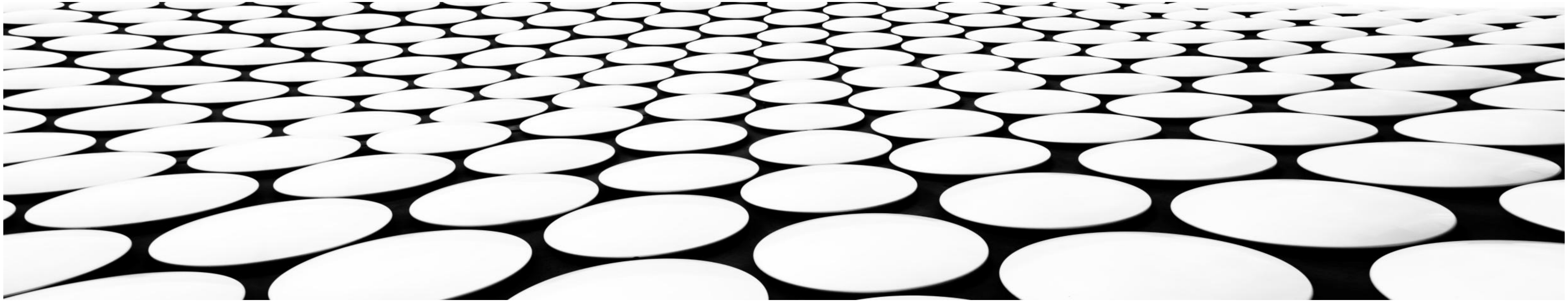
# DATA MINING AND PREDICTIVE DATA ANALYTICS

## CHAPTER-4

# DIMENSION REDUCTION METHOD

BY
*Dr. Sarada Prasanna Pati*
*Professor, Dept. of CSE*



# INTRODUCTION

- Key Issues with Using Too Many Predictor Variables
  - **Complex Interpretation:**
    - The use of too many predictor variables (features) to model a relationship with a target variable can unnecessarily complicate the interpretation of the analysis.
  - **Risk of Overfitting:**
    - The model may fit the training data very well but fail to perform on new data, reducing its general usefulness.
  - **Missed Underlying Patterns:**
    - Looking at each variable separately might overlook deeper relationships or common patterns among variables.

# INTRODUCTION

- **Natural Grouping of Predictors:**
  - Several predictors might fall naturally into a single group (a factor or a **component**) that addresses a single aspect of the data.
- **Example:**
  - Variables such as **savings account balance**, **checking account balance**, **home equity**, **stock portfolio value**, and **401K balance** can all be grouped under one component — **“Assets.”**

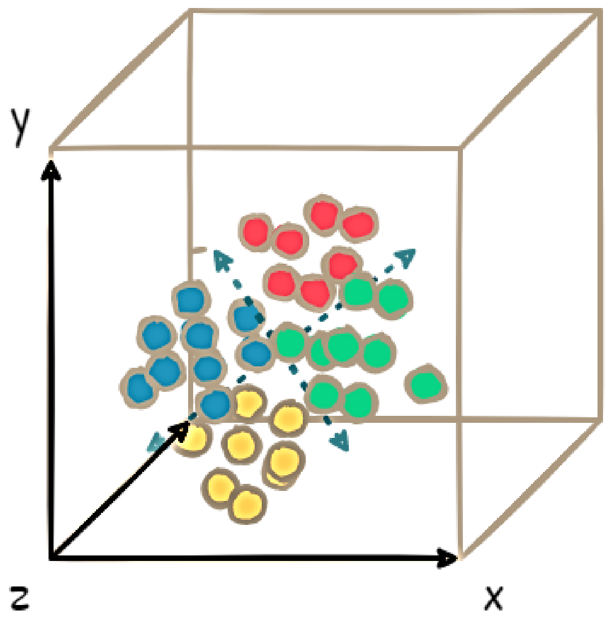
# CURSE OF DIMENSIONALITY

- As the number of dimensions (features) increases:
  - The volume of the space grows exponentially.
  - Data becomes sparse — points are far apart.
  - Distance measures (like Euclidean distance) lose meaning.
  - Models require exponentially more data to achieve the same level of accuracy.
- **Key Problems**
  - Increased computational cost
  - Overfitting due to too many irrelevant features
  - Poor generalization
- **Mitigation**
  - Dimensionality Reduction (PCA, Autoencoders)
  - Feature Selection

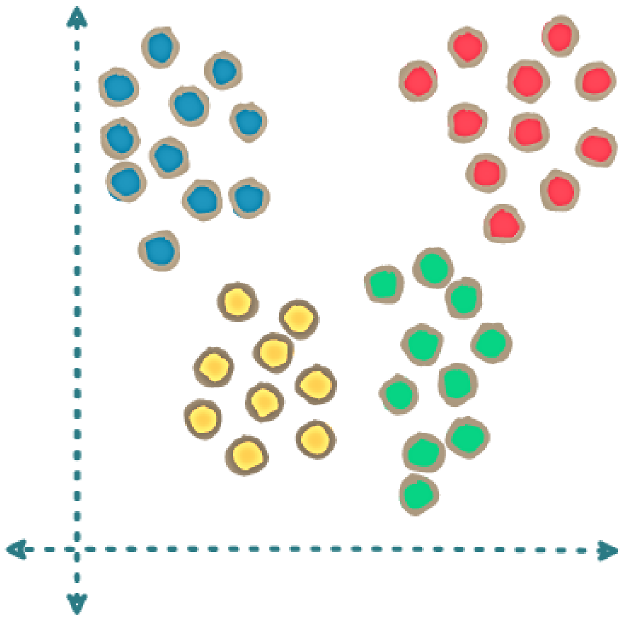
# DIMENSIONALITY REDUCTION

- Dimension reduction methods have the goal of **using the correlation structure** among the predictor variables to accomplish the following:
  - To reduce the number of predictor components
  - To help ensure that these components are independent
  - To provide a framework for interpretability of the results
- Methods
  - **Linear:** PCA, LDA
    - Project data to lower-dimensional linear subspace
  - **Non-linear:** t-SNE, UMAP, Kernel PCA
    - Preserve local structure of complex manifolds

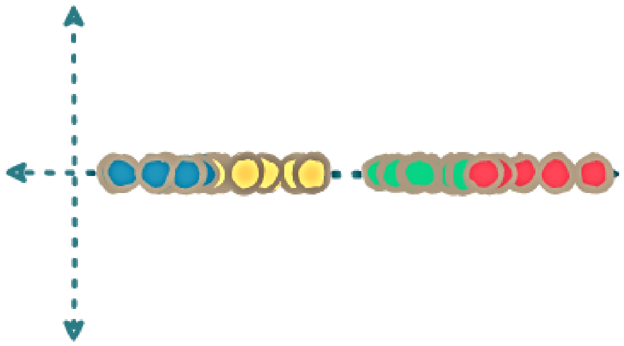
# DIMENSIONALITY REDUCTION




3D



2D



1D

# PRINCIPAL COMPONENTS ANALYSIS

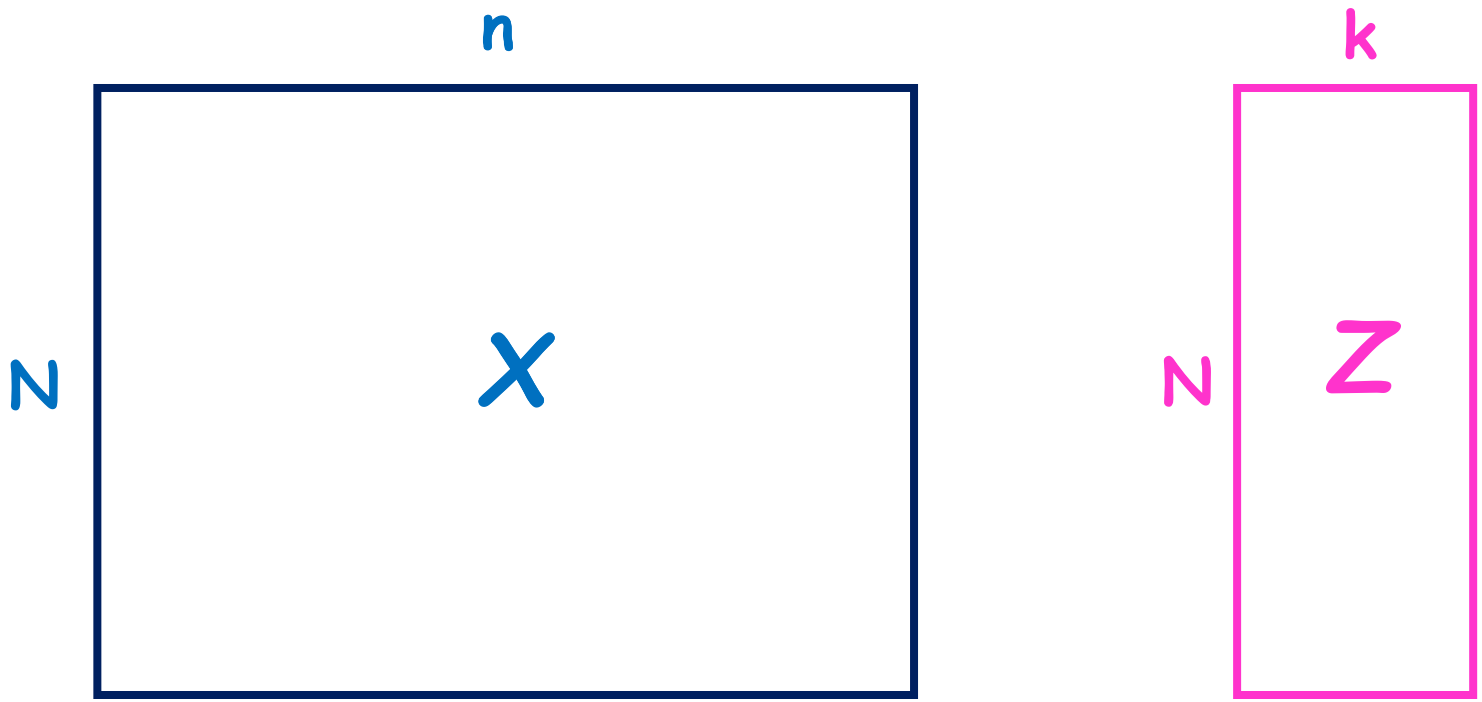
- Principal components analysis (PCA) seeks to explain the **correlation structure** of a **set of predictor variables** using a **smaller set of linear combinations** of these variables.
- These linear combinations are called **components**.
- In simple words
  - PCA takes many related variables and combines them into a smaller number of new variables (called principal components) that still capture most of the important information and relationships in the original data.

# PRINCIPAL COMPONENT ANALYSIS (PCA)

- Takes a data matrix of **N** objects by **n** variables, which may be correlated,
- Summarizes it by uncorrelated axes (principal components or principal axes) that are linear combinations of the original **n** variables
- The first **k** components display as much as possible of the variation among objects.

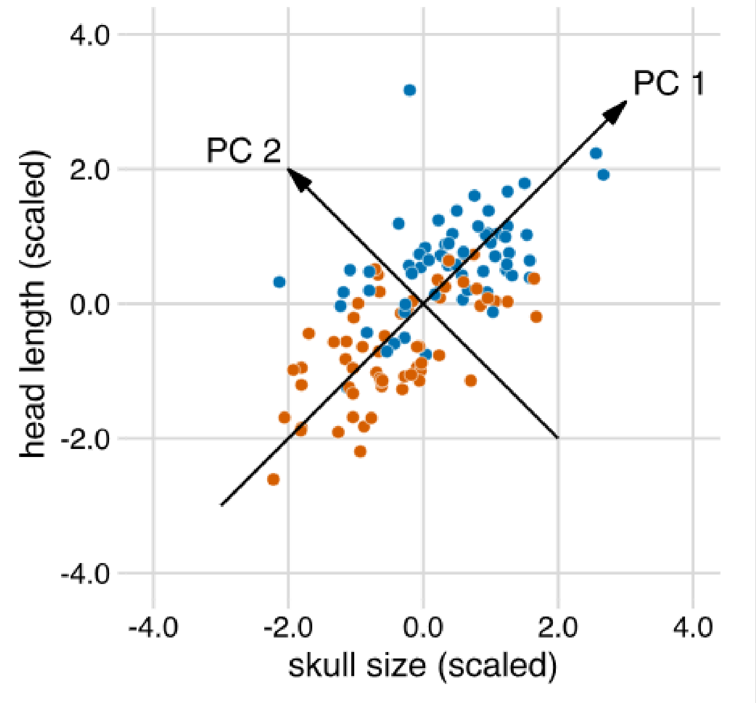
# DATA REDUCTION

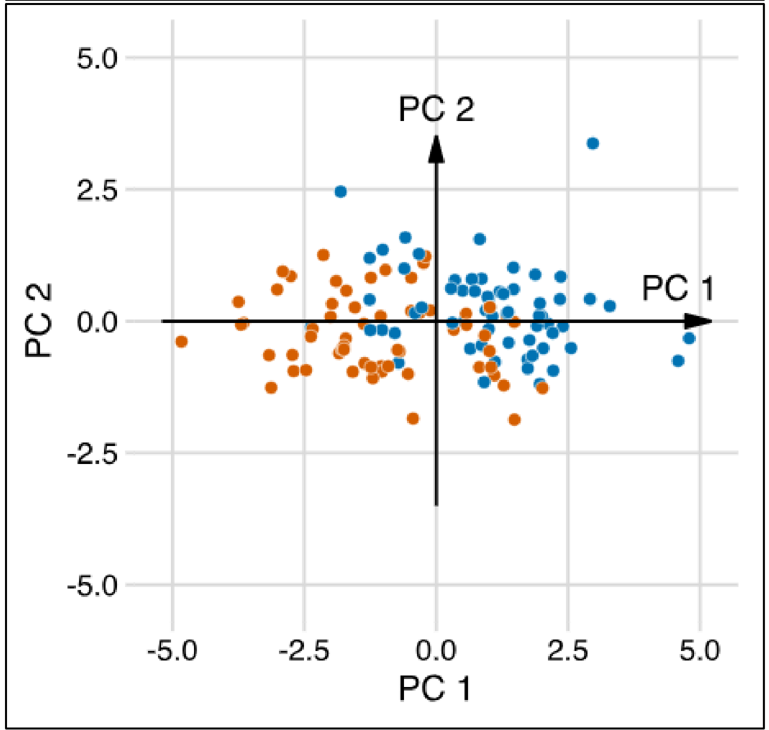
- Summarization of data with many (n) variables by a smaller set of (k) derived (synthetic, composite) variables.



# PCA GEOMETRICAL INTERPRETATION

- The variance along the new PC1 axis is much greater than the variance along the new PC2 axis.
- **Variance Explained:**
  - PC1 now captures most of the variability in the data, as the data spreads widely along it. PC2 captures the remaining, smaller amount of variability.
- **Dimensionality Reduction Potential:**
  - Applying PCA on a dataset with many more dimensions, you might find that a small number of principal components (e.g., PC1 and PC2 here) capture a significant percentage of the total variance.
  - This allows for dimensionality reduction, where you can represent the data effectively using fewer dimensions (the principal components) while retaining most of the important information.



# PCA — STEPWISE EXPLANATION

## 1. Compute the mean vector

- Find the mean of each feature (column) in the dataset.
- This gives you a mean vector ($\bar{X}$ ), representing the average value for each variable.

## 2. Mean-center the data

- Subtract the mean vector from each row of the original data matrix:

$$\boldsymbol{X_c} = \boldsymbol{X} - \bar{\boldsymbol{X}}$$

- This shifts the data so that each feature has a mean of zero.

## 3. Compute the covariance matrix

- Calculate the covariance of the mean-centered data:

$$\mathbf{C = (1\ /\ (n-1))}\ \boldsymbol{X_c}^{\mathbf{T}}\ \boldsymbol{X_c}$$

- This shows how features vary together.

# PCA — STEPWISE EXPLANATION

### 4. Find eigenvalues and eigenvectors of C

- Eigenvalues represent the amount of variance captured by each principal component.
- Eigenvectors define the direction of those components.

### 5. Sort eigenvectors by decreasing eigenvalues

- The eigenvector with the largest eigenvalue corresponds to the most significant principal component.

### 6. Select the top-k eigenvectors

- Choose the first k eigenvectors that capture most of the variance.
- Form the projection matrix W using these eigenvectors.

### 7. Transform the data

- Project the centered data onto the new lower-dimensional space: **Z =** $\boldsymbol{X_c}$ **W**
- Here, Z is the PCA-transformed data, and W is the eigenvector (projection) matrix.

# MATHEMATICAL BACKGROUND (COVARIANCE)

- Given data with $n$ samples and $m$ variables, the **covariance matrix** $C$ summarizes **how variables vary together**.
- Given dataset $X = [x_1, x_2, . , . . x_N]$ with $n$ features:

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

$$\mathrm{Cov}(X) = \mathbf{(1\ /\ (n-1))}\ \boldsymbol{x_c}^{\mathbf{T}}\ \boldsymbol{x_c}$$

- Diagonal entries → variance of each feature
- Off-diagonal entries → covariance between features

# COVARIANCE MATRIX

$$Cov(X) = \begin{bmatrix} Cov(x_1, x_1) & Cov(x_1, x_2) & \cdots & Cov(x_1, x_n) \\ Cov(x_2, x_1) & Cov(x_2, x_2) & \cdots & Cov(x_2, x_n) \\ \cdots & \cdots & \ddots & \vdots \\ Cov(x_n, x_1) & Cov(x_n, x_2) & \cdots & Cov(x_n, x_n) \end{bmatrix}$$

Variances in the diagonal

# EIGENVALUES AND EIGENVECTORS IN PCA

- **Eigenvalues** and **eigenvectors** help us understand how data varies in different directions and how to identify the directions of maximum variance in a dataset.
  - Each **eigenvector** of the covariance matrix points along a principal direction (direction in which the data varies) in the feature space.
  - The eigenvector corresponding to the largest eigenvalue shows the direction in which the data varies most strongly — i.e., where the spread (variance) is maximum.
  - This direction is called the first principal component.
  - The next eigenvector (orthogonal to the first) represents the direction of the next largest variance, and so on.

# EIGENVALUES AND EIGENVECTORS IN PCA

- Each **eigenvalue** quantifies **how much variance** the data has along its corresponding eigenvector (principal axis).
- If your covariance matrix is from standardized data:

  $\lambda$**i** = Variance along principal component *i*.

So, larger eigenvalue → greater spread of data → more “information” captured.

| Term | Meaning | Intuitive role in PCA |
|---|---|---|
| **Eigenvector (Vi)** | Direction in which data spreads | **Principal axis** (orientation) |
| **Eigenvalue ($\lambda$i )** | How much data varies in that direction | **Strength / variance** along that axis |

# EIGENVALUES AND EIGENVECTORS IN PCA

- **Key Takeaways**
  - Eigenvector → Direction of maximum variance in data
  - Eigenvalue → Amount of variance captured along that direction
  - PCA uses eigenvectors of the covariance matrix as principal components
  - Large eigenvalues correspond to meaningful, high-variance directions
  - Small eigenvalues often represent noise and can be discarded

# EIGENVALUE EQUATION

- The eigenvalue equation is:

$$Av = \lambda v$$

- where
  - $v$= eigenvector
  - $\lambda$= eigenvalue
- In words: multiplying $A$ by vector $v$ stretches or shrinks it, but doesn’t rotate it.

# EIGENVALUES AND EIGENVECTORS IN PCA

- In PCA, the matrix $A$ is the **covariance matrix** $C$:

$$C = \frac{1}{n-1}(X-\mu)^{\top}(X-\mu)$$

- Then,

$$Cv_i = \lambda_i v_i$$

  - $v_i$ :direction of maximum variance (Principal Component)
  - $\lambda_i$ :amount of variance captured along that direction

- **The Principal Components can be found out by finding the eigenvectors of the Covariance Matrix.**

# NUMERICAL EXAMPLE

- Dataset (3 features → reduce to 1)

| Sample | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| 1 | 2.5 | 2.4 | 1.5 |
| 2 | 0.5 | 0.7 | 0.2 |
| 3 | 2.2 | 2.9 | 1.8 |
| 4 | 1.9 | 2.2 | 1.0 |
| 5 | 3.1 | 3.0 | 2.0 |

$N = 5$

$n = 3$

# STEP 1 – COMPUTE THE MEAN AND CENTER THE DATA

- $\bar{X}_1 = 2.04, \bar{X}_2 = 2.24, \bar{X}_3 = 1.30$

| Sample | $X_1$ | $X_2$ | $X_3$ | $X_1 - \bar{X}_1$ | $X_2 - \bar{X}_2$ | $X_3 - \bar{X}_3$ |
|---|---|---|---|---|---|---|
| 1 | 2.5 | 2.4 | 1.5 | 0.46 | 0.16 | 0.20 |
| 2 | 0.5 | 0.7 | 0.2 | -1.54 | -1.54 | -1.10 |
| 3 | 2.2 | 2.9 | 1.8 | 0.16 | 0.66 | 0.50 |
| 4 | 1.9 | 2.2 | 1.0 | -0.14 | -0.04 | -0.30 |
| 5 | 3.1 | 3.0 | 2.0 | 0.76 | 0.76 | 0.70 |

# STEP 1 – COMPUTE THE MEAN AND CENTER THE DATA

- Centered matrix $X_c = X - \bar{X}$:

$$X_c = \begin{bmatrix} 0.46 & 0.16 & 0.20 \\ -1.54 & -1.54 & -1.10 \\ 0.16 & 0.66 & 0.50 \\ -0.14 & -0.04 & -0.30 \\ 1.06 & 0.76 & 0.70 \end{bmatrix}$$

# STEP 2 – FIND COVARIANCE MATRIX

$$C = \frac{1}{n-1} X_c^T X_c$$

$$\Rightarrow C = \frac{1}{5-1} \begin{bmatrix} 0.46 & 0.16 & 0.20 \\ -1.54 & -1.54 & -1.10 \\ 0.16 & 0.66 & 0.50 \\ -0.14 & -0.04 & -0.30 \\ 1.06 & 0.76 & 0.70 \end{bmatrix}^T \begin{bmatrix} 0.46 & 0.16 & 0.20 \\ -1.54 & -1.54 & -1.10 \\ 0.16 & 0.66 & 0.50 \\ -0.14 & -0.04 & -0.30 \\ 1.06 & 0.76 & 0.70 \end{bmatrix}$$

$$\Rightarrow C = \begin{bmatrix} 0.9380 & 0.8405 & 0.6625 \\ 0.8405 & 0.8530 & 0.6500 \\ 0.6625 & 0.6500 & 0.5200 \end{bmatrix}$$

# STEP 3 – FIND EIGEN VALUES

- Set $C - \lambda I = 0$

$$\Rightarrow \begin{bmatrix} 0.9380 & 0.8405 & 0.6625 \\ 0.8405 & 0.8530 & 0.6500 \\ 0.6625 & 0.6500 & 0.5200 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = 0$$

$$\Rightarrow \begin{bmatrix} 0.9380 - \lambda & 0.8405 & 0.6625 \\ 0.8405 & 0.8530 - \lambda & 0.6500 \\ 0.6625 & 0.6500 & 0.5200 - \lambda \end{bmatrix} = 0$$

$$\Rightarrow -\lambda^3 + 2.311\lambda^2 + 0.1637\lambda - 0.0018 = 0 \Rightarrow \begin{cases} \lambda_1 \approx 2.228 \\ \lambda_2 \approx 0.078 \\ \lambda_3 \approx 0.004 \end{cases}$$

Eigen Value 1

Eigen Value 2

Eigen Value 3

# STEP 4 – FINDING EIGENVECTORS

- Once you have an eigenvalue ($\lambda$), you can substitute it back into the equation $(C - \lambda I)\mathbf{v} = \mathbf{0}$ and solve for the components of the eigenvector $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$.

# STEP 4 – FINDING EIGENVECTORS

For the first eigen value we set $(C - \lambda_1 I)\mathbf{v_1} = \mathbf{0}$

$$\Longrightarrow \begin{bmatrix} 0.9380 - 2.228 & 0.8405 & 0.6625 \\ 0.8405 & 0.8530 - 2.228 & 0.6500 \\ 0.6625 & 0.6500 & 0.5200 - 2.228 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\Longrightarrow \begin{bmatrix} -1.290 & 0.8405 & 0.6625 \\ 0.8405 & -1.3750 & 0.6500 \\ 0.6625 & 0.6500 & -1.708 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} -1.29v_1 + 0.8405v_2 + 0.6625v_3 &= 0 \\ 0.8405v_1 - 1.3750v_2 + 0.6500v_3 &= 0 \\ 0.6625v_1 + 0.6500v_2 - 1.708v_3 &= 0 \end{aligned} \quad \Longrightarrow \mathbf{v_1} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0.603} \\ \mathbf{0.579} \\ \mathbf{0.548} \end{bmatrix}$$

# STEP 4 – FINDING EIGENVECTORS

- For $\lambda_1 \approx 2.228,\ \mathbf{v_1} = \begin{bmatrix} \mathbf{0.603} \\ \mathbf{0.579} \\ \mathbf{0.548} \end{bmatrix}$
- For $\lambda_2 \approx 0.078,\ \mathbf{v_2} = \begin{bmatrix} \mathbf{-0.520} \\ \mathbf{0.224} \\ \mathbf{-0.825} \end{bmatrix}$
- For $\lambda_3 \approx 0.004,\ \mathbf{v_3} = \begin{bmatrix} \mathbf{0.602} \\ \mathbf{-0.785} \\ \mathbf{0.155} \end{bmatrix}$

# STEP 5 – ORDER EIGENVALUES AND EIGENVECTORS

- They are already ordered from largest to smallest. The eigenvalue represents the amount of variance explained by its corresponding principal component.
  - $\boldsymbol{\lambda_1 = 2.228}$
  - $\boldsymbol{\lambda_2 = 0.078}$
  - $\boldsymbol{\lambda_3 = 0.004}$
- **Total variance:** $\lambda_1 + \lambda_2 + \lambda_3 = 2.228 + 0.078 + 0.004 = 2.31$
- **Proportion of variance explained by each component:**
  - **PC1:** $\boldsymbol{\frac{2.228}{2.31} \approx 0.9645}$ **(96.45%) (explains the most variance)**
  - **PC2:** $\boldsymbol{\frac{0.078}{2.31} \approx 0.0338}$ **(3.38%)**
  - **PC3:** $\boldsymbol{\frac{0.004}{2.31} \approx 0.0017}$ **(0.17%)**

# STEP 6 – SELECT PRINCIPAL COMPONENTS

- We can choose to retain components that explain a significant amount of variance.
- Often, a threshold like 85% or 90% is used.
- In this case, **PC1** alone explains over 96% of the variance, so we might choose to keep only **PC1**.
- If we want to capture almost all the variance, we would keep **PC1** and **PC2**.
- Let's assume we want to reduce the dimensionality to 1 principal component (**PC1**). The projection matrix (P) would be the eigenvector corresponding to $\lambda_1$

$$P = v_1 = \begin{bmatrix} 0.603 \\ 0.579 \\ 0.548 \end{bmatrix}$$

# STEP 7 – TRANSFORM THE DATA

- To transform the original centered data into the new principal component space, we multiply the centered data matrix by the selected eigenvector(s).
- If we choose to keep only **PC1**, the transformed data (scores) for each sample will be:

$$Z = X_{centered} \cdot \mathrm{P}$$

$$\Rightarrow Z = \begin{bmatrix} 0.480 \\ -2.423 \\ 0.753 \\ -0.272 \\ 1.463 \end{bmatrix}$$