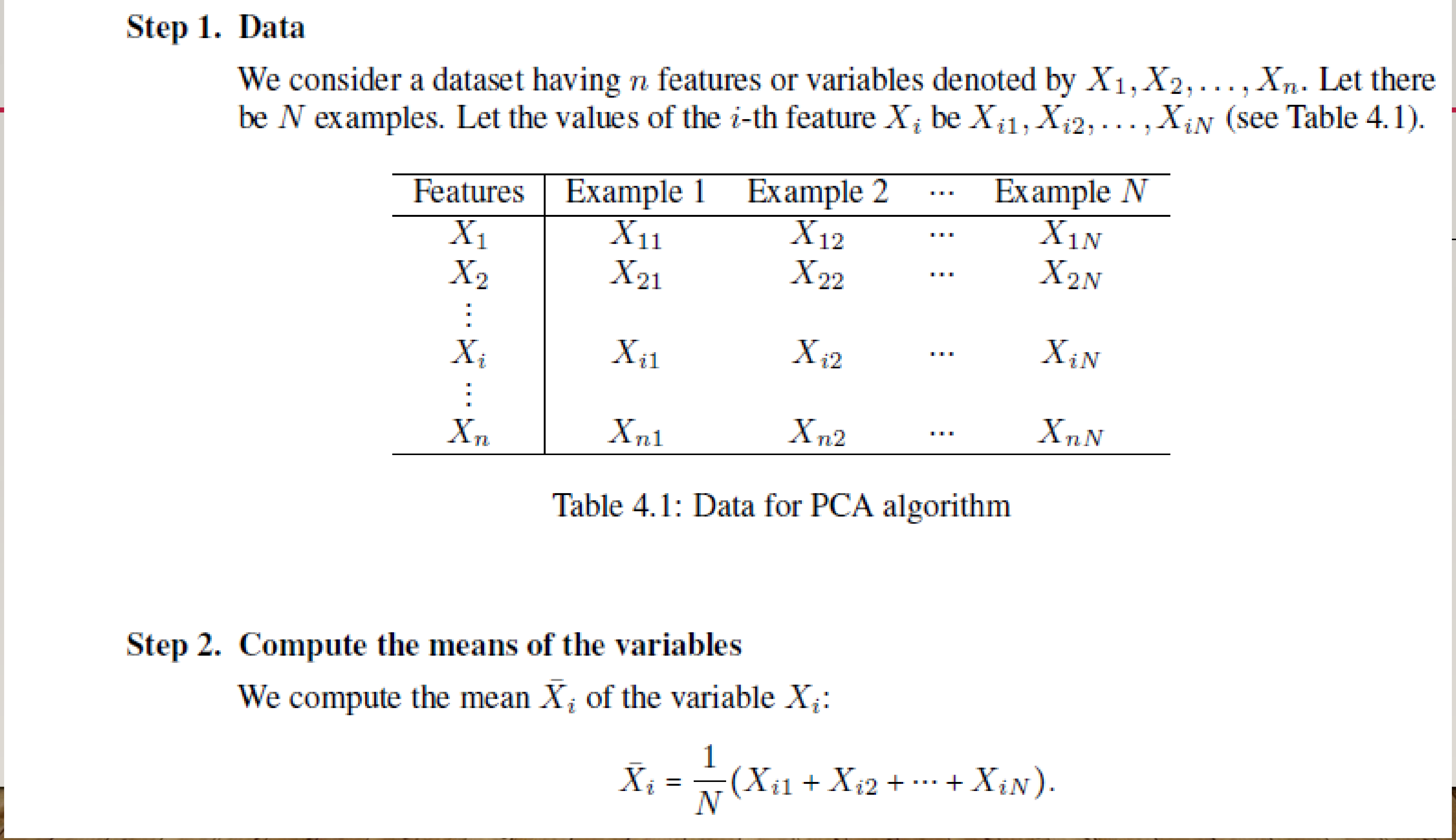
PCA

- By finding the **eigenvalues and eigenvectors of the covariance matrix**, we find that the eigenvectors with **the largest eigenvalues** correspond to the dimensions that have the strongest correlation in the dataset.
- ***This is the principal component.***

EXAMPLE APPLICATIONS

- Face Recognition
- Image Compression
- Pattern finding
- Gene Expression Analysis
- Data Reduction
- Data Classification
- Trend Analysis
- Factor Analysis
- Noise Reduction

PCA PROCESS –STEP 1



Step 1. Data

We consider a dataset having n features or variables denoted by $X_1, X_2, \dots, X_n$. Let there be N examples. Let the values of the i-th feature X_i be $X_{i1}, X_{i2}, \dots, X_{iN}$ (see Table 4.1).

Features	Example 1	Example 2	$\cdots$	Example N
X_1	X_{11}	X_{12}	$\cdots$	X_{1N}
X_2	X_{21}	X_{22}	$\cdots$	X_{2N}
$\vdots$				
X_i	X_{i1}	X_{i2}	$\cdots$	X_{iN}
$\vdots$				
X_n	X_{n1}	X_{n2}	$\cdots$	X_{nN}

Table 4.1: Data for PCA algorithm

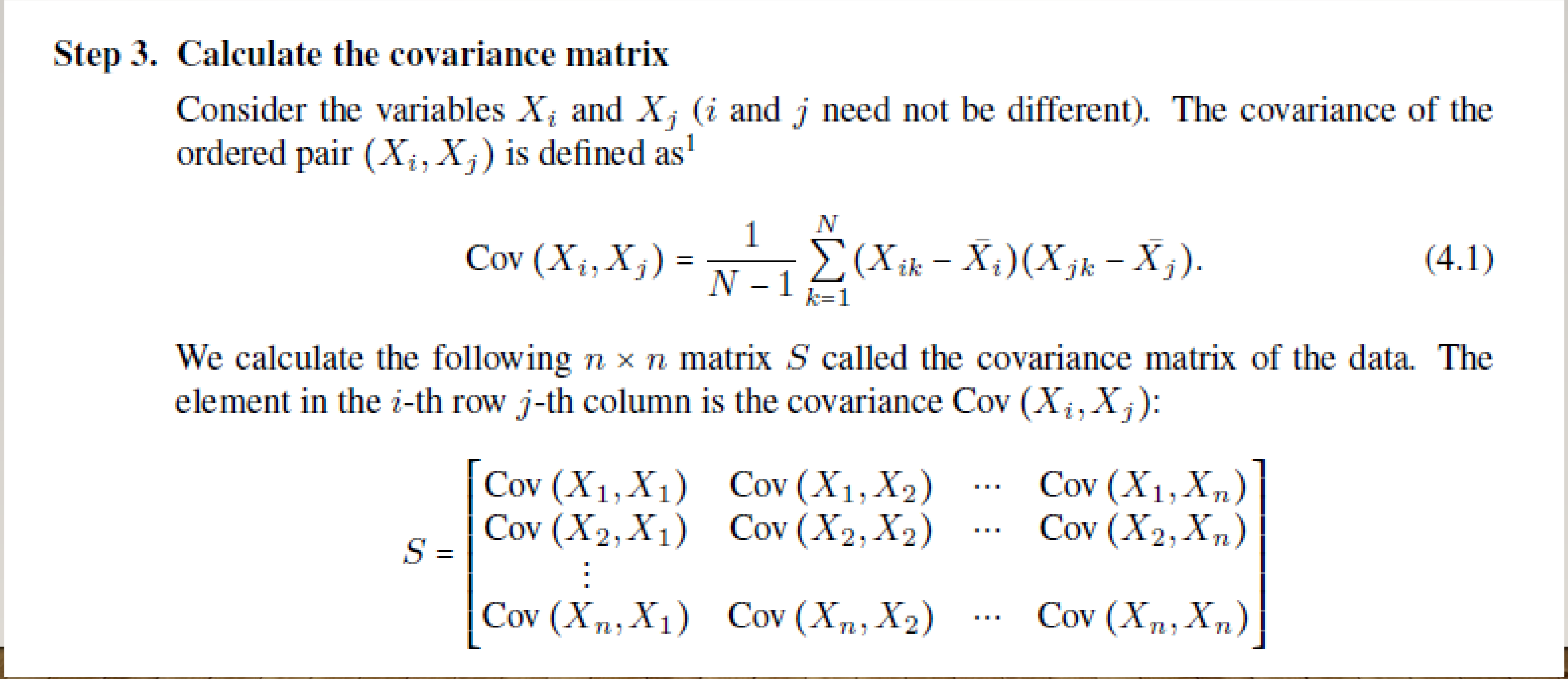
Step 2. Compute the means of the variables

We compute the mean $\bar{X}_i$ of the variable X_i:

$$\bar{X}_i = \frac{1}{N}(X_{i1} + X_{i2} + \cdots + X_{iN}).$$

PCA PROCESS –STEP 2

- Calculate the covariance matrix



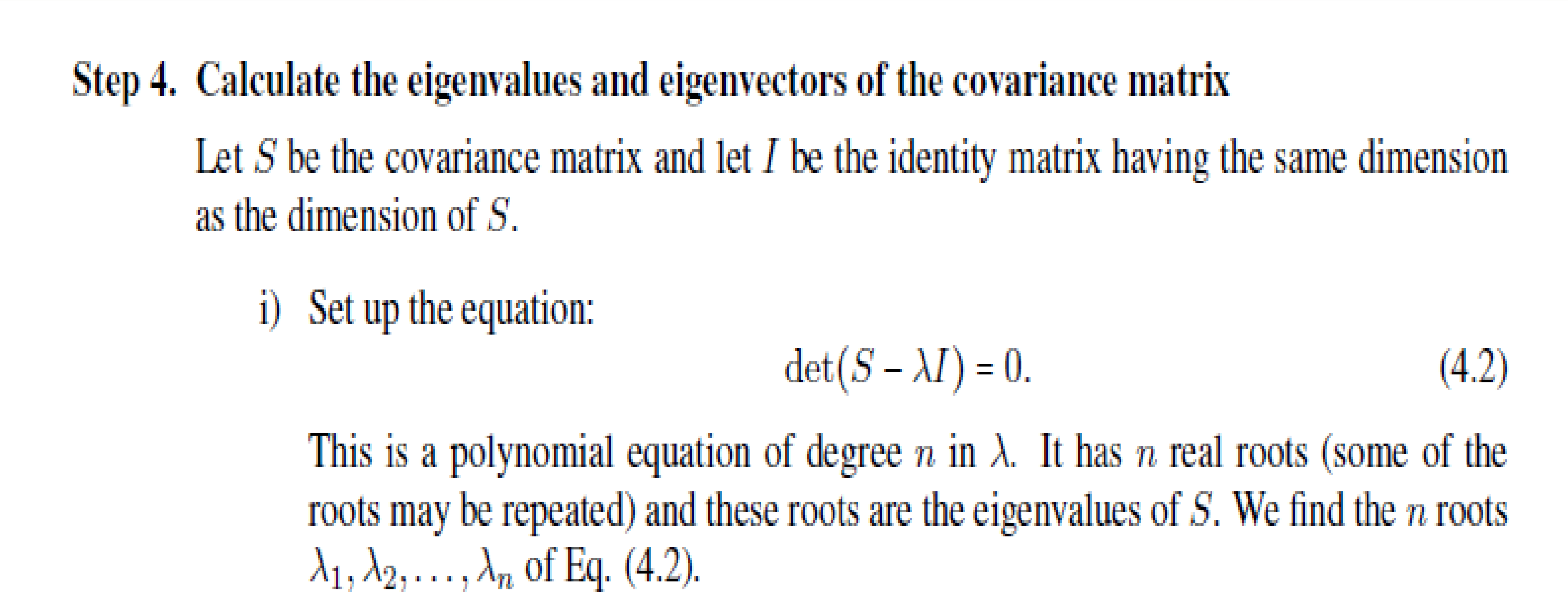
Step 3. Calculate the covariance matrix

Consider the variables X_i and X_j (i and j need not be different). The covariance of the ordered pair (X_i, X_j) is defined as[1]

$$\operatorname{Cov}(X_i, X_j) = \frac{1}{N-1}\sum_{k=1}^{N}(X_{ik} - \bar{X}_i)(X_{jk} - \bar{X}_j). \tag{4.1}$$

We calculate the following $n \times n$ matrix S called the covariance matrix of the data. The element in the i-th row j-th column is the covariance $\operatorname{Cov}(X_i, X_j)$:

$$S = \begin{bmatrix} \operatorname{Cov}(X_1, X_1) & \operatorname{Cov}(X_1, X_2) & \cdots & \operatorname{Cov}(X_1, X_n) \\ \operatorname{Cov}(X_2, X_1) & \operatorname{Cov}(X_2, X_2) & \cdots & \operatorname{Cov}(X_2, X_n) \\ \vdots & & & \\ \operatorname{Cov}(X_n, X_1) & \operatorname{Cov}(X_n, X_2) & \cdots & \operatorname{Cov}(X_n, X_n) \end{bmatrix}$$



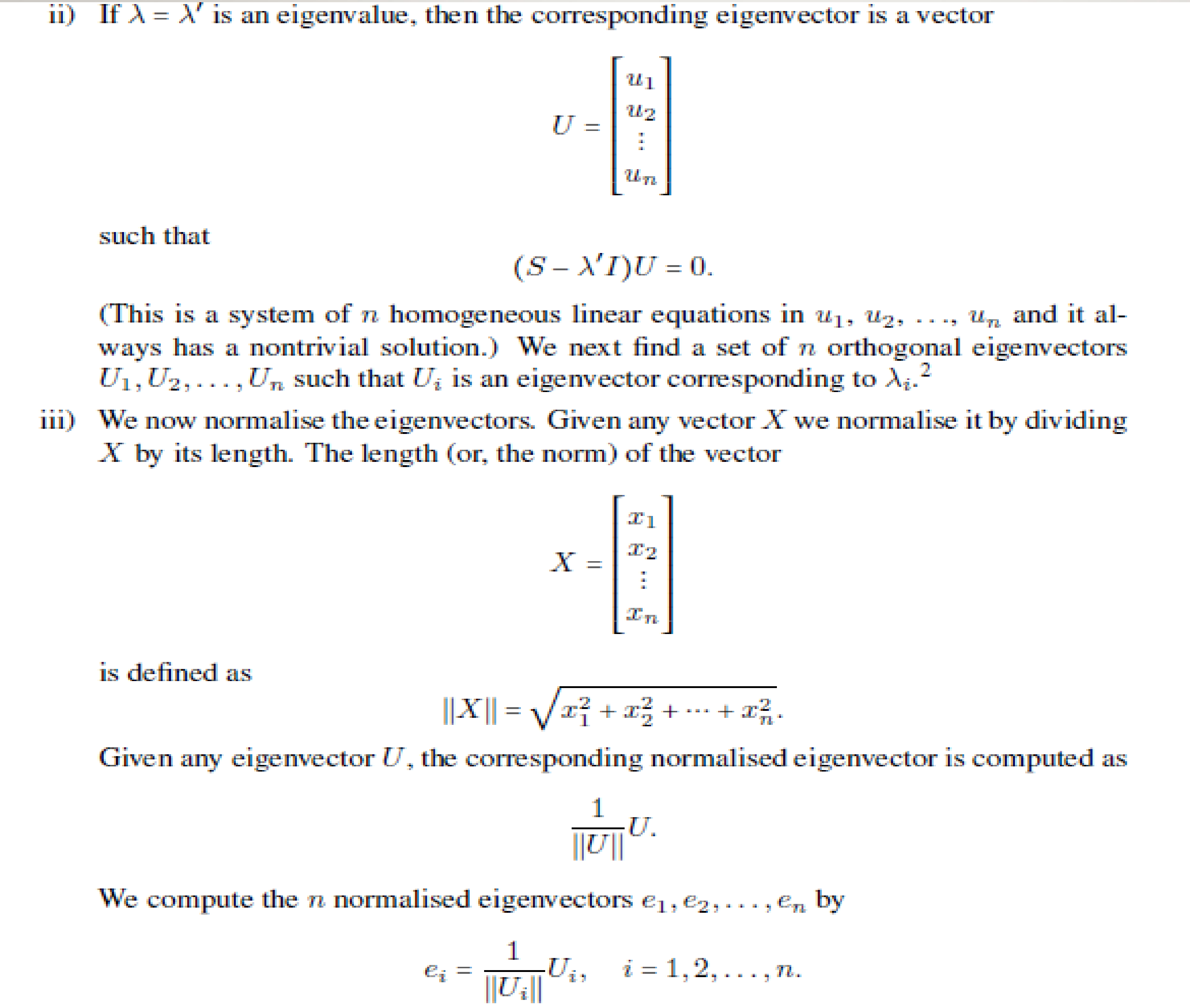
Step 4. Calculate the eigenvalues and eigenvectors of the covariance matrix

Let S be the covariance matrix and let I be the identity matrix having the same dimension as the dimension of S.

i) Set up the equation:

$$\det(S - \lambda I) = 0. \tag{4.2}$$

This is a polynomial equation of degree n in λ. It has n real roots (some of the roots may be repeated) and these roots are the eigenvalues of S. We find the n roots $\lambda_1, \lambda_2, \ldots, \lambda_n$ of Eq. (4.2).



ii) If $\lambda = \lambda'$ is an eigenvalue, then the corresponding eigenvector is a vector

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

such that

$$(S - \lambda' I)U = 0.$$

(This is a system of n homogeneous linear equations in u_1, u_2, ..., u_n and it always has a nontrivial solution.) We next find a set of n orthogonal eigenvectors $U_1, U_2, \ldots, U_n$ such that U_i is an eigenvector corresponding to λ_i.[2]

iii) We now normalise the eigenvectors. Given any vector X we normalise it by dividing X by its length. The length (or, the norm) of the vector

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

is defined as

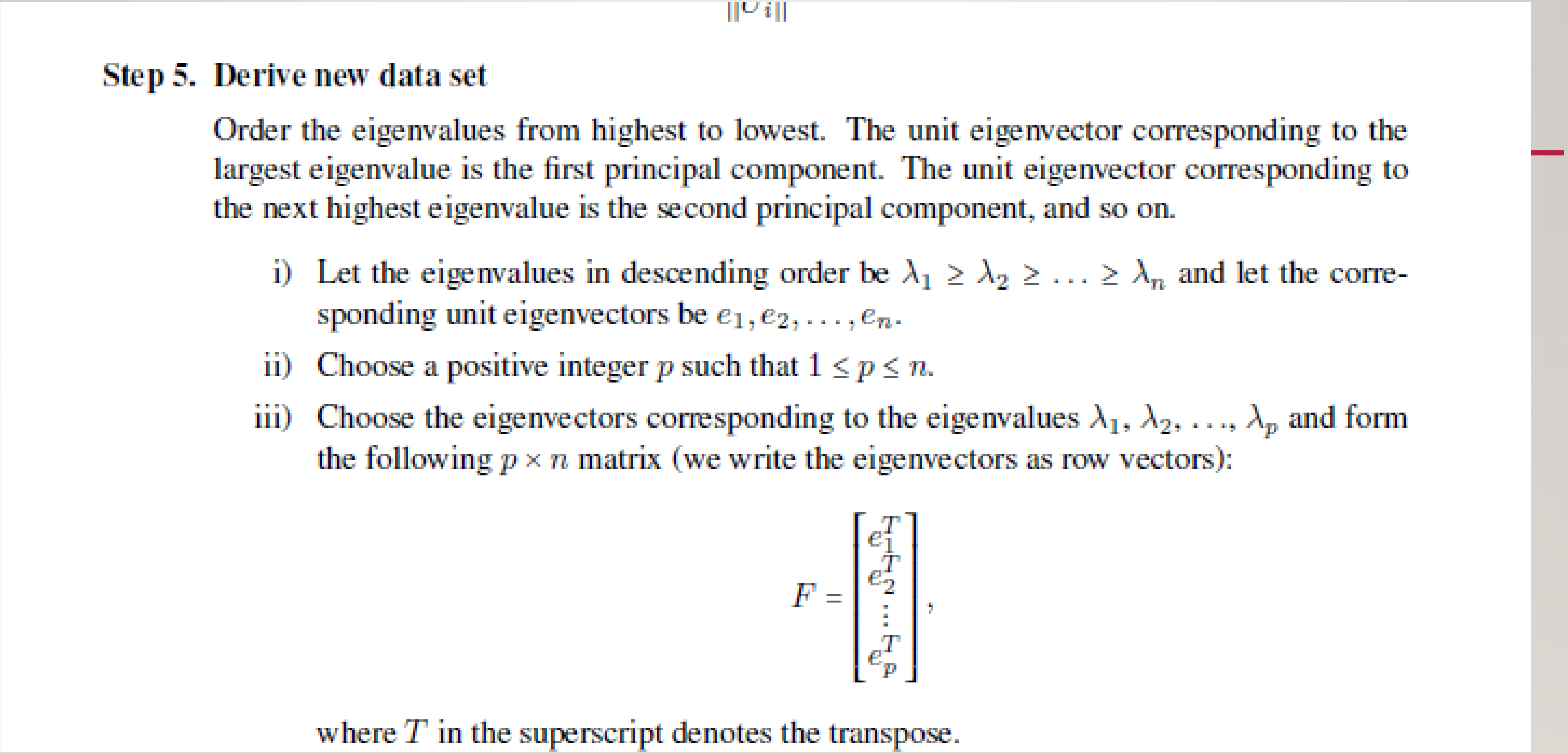
$$||X|| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

Given any eigenvector U, the corresponding normalised eigenvector is computed as

$$\frac{1}{||U||} U.$$

We compute the n normalised eigenvectors $e_1, e_2, \ldots, e_n$ by

$$e_i = \frac{1}{||U_i||} U_i, \quad i = 1, 2, \ldots, n.$$



$||U_i||$

Step 5. Derive new data set

Order the eigenvalues from highest to lowest. The unit eigenvector corresponding to the largest eigenvalue is the first principal component. The unit eigenvector corresponding to the next highest eigenvalue is the second principal component, and so on.

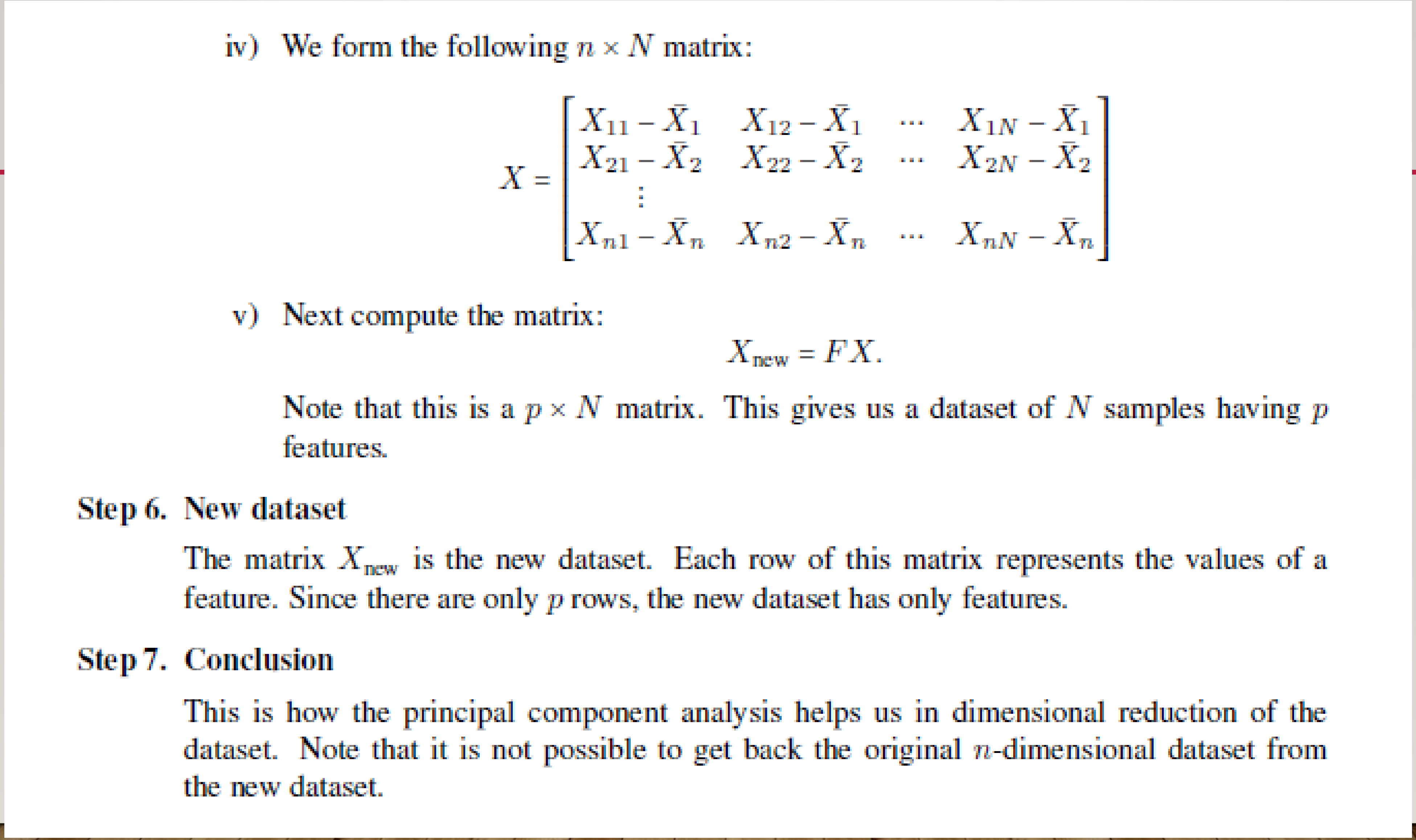
i) Let the eigenvalues in descending order be $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ and let the corresponding unit eigenvectors be $e_1, e_2, \ldots, e_n$.

ii) Choose a positive integer p such that $1 \leq p \leq n$.

iii) Choose the eigenvectors corresponding to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ and form the following $p \times n$ matrix (we write the eigenvectors as row vectors):

$$F = \begin{bmatrix} e_1^T \\ e_2^T \\ \vdots \\ e_p^T \end{bmatrix},$$

where T in the superscript denotes the transpose.



iv) We form the following $n \times N$ matrix:

$$X = \begin{bmatrix} X_{11} - \bar{X}_1 & X_{12} - \bar{X}_1 & \cdots & X_{1N} - \bar{X}_1 \\ X_{21} - \bar{X}_2 & X_{22} - \bar{X}_2 & \cdots & X_{2N} - \bar{X}_2 \\ \vdots & & & \\ X_{n1} - \bar{X}_n & X_{n2} - \bar{X}_n & \cdots & X_{nN} - \bar{X}_n \end{bmatrix}$$

v) Next compute the matrix:

$$X_{\text{new}} = FX.$$

Note that this is a $p \times N$ matrix. This gives us a dataset of N samples having p features.

Step 6. New dataset

The matrix X_{new} is the new dataset. Each row of this matrix represents the values of a feature. Since there are only p rows, the new dataset has only features.

Step 7. Conclusion

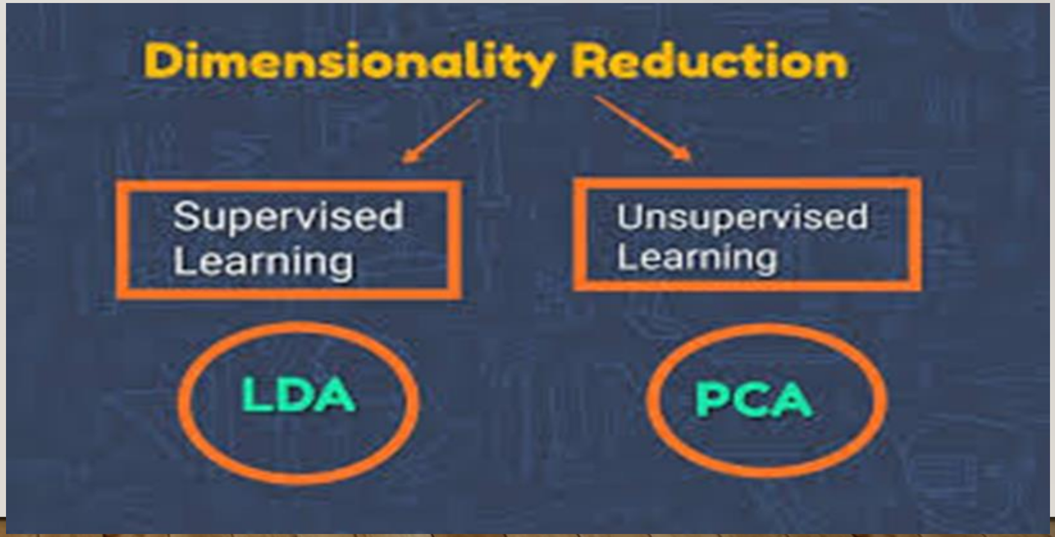
This is how the principal component analysis helps us in dimensional reduction of the dataset. Note that it is not possible to get back the original n-dimensional dataset from the new dataset.

Dimensionality Reduction Technique

2. Linear Discriminant analysis (LDA)

LDA is a technique that transforms a set of features or variables into a smaller set of new features, called **linear discriminants**, that are optimal for **separating different classes** or categories of the data.

LDA aims to find the directions or axes that maximize the between-class variance and **minimize the within-class variance**, and project the data onto those axes.



LINEAR DISCRIMINANT ANALYSIS (LDA)

- Linear Discriminant Analysis (LDA) is a powerful technique in machine learning and statistics.

- It is primarily used for **dimensionality reduction and classification tasks.**

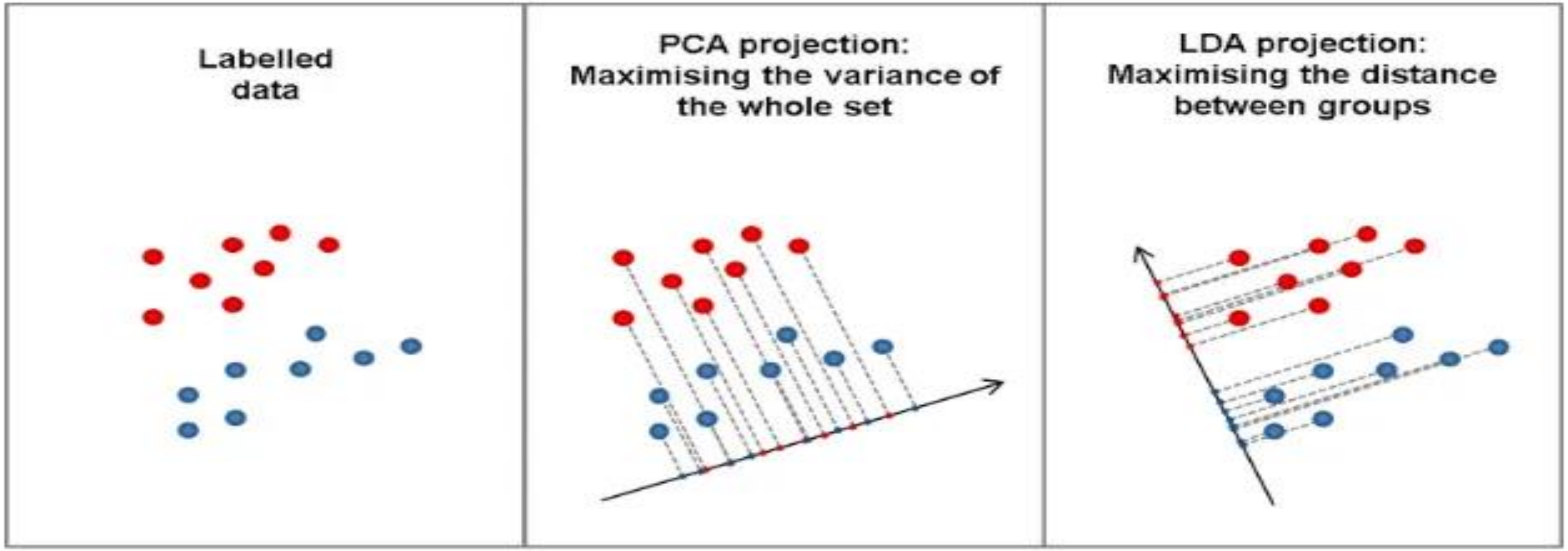
- LDA is a **supervised learning method** that ***finds a linear combination of features that best separates or discriminates between classes in a dataset***.

- **Used in data preprocessing**
- **LDA separates multiple classes with multiple features through data dimensionality reduction.**

Why Use Linear Discriminant Analysis?

1. Dimensionality Reduction
2. Feature Extraction: reduce noise or redundancy
3. Improving classification accuracy
4. Data Visualization
5. Handling multiple class problems
6. Reducing overfitting
7. Assumption of Normality: data in Gaussian distribution

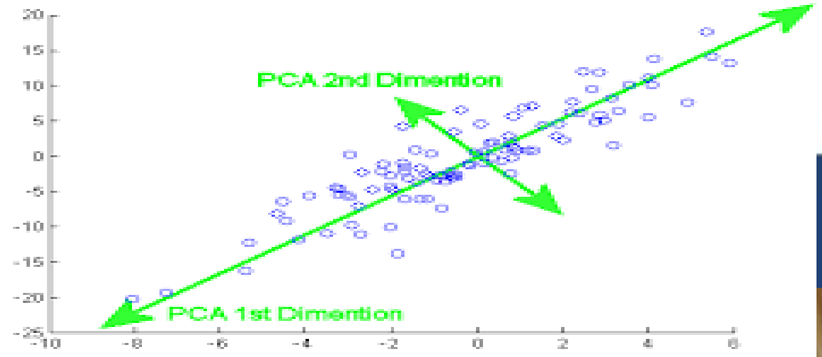
Linear Discriminant Analysis



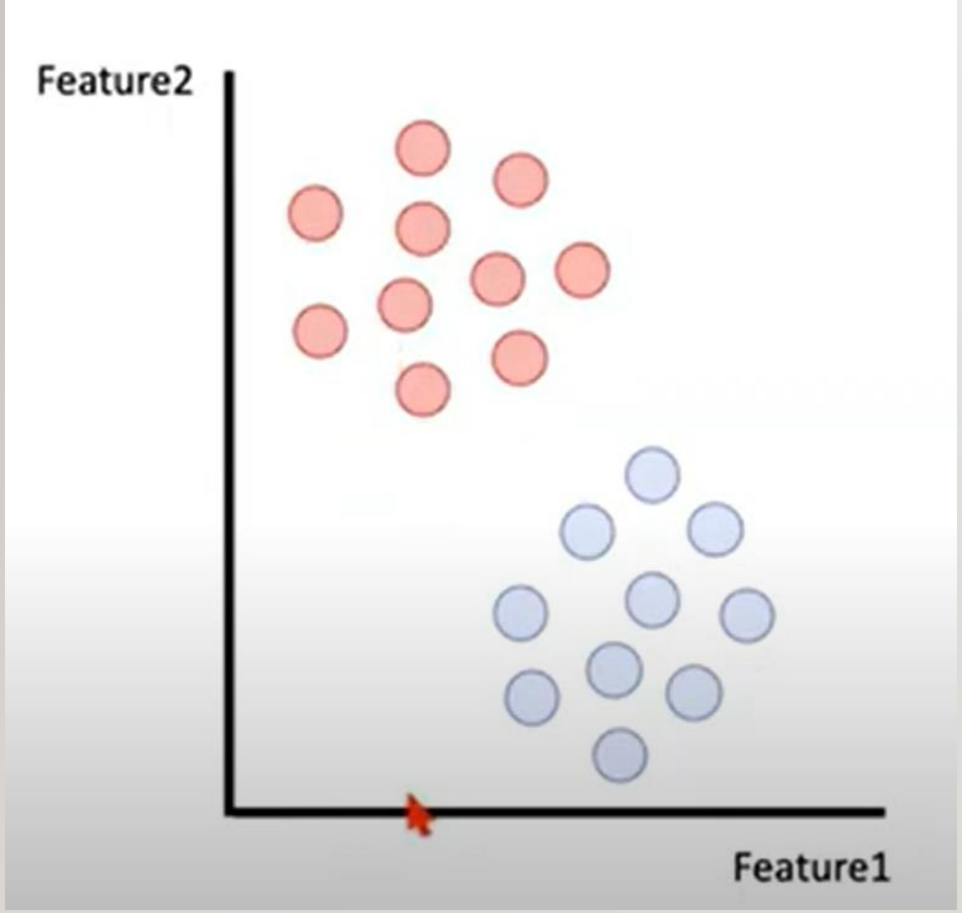
Labelled data

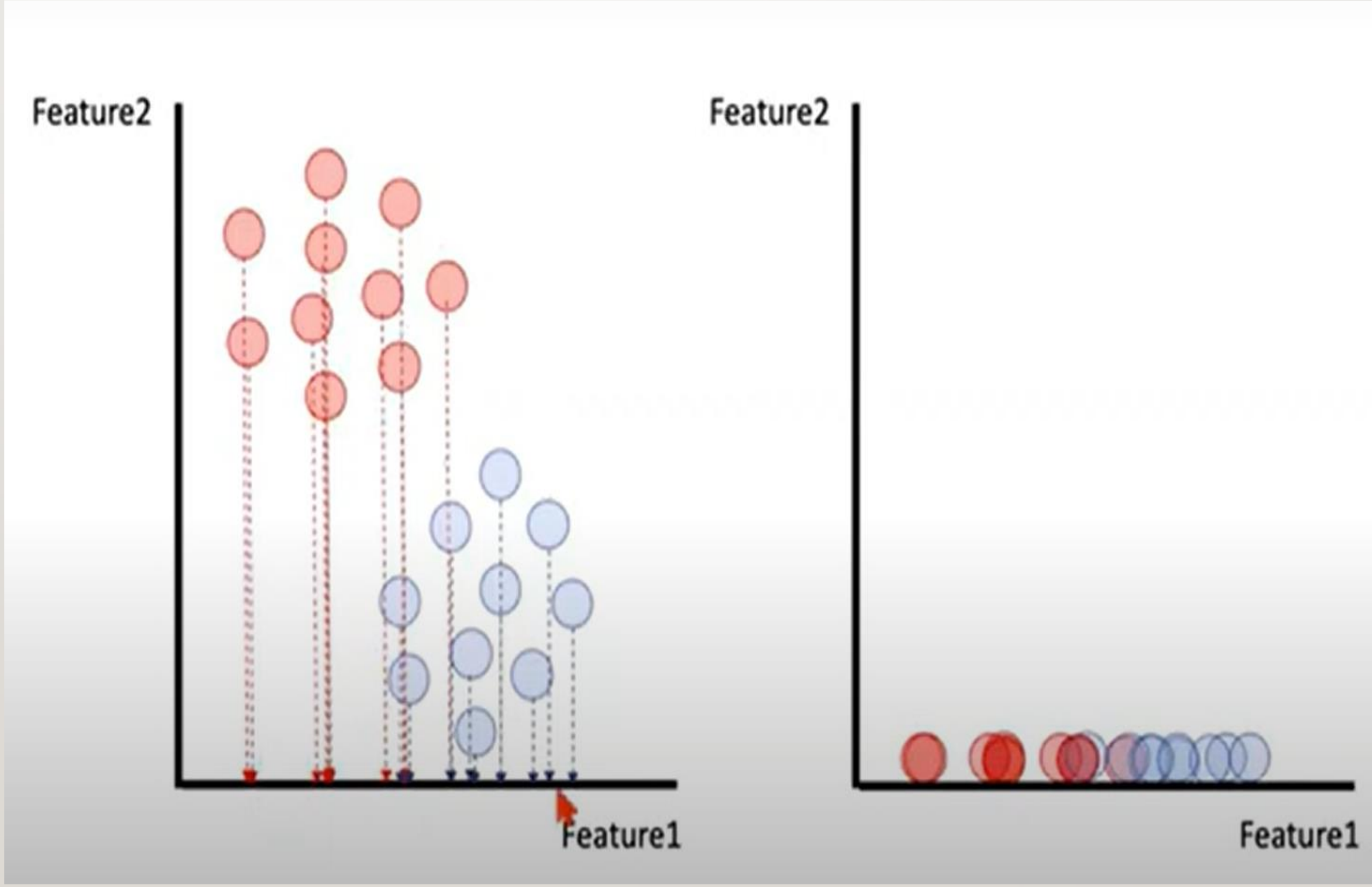
PCA projection: Maximising the variance of the whole set

LDA projection: Maximising the distance between groups

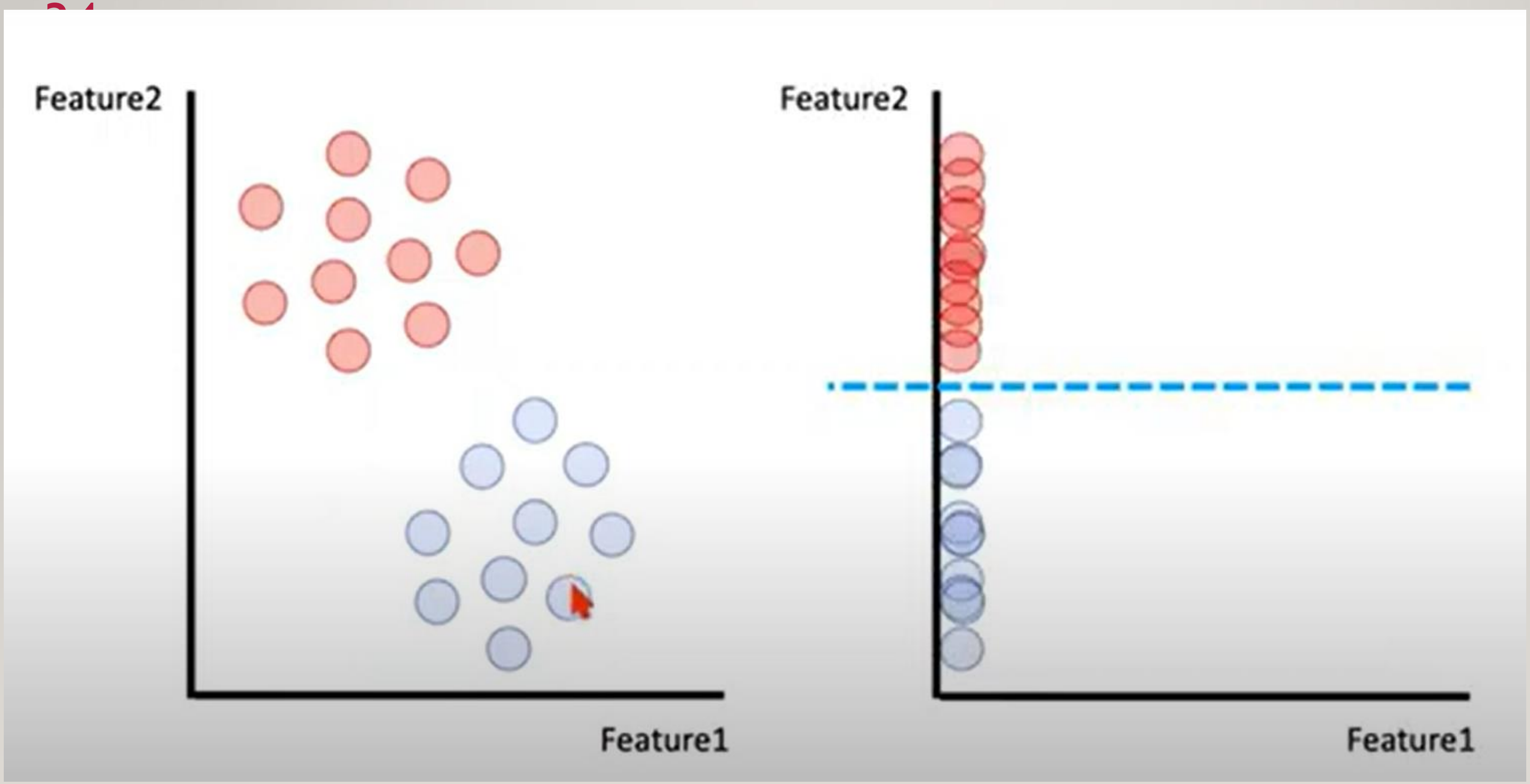


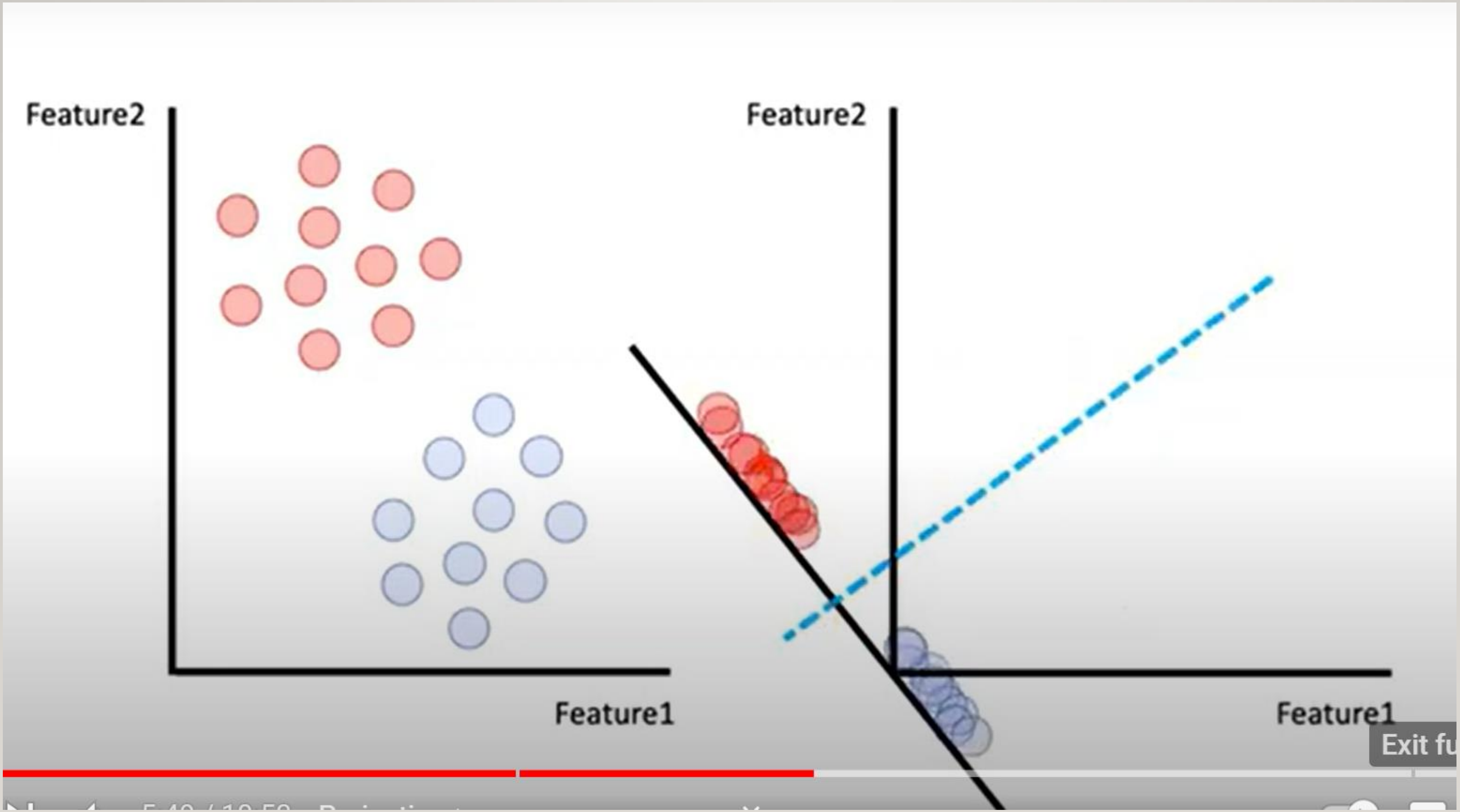
Mapping of data to Feature1 axis

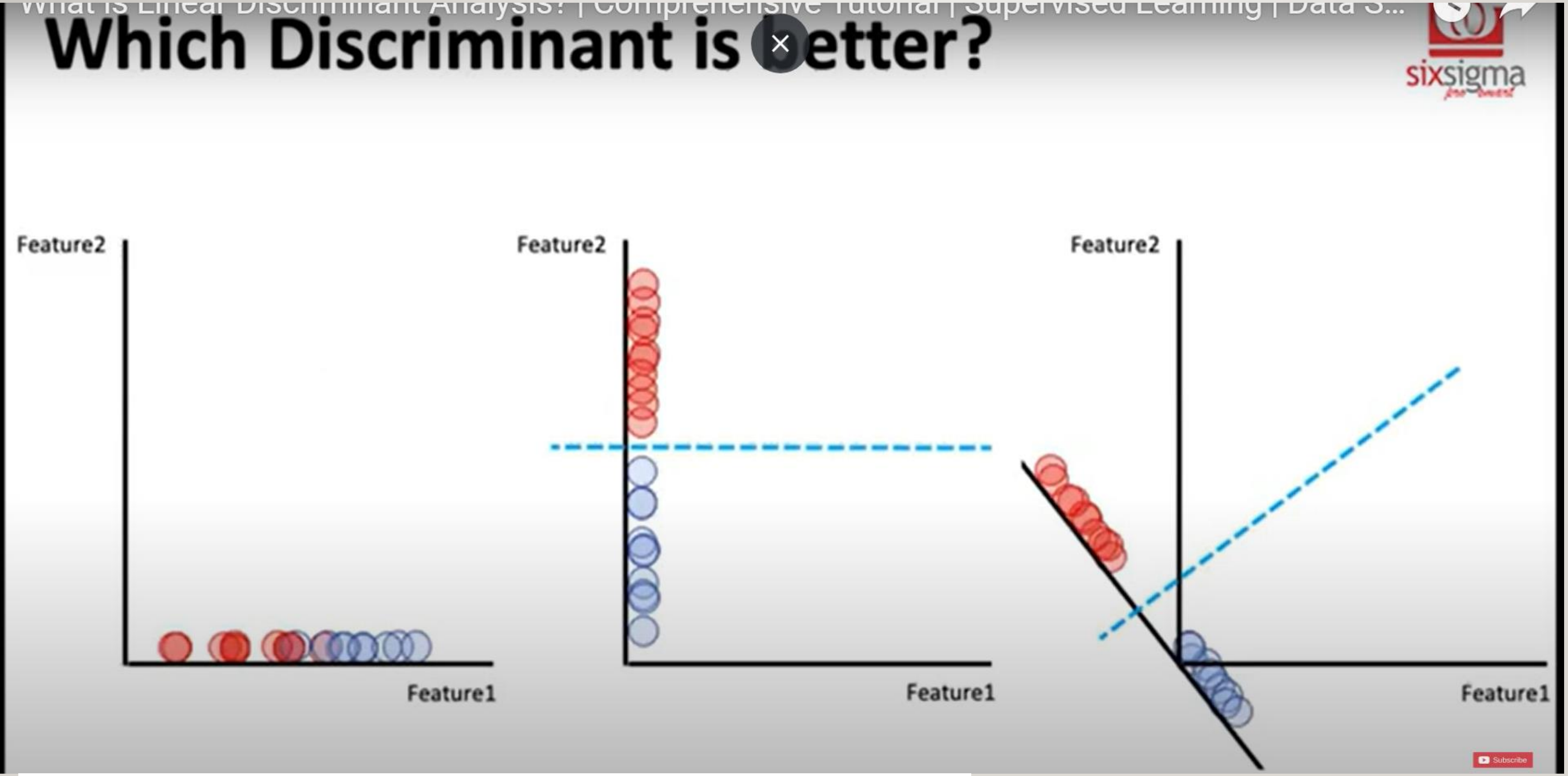




Mapping of data to Feature2 axis

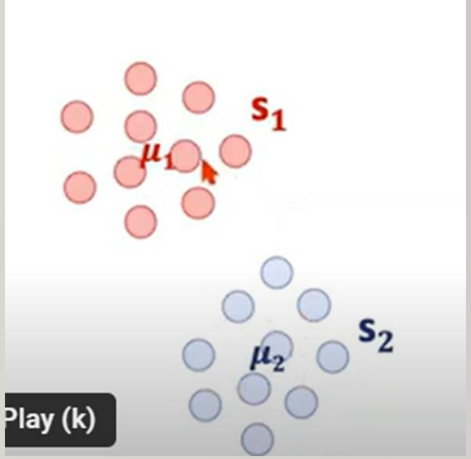



Feature2
Feature1
Feature2
Feature1
Exit fu



What is Linear Discriminant Analysis? | Comprehensive Tutorial | Supervised Learning | Data S...

Which Discriminant is better?

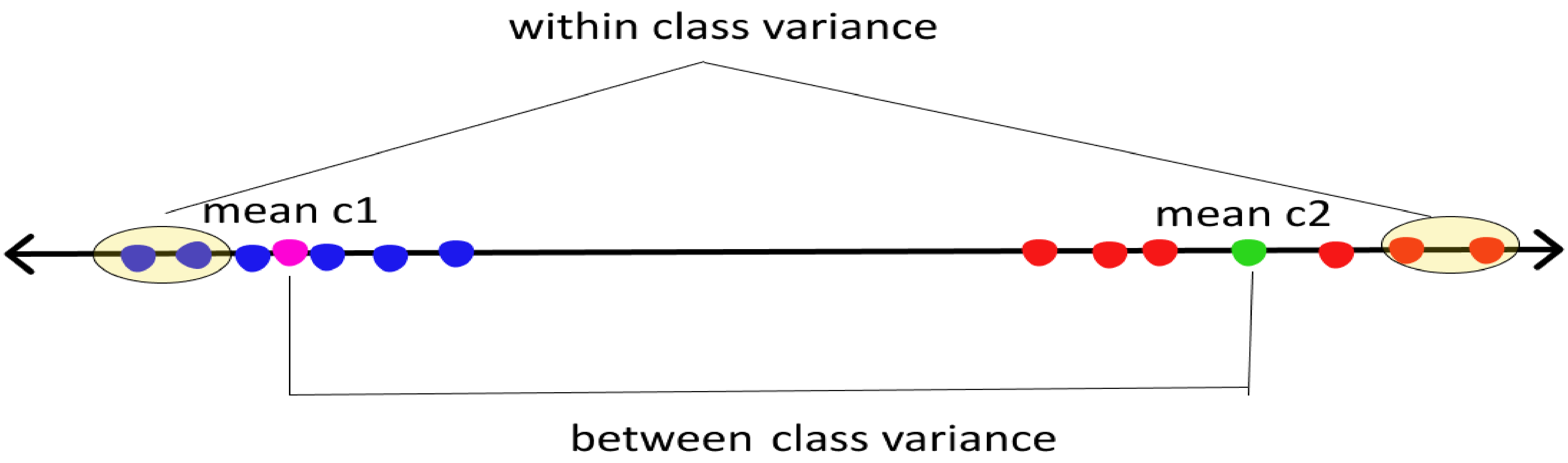


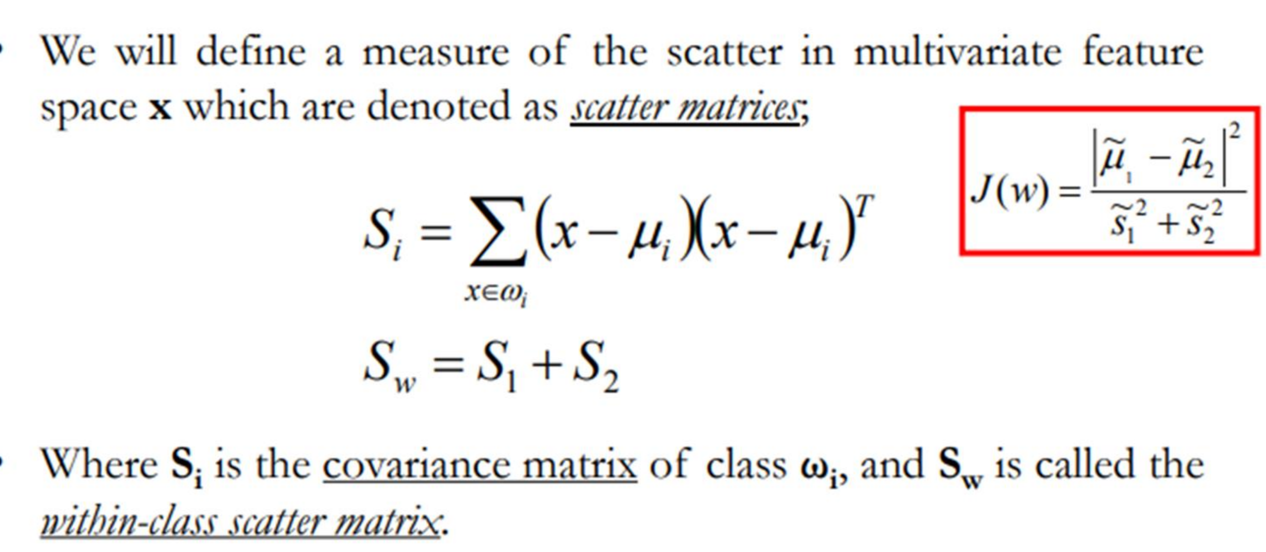
Two criteria are used by LDA to create a new axis:

- Maximize the distance between means of the two classes.
- Minimize the variation within each class.

Of all the possible lines we would like to select the one that maximizes the separability of the scalars.

LDA working





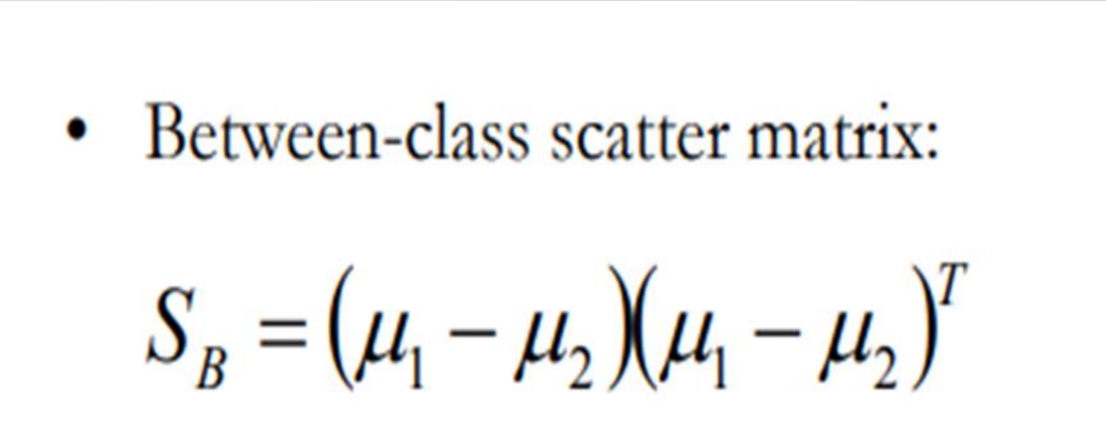
- We will define a measure of the scatter in multivariate feature space **x** which are denoted as ***scatter matrices***;

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

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

$$S_w = S_1 + S_2$$

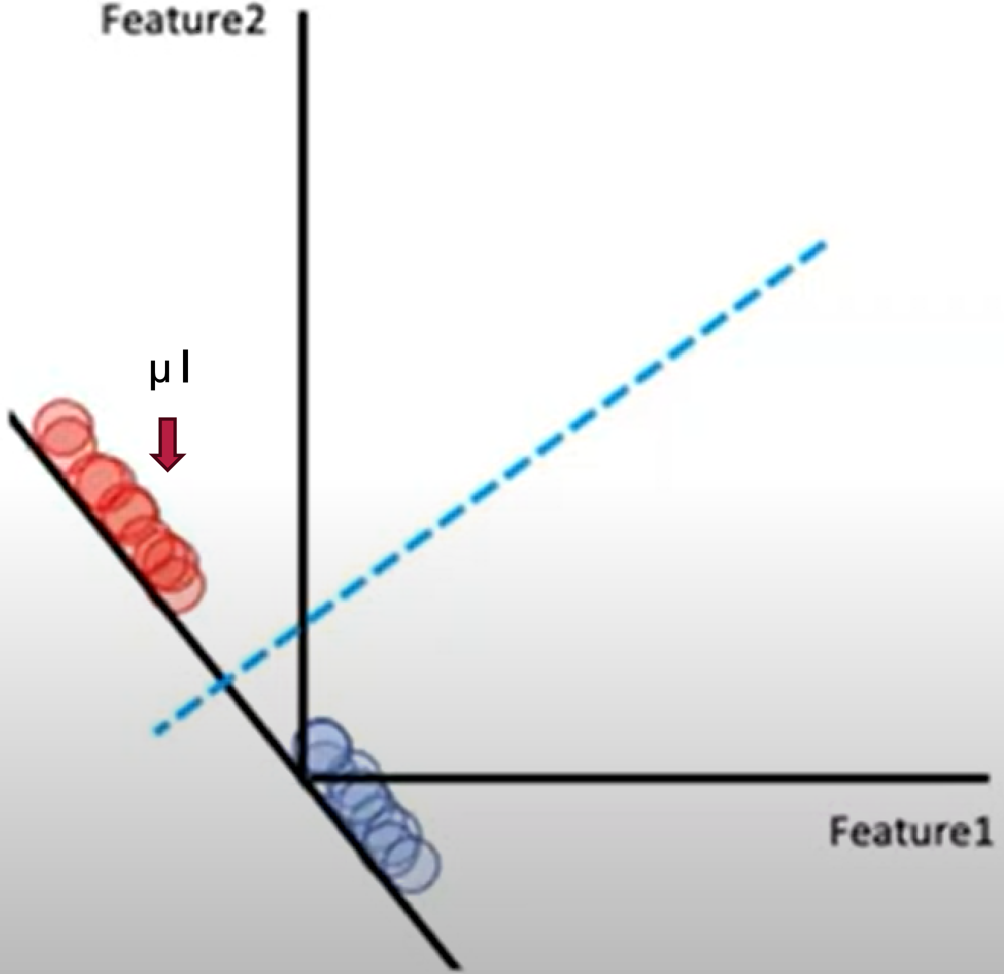
- Where $\mathbf{S}_\mathbf{i}$ is the covariance matrix of class $\boldsymbol{\omega}_\mathbf{i}$, and $\mathbf{S}_\mathbf{w}$ is called the *within-class scatter matrix*.



- Between-class scatter matrix:

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

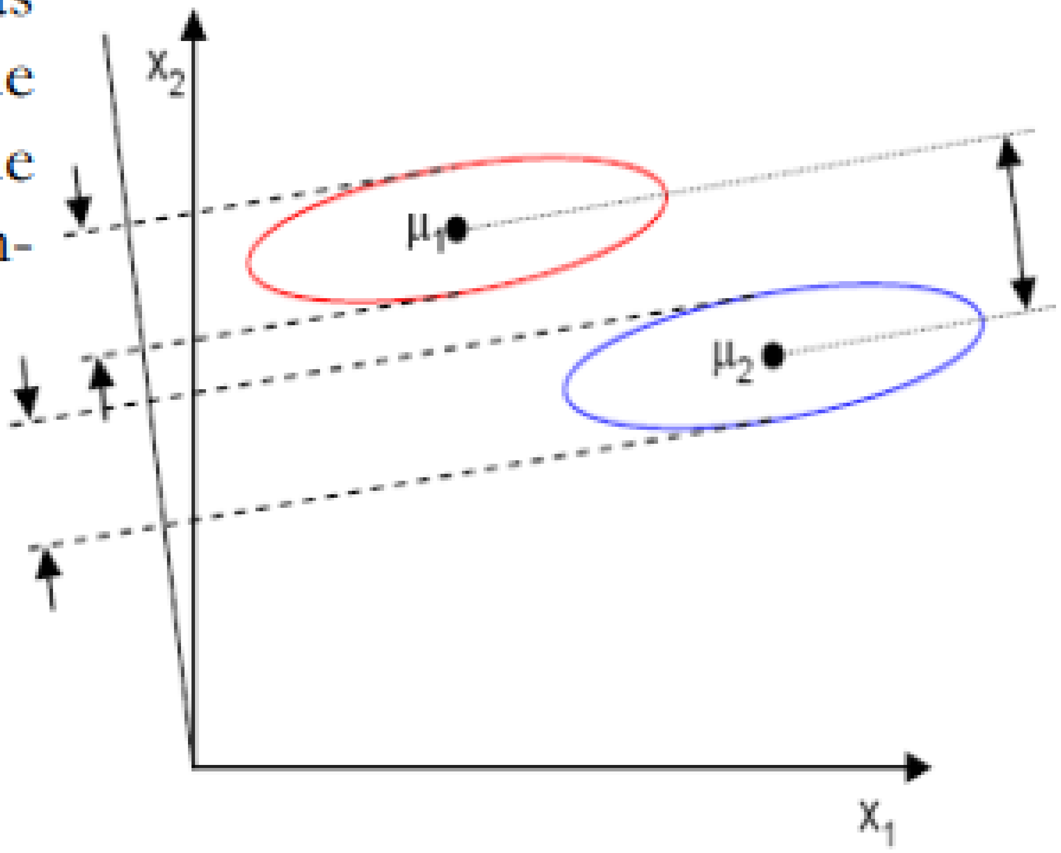
A good classifier will try to keep μ_1 & μ_2 apart while keeping s_1 & s_2 low.



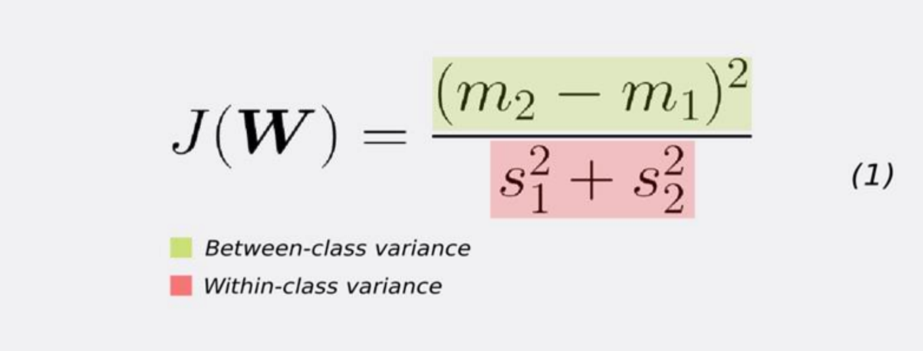
$$\max\left[\frac{(\mu_1 - \mu_2)^2}{s_1{}^2 + s_2{}^2}\right]$$

- The Fisher linear discriminant is defined as the linear function $\mathbf{w}^{\mathrm{T}}\mathbf{x}$ that maximizes the criterion function: (the distance between the projected means normalized by the within-class scatter of the projected samples.

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$



- Therefore, we will be looking for a projection where examples from the same class are projected very close to each other and, at the same time, the projected means are as farther apart as possible



$$J(\mathbf{W}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \qquad (1)$$

🟩 Between-class variance

🟥 Within-class variance

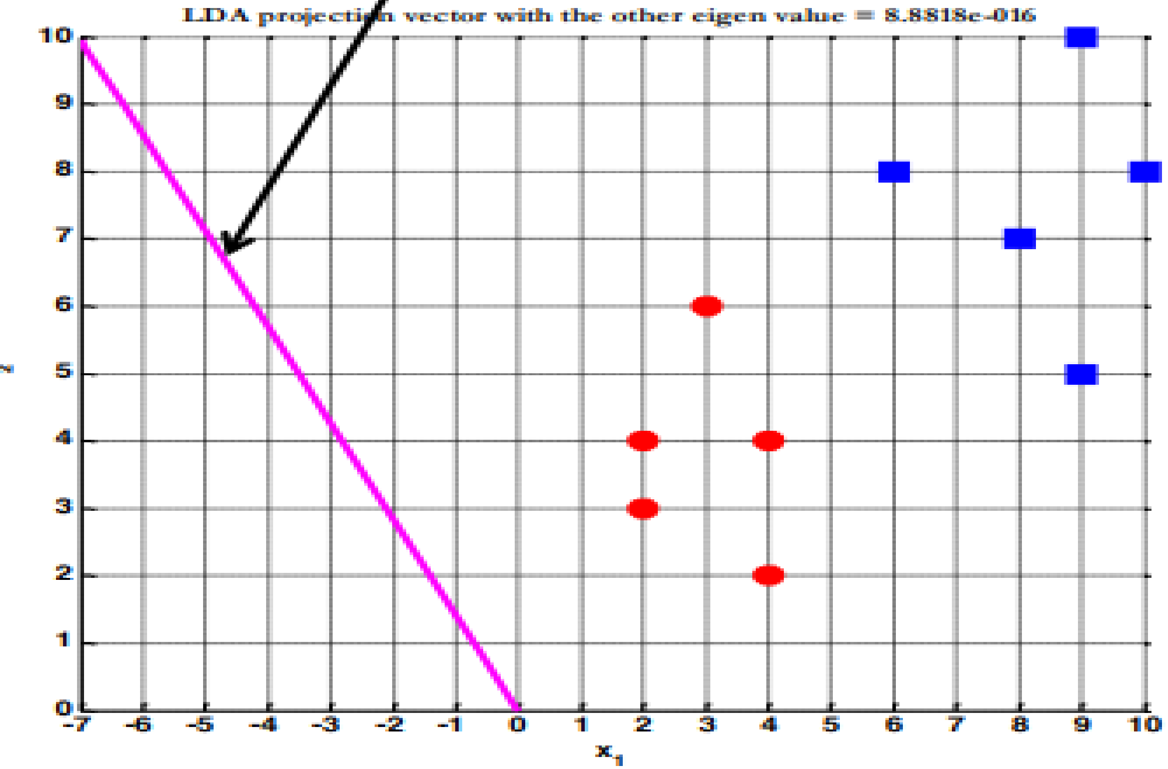
STEPS OF LDA

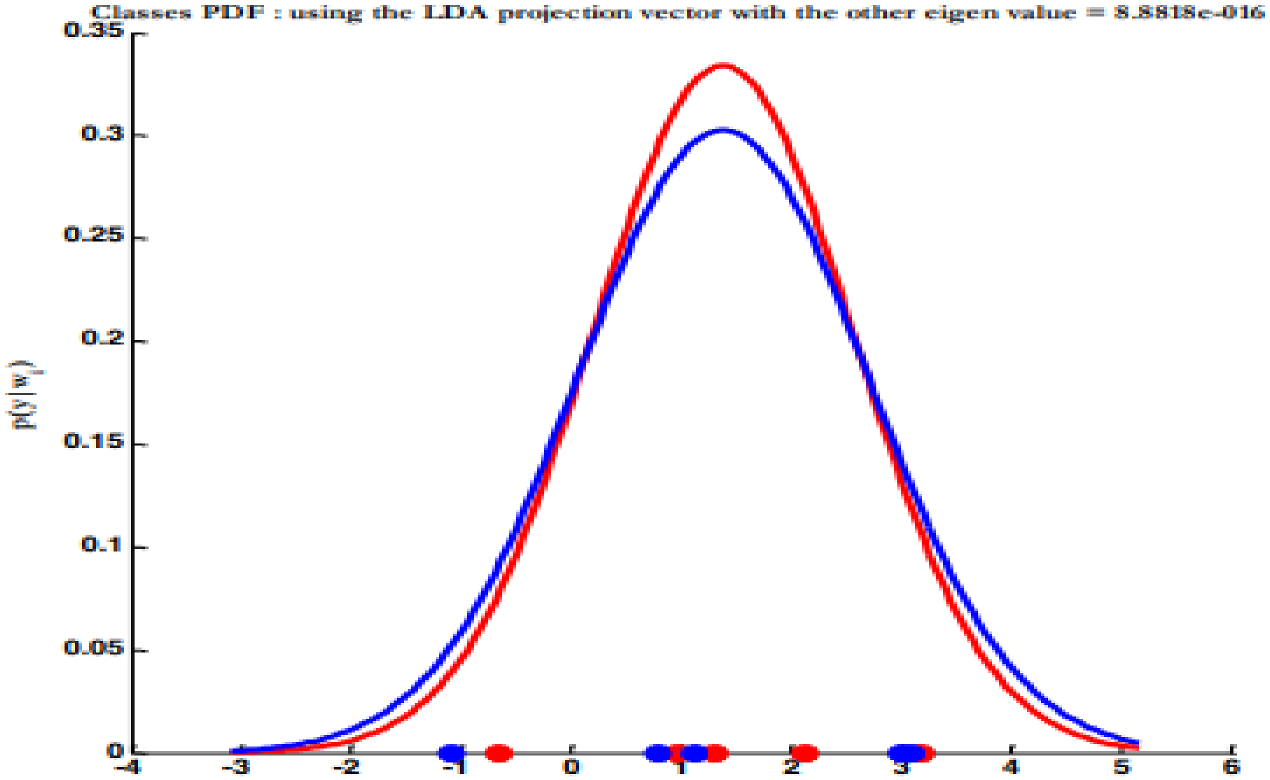
5 steps:

1) Compute the mean vectors for the different classes from the dataset.
2) Compute the scatter matrices (***between-class and within-class scatter matrices***).
3) Compute the eigenvectors and corresponding eigenvalues for the scatter matrices.
4) Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues.
5) Use this eigenvector matrix to transform the samples onto the new subspace.

LDA - Projection

The projection vector corresponding to the **smallest** eigen value

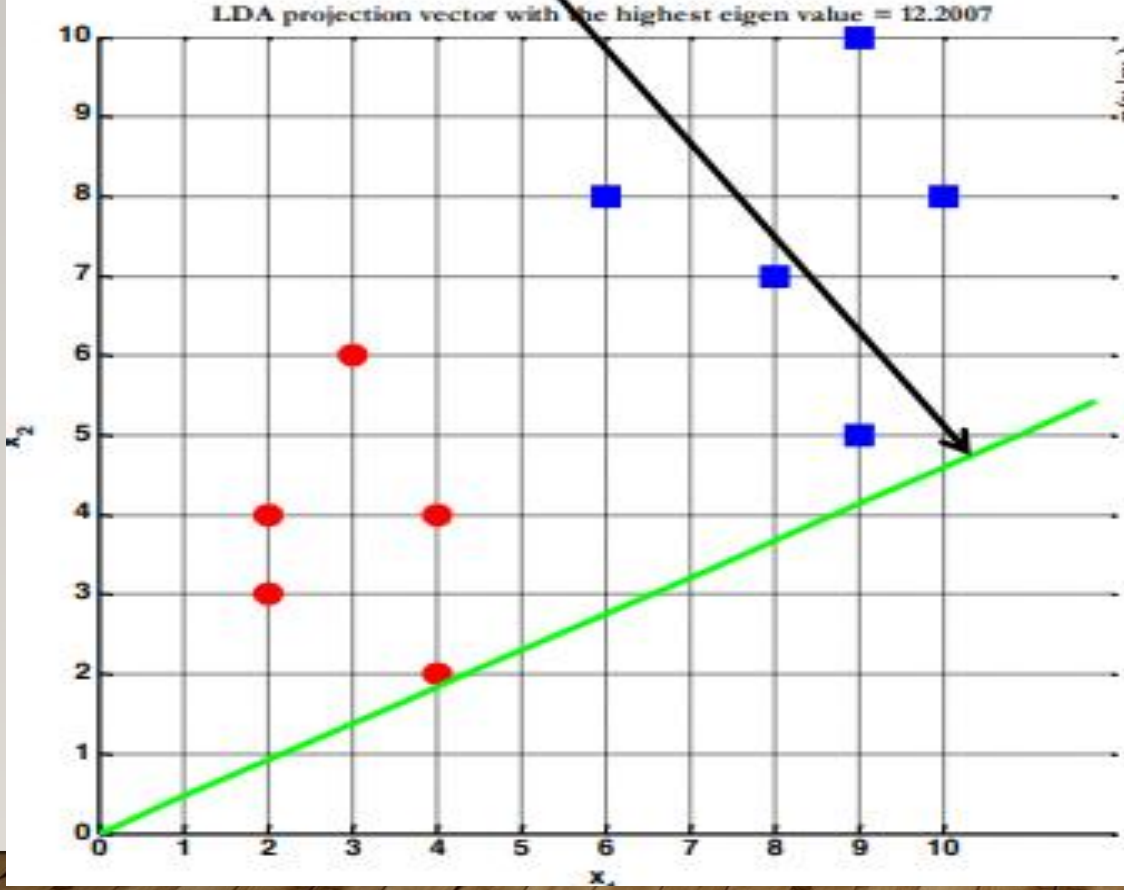




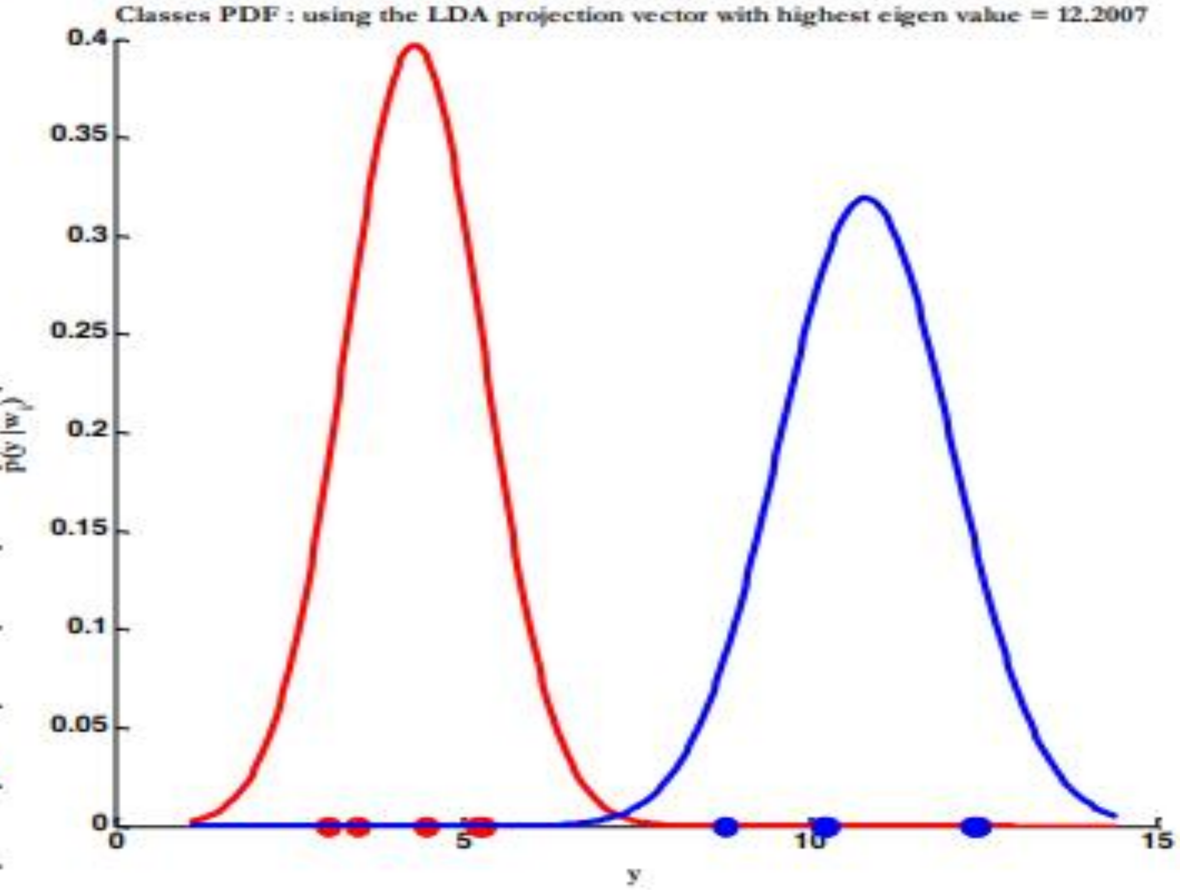
Using this vector leads to **bad separability** between the two classes

LDA - Projection

The projection vector corresponding to the **highest** eigen value



LDA projection vector with the highest eigen value = 12.2007

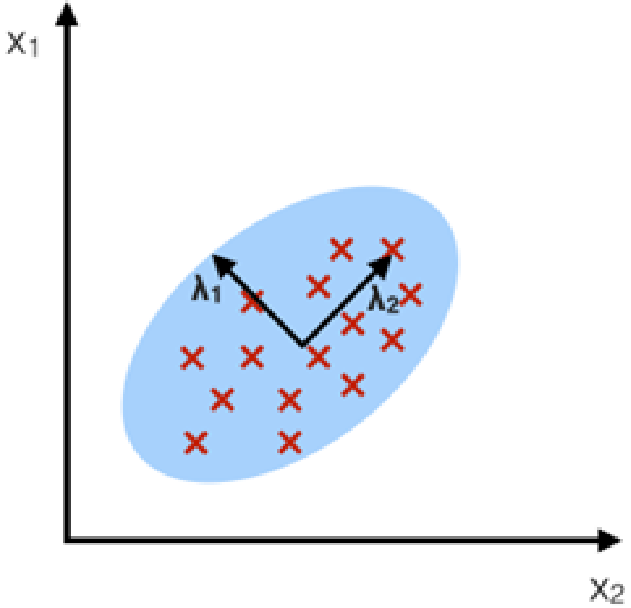


Classes PDF : using the LDA projection vector with highest eigen value = 12.2007

Using this vector leads to **good separability** between the two classes

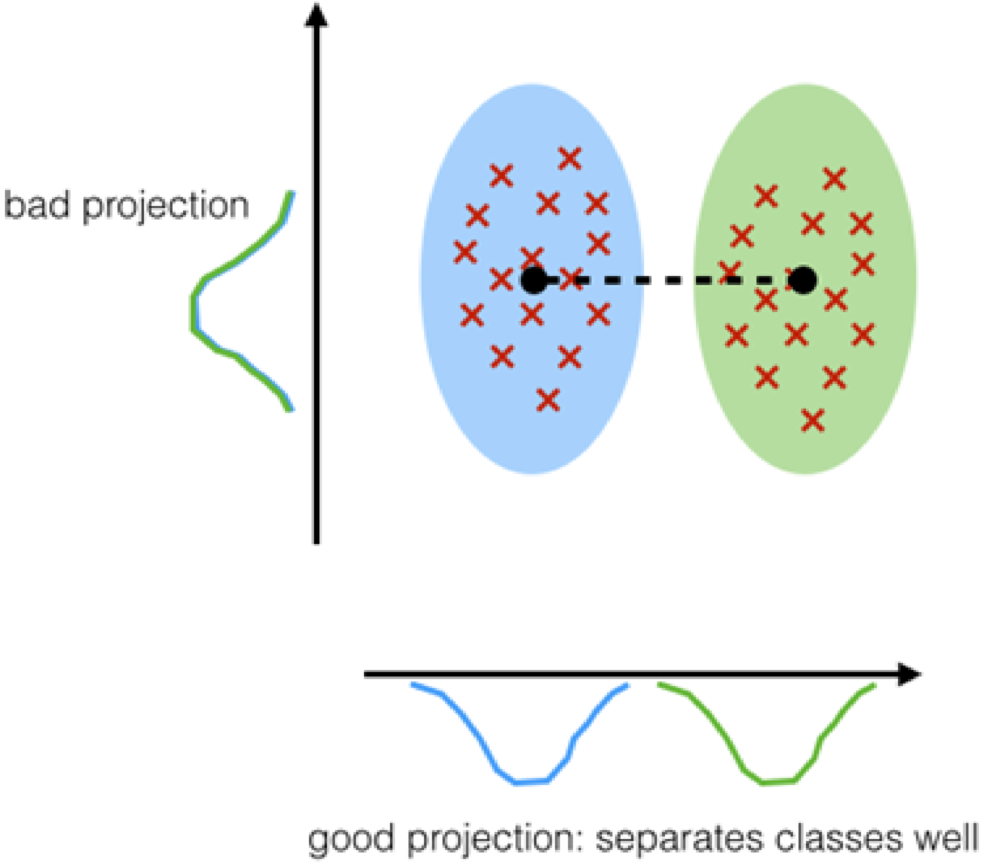
PCA:

component axes that maximize the variance



LDA:

maximizing the component axes for class-separation



LDA:

- LDA is a method ***that transforms high-dimensional data into a lower-dimensional space while maximizing the separation between classes***.
- Unlike *Principal Component Analysis (PCA), which focuses on preserving variance*, ***LDA aims to maximize class differences.***
- the main purpose of LDA is to find the line (or plane) that best separates data points belonging to different classes.
- The key idea behind LDA is that the decision boundary should be chosen such that it ***maximizes the distance between the means of the two classes***

EXTENSIONS OF LDA

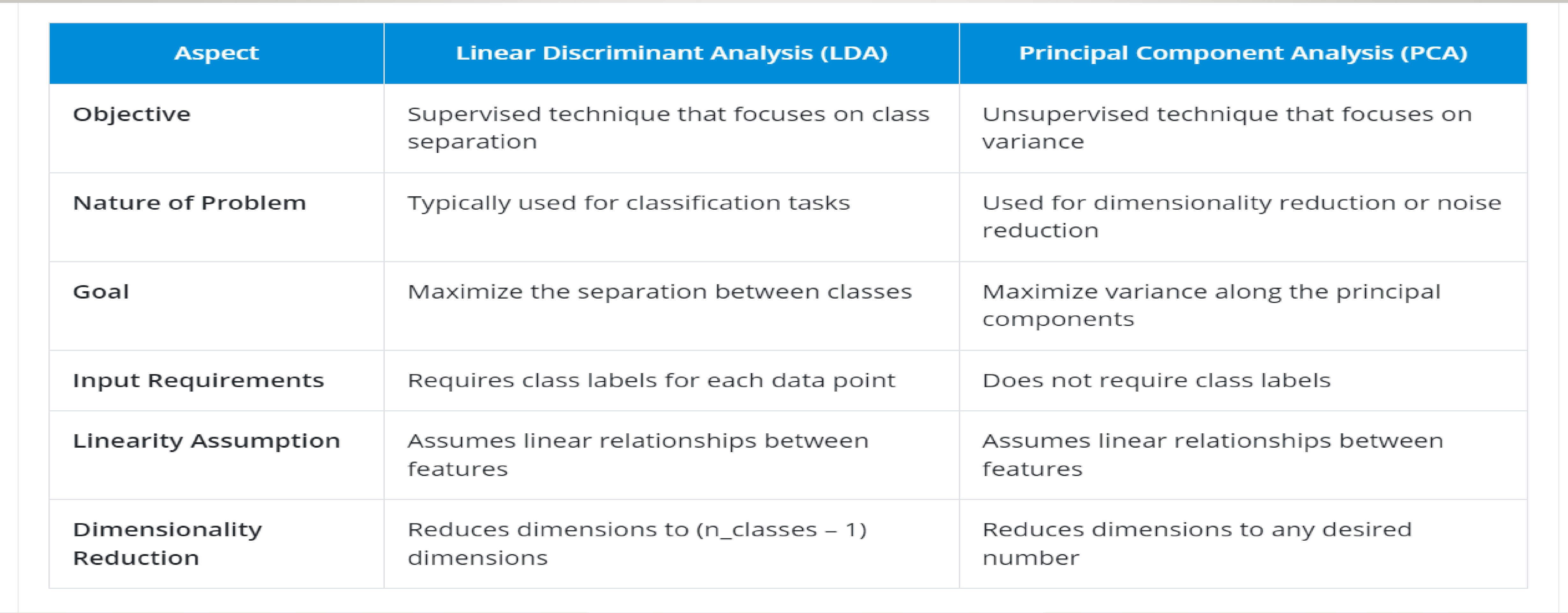
- LDA is an effective and simple method of solving classification problems. It has several variations and extensions, some of which are as follows:

1) Q.D.A. (Quadratic Discriminant Analysis)
2) F.D.A. (Flexible Discriminant Analysis)
3) R.D.A. (Regularized Discriminant Analysis)

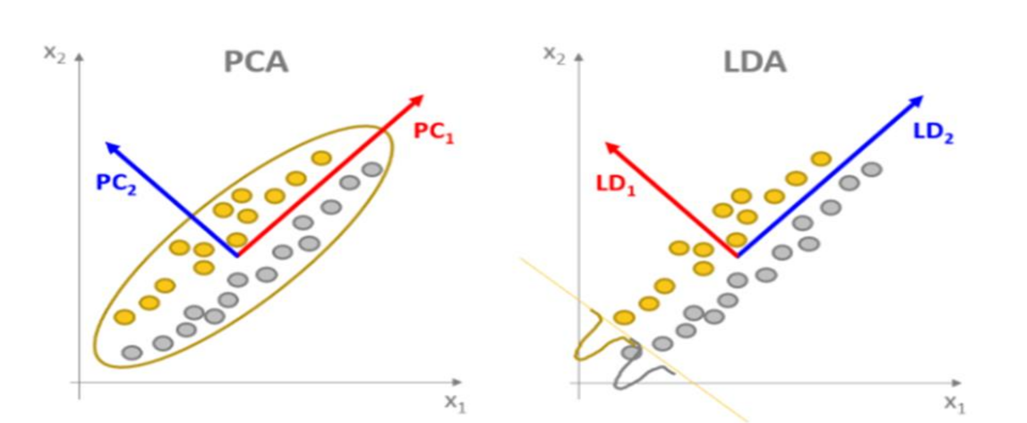
REAL-LIFE APPLICATIONS OF LDA

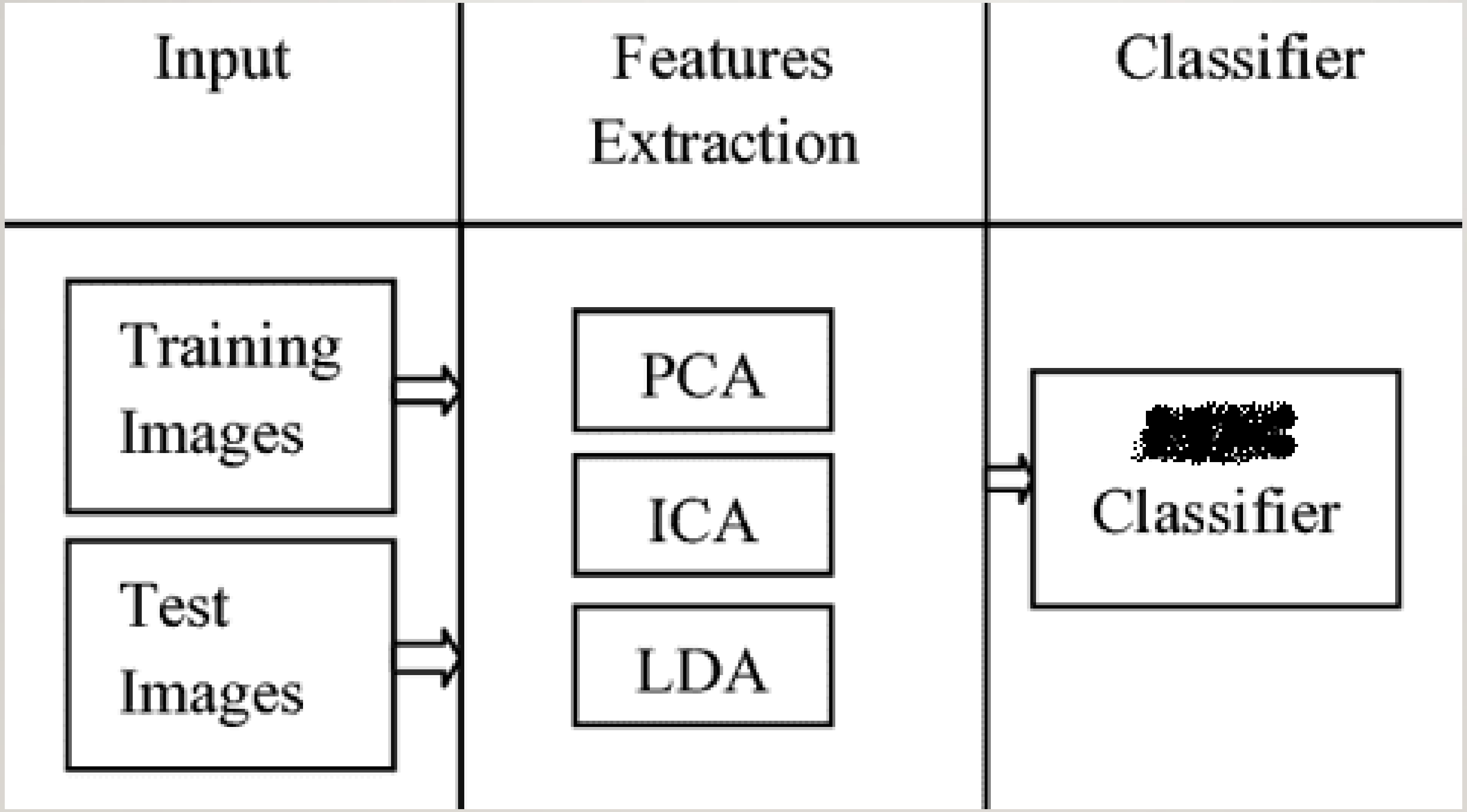
- Face Recognition – LDA is used in face recognition to reduce the number of attributes to a more manageable number before the actual classification.The dimensions that are generated are a linear combination of pixels that forms a template.These are called Fisher's faces.

- Medical – You can use LDA to classify the patient disease as mild, moderate or severe.The classification is done upon the various parameters of the patient and his medical trajectory.



Aspect	Linear Discriminant Analysis (LDA)	Principal Component Analysis (PCA)
Objective	Supervised technique that focuses on class separation	Unsupervised technique that focuses on variance
Nature of Problem	Typically used for classification tasks	Used for dimensionality reduction or noise reduction
Goal	Maximize the separation between classes	Maximize variance along the principal components
Input Requirements	Requires class labels for each data point	Does not require class labels
Linearity Assumption	Assumes linear relationships between features	Assumes linear relationships between features
Dimensionality Reduction	Reduces dimensions to (n_classes – 1) dimensions	Reduces dimensions to any desired number





ICA : independent component analysis

YOU TUBE VIDEOS

- https://www.youtube.com/watch?v=asWoSpdLxTU
- https://www.youtube.com/watch?v=9ruQ3EWxpaE

- https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning