Unit-4
Dimensionality Reduction

Syllabus

- **Dimensionality Reduction:**

  Singular Value Decomposition

  Principal Component Analysis

  Linear Discriminated Analysis

Characteristic Polynomial and characteristic equation
and
Eigen Values and Eigen Vectors

Computation for 2x2 and 3x3 Square Matrix

Eigen Values and Eigen Vectors

Definition
Let A be an $n \times n$ matrix. A *scalar* λ is called an **eigenvalue** of A if there exists a nonzero vector $\mathbf{x}$ in $\mathbf{R}^n$ such that

$$A\mathbf{x} = \lambda\mathbf{x}.$$

The vector $\mathbf{x}$ is called an **eigenvector** corresponding to λ.

The eigenvectors **x** and eigenvalues λ of a matrix A satisfy

$$\mathbf{Ax = \lambda x}$$

If A is an n x n matrix, then **x** is an n x 1 vector, and λ is a constant.

The equation can be rewritten as $(A - \lambda I)\,\mathbf{x} = 0$, where I is the n x n identity matrix.

Solving the equation $|A - \lambda I_n| = 0$ for λ leads to all the eigenvalues of A.

On expending the determinant $|A - \lambda I_n|$, we get a polynomial in λ.

This polynomial is called the **characteristic polynomial** of A.

The equation $|A - \lambda I_n| = 0$ is called the **characteristic equation** of A.

2 X 2 Example : Compute Eigen Values

$A = \begin{bmatrix} 1 & -2 \\ 3 & -4 \end{bmatrix}$ so $A - \lambda I = \begin{bmatrix} 1 - \lambda & -2 \\ 3 & -4 - \lambda \end{bmatrix}$

$$\det(A - \lambda I) = (1 - \lambda)(-4 - \lambda) - (3)(-2)$$
$$= \lambda^2 + 3\lambda + 2$$

Set $\lambda^2 + 3\lambda + 2$ to 0

Then = λ = (-3 +/- sqrt(9-8))/2

So the two values of λ are -1 and -2.

Example 1: Find the eigenvalues and eigenvectors of the matrix

$$A = \begin{bmatrix} -4 & -6 \\ 3 & 5 \end{bmatrix}$$

Solution Let us first derive the characteristic polynomial of A.

We get

$$A - \lambda I_2 = \begin{bmatrix} -4 & -6 \\ 3 & 5 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -4-\lambda & -6 \\ 3 & 5-\lambda \end{bmatrix}$$

$$|A - \lambda I_2| = (-4-\lambda)(5-\lambda) + 18 = \lambda^2 - \lambda - 2$$

We now solve the characteristic equation of A.

$$\lambda^2 - \lambda - 2 = 0 \Rightarrow (\lambda - 2)(\lambda + 1) = 0 \Rightarrow \lambda = 2 \text{ or } -1$$

The eigenvalues of A are 2 and –1.

The corresponding eigenvectors are found by using these values of λ in the equation $(A - \lambda I_2)\mathbf{x} = \mathbf{0}$.

There are many eigenvectors corresponding to each eigenvalue.

For $\underline{\lambda = 2}$

We solve the equation $(A - 2I_2)\mathbf{x} = \mathbf{0}$ for $\mathbf{x}$.

The matrix $(A - 2I_2)$ is obtained by subtracting 2 from the diagonal elements of A.

We get

$$\begin{bmatrix} -6 & -6 \\ 3 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{0}$$

This leads to the system of equations

$$\begin{aligned} -6x_1 - 6x_2 &= 0 \\ 3x_1 + 3x_2 &= 0 \end{aligned}$$

giving $x_1 = -x_2$. The solutions to this system of equations are $x_1 = -r$, $x_2 = r$, where r is a scalar. Thus the eigenvectors of A corresponding to $\lambda = 2$ are nonzero vectors of the form

$$\mathbf{v}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_2 \begin{bmatrix} -1 \\ 1 \end{bmatrix} = r \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

For $\underline{\lambda = -1}$

We solve the equation $(A + 1I_2)x = 0$ for x.

The matrix $(A + 1I_2)$ is obtained by adding 1 to the diagonal elements of A. We get

$$\begin{bmatrix} -3 & -6 \\ 3 & 6 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{0}$$

This leads to the system of equations

$$\begin{aligned} -3x_1 - 6x_2 &= 0 \\ 3x_1 + 6x_2 &= 0 \end{aligned}$$

Thus $x_1 = -2x_2$. The solutions to this system of equations are $x_1 = -2s$ and $x_2 = s$, where s is a scalar. Thus the **eigenvectors** of A corresponding to $\lambda = -1$ are nonzero vectors of the form

$$\mathbf{v}_2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_2\begin{bmatrix} -2 \\ 1 \end{bmatrix} = s\begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

Example 3: Eigenvalues and Eigenvectors

Find the eigenvalues and eigenvectors of the matrix

$$A = \begin{bmatrix} 5 & 4 & 2 \\ 4 & 5 & 2 \\ 2 & 2 & 2 \end{bmatrix}$$

Solution The matrix $A - \lambda I_3$ is obtained by subtracting λ from the diagonal elements of A. Thus

$$A - \lambda I_3 = \begin{bmatrix} 5-\lambda & 4 & 2 \\ 4 & 5-\lambda & 2 \\ 2 & 2 & 2-\lambda \end{bmatrix}$$

The characteristic polynomial of A is $|A - \lambda I_3|$. Using row and column operations to simplify determinants, we get

Alternate Solution

$|A - \lambda I_3| = 0$

$A = \begin{bmatrix} 5 & 4 & 2 \\ 4 & 5 & 2 \\ 2 & 2 & 2 \end{bmatrix}$

$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

λ = Eigen values

For 3 * 3 matrix Eigen values can be computed using the following equations

$$\boxed{\lambda^3 - S_1\lambda^2 + S_2\lambda - S_3 = 0}$$

S_1 = Sum of principal diagonal elements

S_2 = Sum of minors of principal diagonal ele

S_3 = Determinant of A

$$A = \begin{array}{c} \begin{matrix} + & - & + \end{matrix} \\ \begin{bmatrix} 5 & 4 & 2 \\ 4 & 5 & 2 \\ 2 & 2 & 2 \end{bmatrix} \end{array}$$

$S_1 = 5+5+2$

$$S_2 = \begin{vmatrix} 5 & 2 \\ 2 & 2 \end{vmatrix} + \begin{vmatrix} 5 & 2 \\ 2 & 2 \end{vmatrix} + 2\begin{vmatrix} 5 & 4 \\ 4 & 5 \end{vmatrix}$$

$= 6+6$

$$S_2 = \begin{vmatrix} 5 & 2 \\ 2 & 2 \end{vmatrix} + \begin{vmatrix} 5 & 2 \\ 2 & 2 \end{vmatrix} + \begin{vmatrix} 5 & 4 \\ 4 & 5 \end{vmatrix}$$

$= (10-4) + (10-4) + (25-16)$

$= 6+6+9$

$= 21$

$\boxed{S_2 = 21}$

$S_3 =$ Determinant of A

$$S_3 = |A| = \left| \begin{array}{c} \begin{matrix} + & - & + \end{matrix} \\ \begin{bmatrix} 5 & 4 & 2 \\ 4 & 5 & 2 \\ 2 & 2 & 2 \end{bmatrix} \end{array} \right|$$

$$= 5 * \begin{vmatrix} 5 & 2 \\ 2 & 2 \end{vmatrix} - 4 * \begin{vmatrix} 4 & 2 \\ 2 & 2 \end{vmatrix} + 2 * \begin{vmatrix} 4 & 5 \\ 2 & 2 \end{vmatrix}$$

$$= 5 * (10-4) - 4 * (8-4) + 2 * (8-10)$$

$$= 5 * 6 - 4 * 4 + 2(-2)$$

$$= 30 - 16 - 4$$

$$\boxed{S_3 = 10}$$

$\lambda^3 - 21\lambda^2 + 32\lambda - 53 = 0$

$\lambda^3 - 12\lambda^2 + 21\lambda - 10 = 0$

Eigen values will be factors of 10

Factors of 10 = 1, 2, 5, 10

check for each one [illegible]

[illegible]

$\lambda = \{10, 1, 1\}$

$\lambda_1 = 10$

$\lambda_2 = 1$

Eigen vector:

$$(A - \lambda I_3) X = 0$$

$A = \begin{bmatrix} 5 & 4 & 2 \\ 4 & 5 & 2 \\ 2 & 2 & 2 \end{bmatrix}$ $\quad \lambda = \{10, 1\}$ $\quad I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$\lambda = 10$

$$\left(\begin{bmatrix} 5 & 4 & 2 \\ 4 & 5 & 2 \\ 2 & 2 & 2 \end{bmatrix} - 10 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\Rightarrow \left(\begin{bmatrix} 5 & 4 & 2 \\ 4 & 5 & 2 \\ 2 & 2 & 2 \end{bmatrix} - \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\Rightarrow \begin{bmatrix} -5 & 4 & 2 \\ 4 & -5 & 2 \\ 2 & 2 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$-5x_1 + 4x_2 + 2x_3 = 0$$
$$4x_1 - 5x_2 + 2x_3 = 0$$
$$2x_1 + 2x_2 + 8x_3 = 0$$

Consider any two equations

$$-5x_1 + 4x_2 + 2x_3 = 0$$
$$4x_1 - 5x_2 + 2x_3 = 0$$

$$\frac{x_1}{(4\times 2)-(-10)} = \frac{-x_2}{-(10)-8} = \frac{x_3}{25-16}$$

$$\frac{x_1}{8+10} = \frac{x_2}{18} = \frac{x_3}{9}$$

$$\frac{x_1}{18} = \frac{x_2}{18} = \frac{x_3}{9} = k$$

$$\Rightarrow \frac{x_1}{2} = \frac{x_2}{2} = \frac{x_3}{1} = 2k = k_1$$

$$\frac{x_1}{2} = \frac{x_2}{2} = \frac{x_3}{1} \; ; \quad x_1 = 2k_1, \; x_2 = 2k_1, \; x_3 = k_1$$

Hence $X_1 = \begin{bmatrix} 2k_1 \\ 2k_1 \\ k_1 \end{bmatrix}$

$$= k_1 \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$$

What is Dimensionality Reduction?

- The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction.

- A dataset contains a huge number of input features in various cases, which makes the predictive modeling task more complicated, for such cases, dimensionality reduction techniques are required to use.

Dimensionality Reduction…?

- Dimensionality reduction technique can be defined as, ***"It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information."***

- These techniques are widely used in Machine Learning for obtaining a better fit predictive model while solving the classification and regression problems.

- Handling the high-dimensional data is very difficult in practice, commonly known as the *curse of dimensionality.*

- **Dimensionality reduction is a data preparation technique performed on data prior to modeling. It might be performed after data cleaning and data scaling and before training a predictive model.**

Benefits of Dimensionality Reduction..

- By reducing the dimensions of the features, the space required to store the dataset also gets reduced.
- Less Computation training time is required for reduced dimensions of features.
- Reduced dimensions of features of the dataset help in visualizing the data quickly.
- It removes the redundant features (if present).

Two ways of Dimensionality Reduction

- 1. Feature Selection
- 2. Feature Extraction

Feature Selection

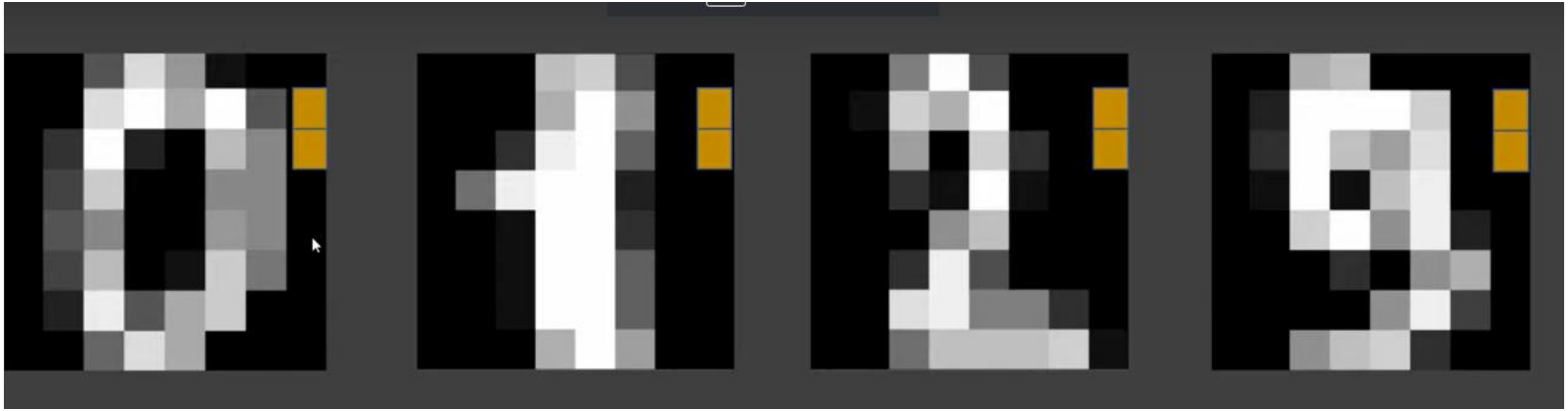
- Feature selection is the process of selecting the subset of the relevant features and leaving out the irrelevant features present in a dataset to build a model of high accuracy. In other words, it is a way of selecting the optimal features from the input dataset.

General – features reduction technique

- In this example number 2 has 64 features… but many of them are of no importance to decide the characteristics of 2, are removed first.

0	0	11	16	9	0	0	0
0	0	13	11	12	0	0	0
0	0	5	0	2	7	0	0
0	0	3	0	4	5	0	0
0	0	0	6	13	4	0	0
0	0	3	3	16	7	0	0
0	0	8	1	3	10	0	0
0	0	7	8	8	8	11	1

Remove features which are of no importance



Feature Selection – 3 Methods

- 1.Filter Method
    - **Correlation**
    - **Chi-Square Test**
    - **ANOVA**
    - **Information Gain, etc.**

- 2.Wrapper Method
    - Forward Selection
    - Backward Selection
    - Bi-directional Elimination

- 3.Embedded Method
    - **LASSO**
    - **Elastic Net**
    - **Ridge Regression, etc.**

Feature Extraction

- Feature extraction is the process of transforming the space containing many dimensions into space with fewer dimensions.

- This approach is useful when we want to keep the whole information but use fewer resources while processing the information.

FEATURE EXTRACTION

This is about **extracting/deriving** information from the original features set to create a new features subspace.

The primary idea behind feature extraction is to compress the data with the goal of maintaining most of the relevant information.

Feature extraction techniques are also used for reducing the number of features from the original features set to reduce model complexity, model overfitting, enhance model computation efficiency and reduce generalization error.

The following are different types of feature extraction techniques:

PCA- Principal Component Analysis

LDA - Linear Discriminant Analysis

Feature selection is a process in machine learning that involves identifying and **selecting the most relevant subset of features** out of the original features in a dataset to be used as inputs for a model.

The goal of feature selection is to improve model performance by reducing the number of irrelevant or redundant features that may introduce noise or bias into the model.

The **key difference** between feature selection and feature extraction techniques used for dimensionality reduction is that while the **original features are maintained** in the case of feature selection algorithms, the feature extraction algorithms **transform the data onto a new feature space**.

Feature selection techniques can be used if the requirement is to **maintain the original features,** unlike the feature extraction techniques which derive useful information from data to construct a new feature subspace.

- **Feature extraction and feature engineering**: transformation of raw data into features suitable for modeling;
- **Feature transformation**: transformation of data to improve the accuracy of the algorithm;
- **Feature selection**: removing unnecessary features. Feature selection is applied either to prevent redundancy and/or irrelevancy existing in the features or just to get a limited number of features to prevent from overfitting.

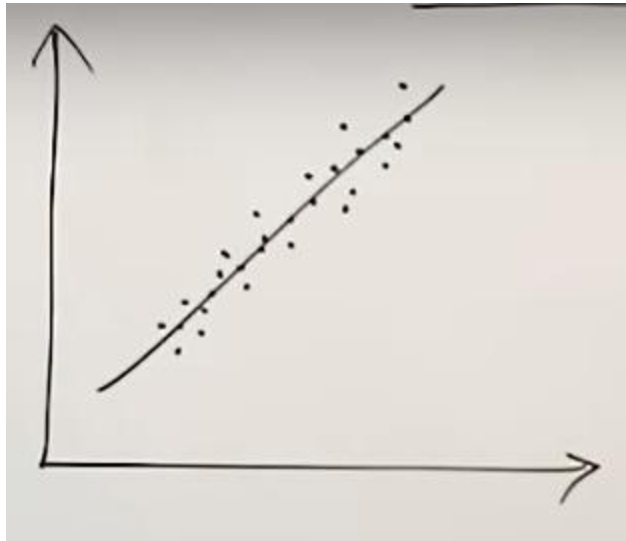
Some common feature extraction techniques are:

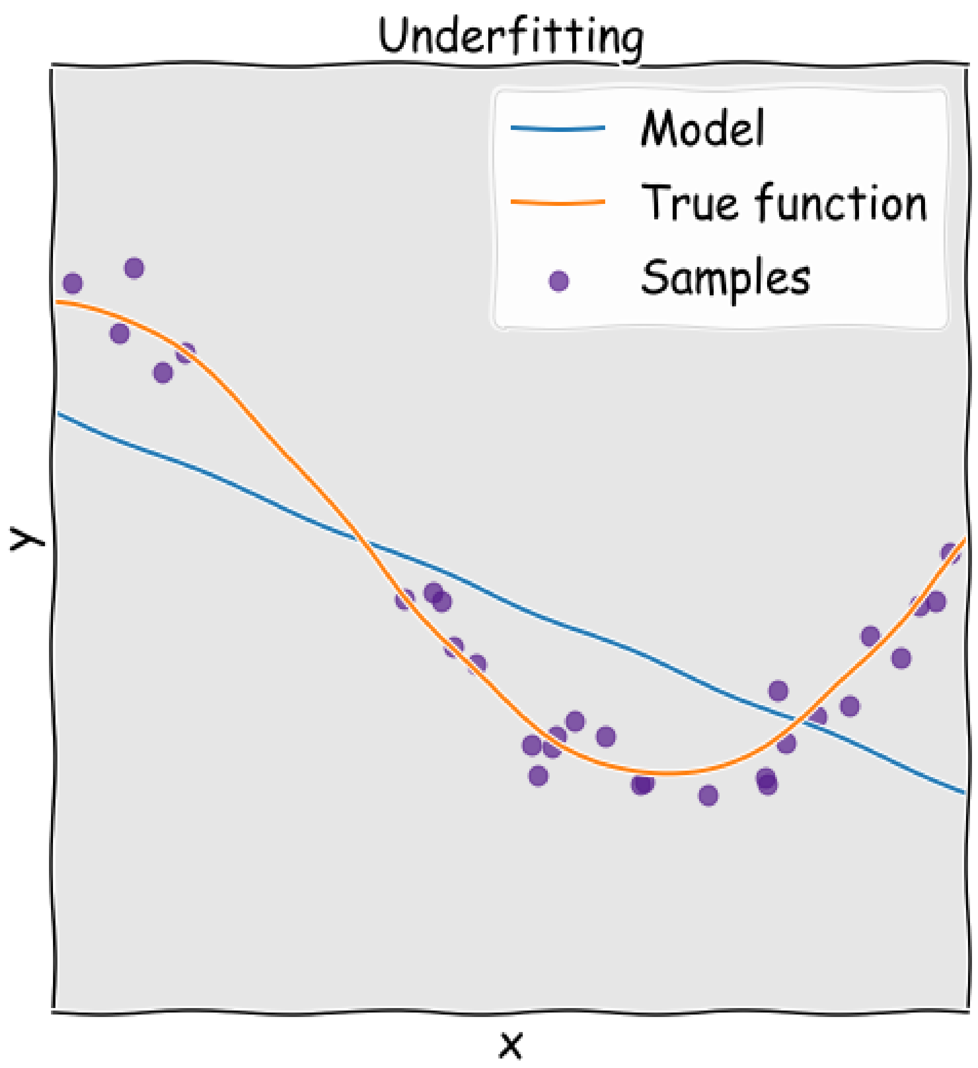
1. Principal Component Analysis (PCA)
2. Linear Discriminant Analysis (LDA)
3. Kernel PCA
4. Quadratic Discriminant Analysis (QDA)etc.

ML Model design

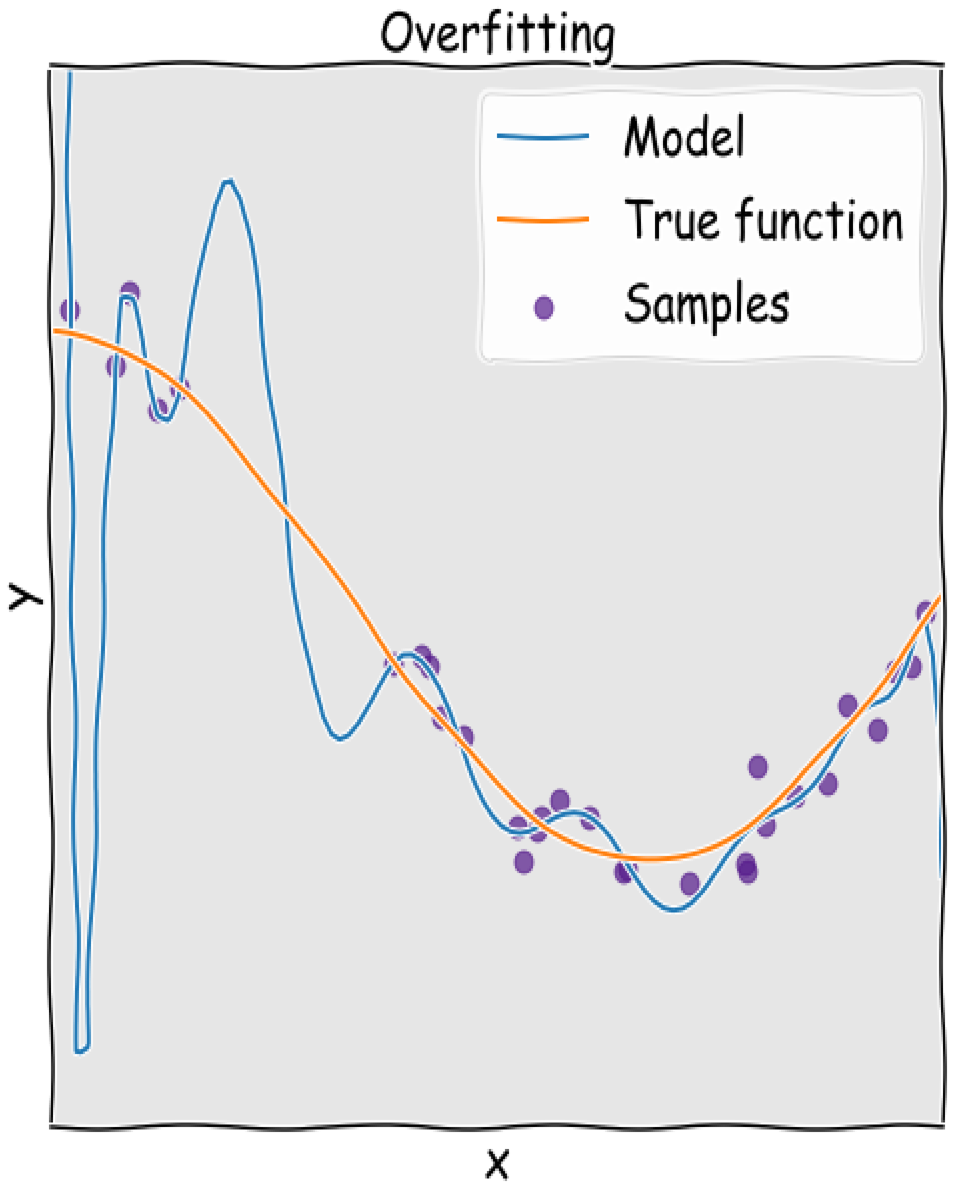
- Consider the line passing through the samples in the diagram.
- It (line) is the model/function/hypothesis generated after the training phase.

The line is trying to reach all the samples as close as possible.

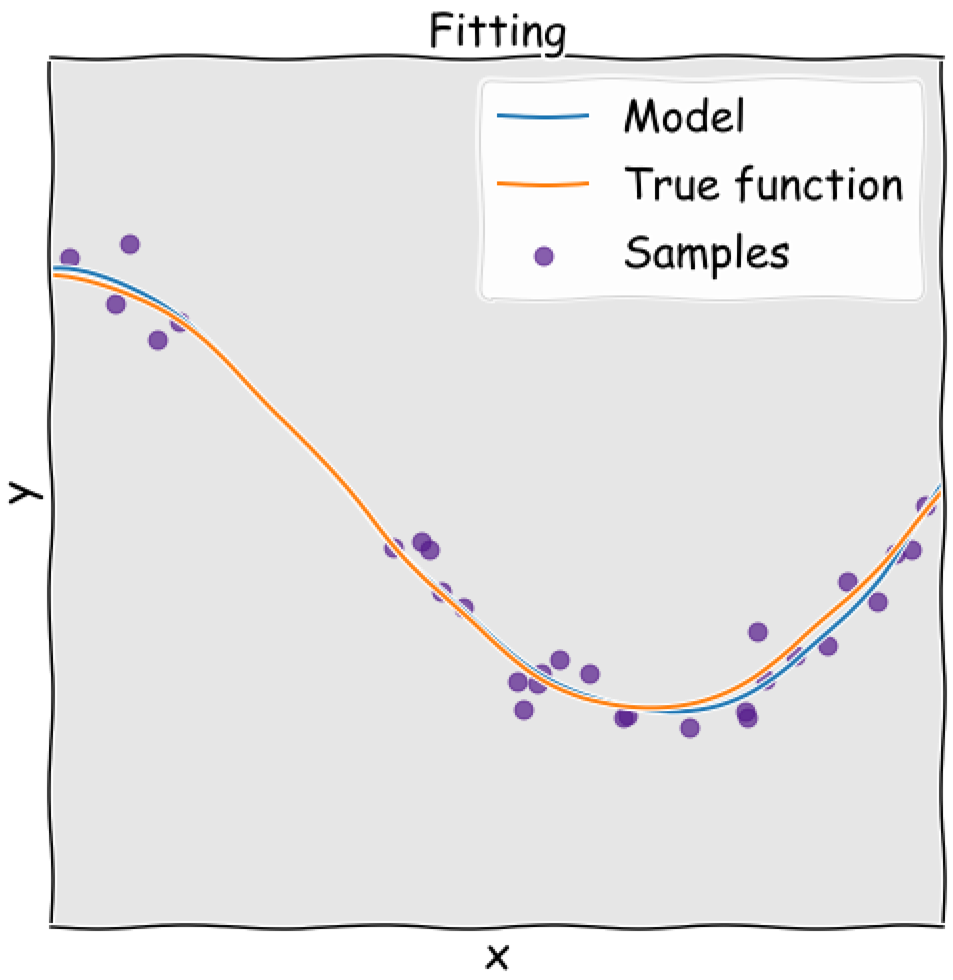




- If we have an underfitted model, this means that we do not have enough parameters to capture the trends in the underlying system.
- In general, in underfitting, model fails during testing as well as training.



- In this a complex model is built using too many features.
- During training phase, model works well. But it fails during testing.



- Under/Overfitting can be solved in different ways.
- One of the solution for overfitting is dimensionality reduction.
- Diagram shows that model neither suffers from under or overfitting.

Example to show requirement of Dimensionality reduction

- In this example important features to decide the price are town, area and plot size. Features like number of bathroom and trees nearby may not be significant, hence can be dropped.

town	area	bathroom	plot	trees nearby	price
monroe	2600	2	8500	2	550000
monroe	3000	3	9200	2	565000
monroe	3200	3	8750	2	610000
monroe	3600	4	10200	2	680000
monroe	4000	4	15000	2	725000
west windsor	2600	2	7000	2	585000
west windsor	2800	3	9000	2	615000
west windsor	3300	4	10000	1	650000
west windsor	3600	4	10500	1	710000

PCA

- PCA is a method of Dimensionality Reduction.
- PCA is a process of identifying Principal Components of the samples.
- It tries to address the problem of overfitting.

Principal Component Analysis

- Principal components is a form of multivariate statistical analysis and is one method of studying the correlation or covariance structure in a set of measurements on m variables for n observations.

- Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality 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.

- **Reducing the number of variables of a data set naturally comes at the expense of accuracy**, but the trick in dimensionality reduction is to **trade a little accuracy for simplicity**.

- **Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.**

- So to sum up, the idea of PCA is simple — reduce the number of variables of a data set, while preserving as much information as possible.

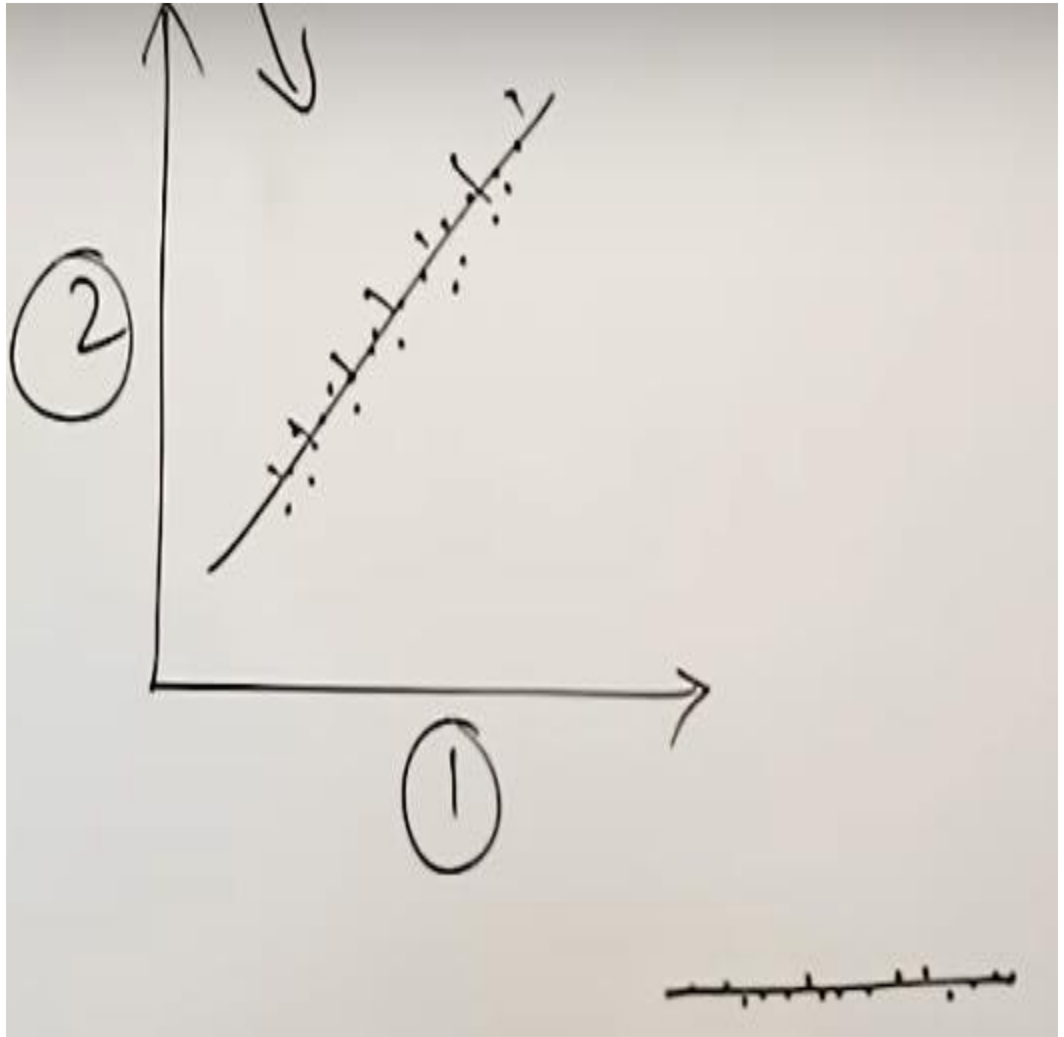
Example for PCA (from SK learn (SciKit Learn) library)

	pixel_0	pixel_1	pixel_2	pixel_3	pixel_4	pixel_5	pixel_6	...	pixel_64	Digit
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	...	0.0	5
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	...	0.0	0
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	...	5.0	2
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	...	9.0	9
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	...	0.0	1

PCA (n_components=6)

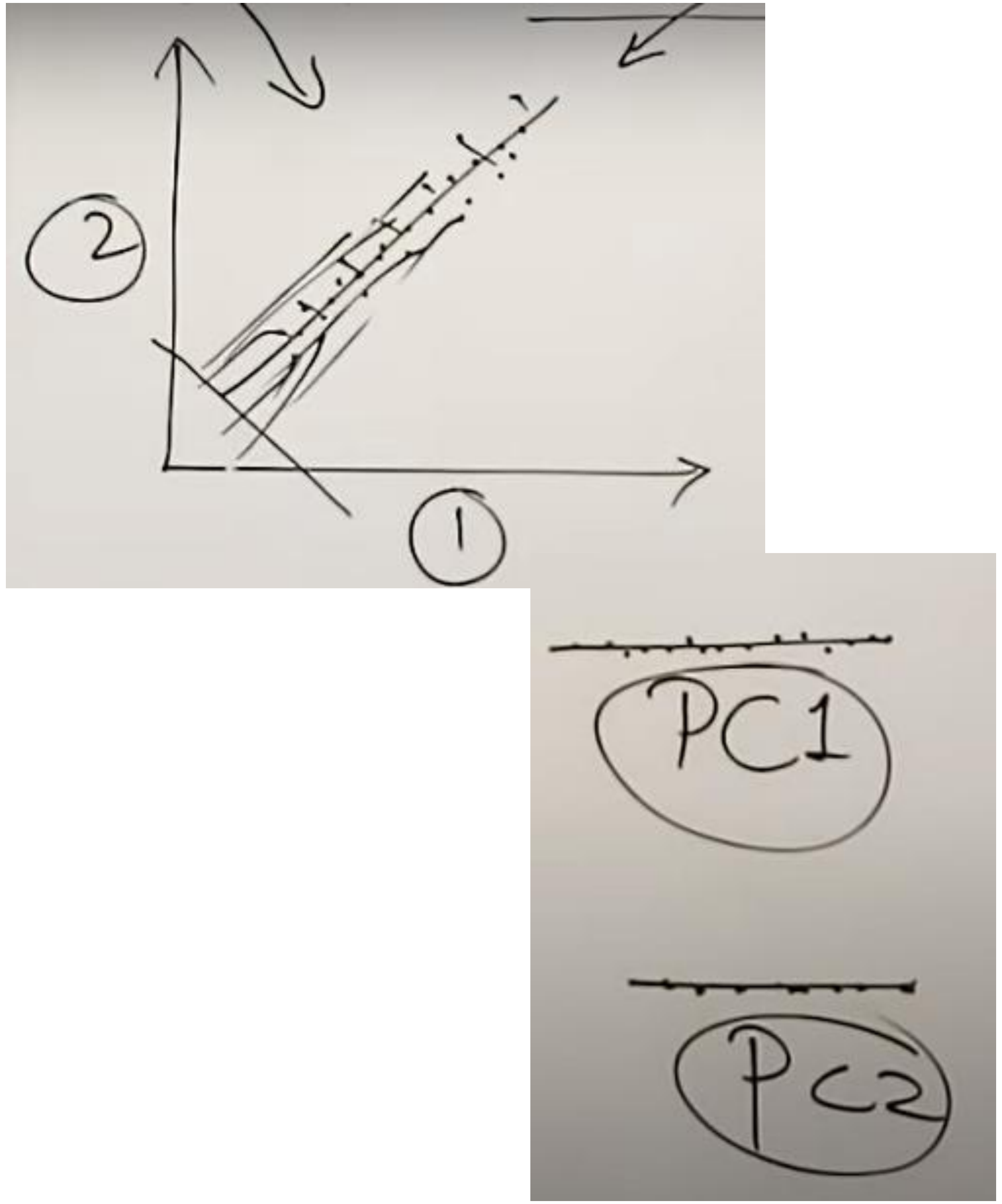
	PC 1	PC 2	PC 4	PC 5	PC 6	Digit
0	5.2	15.0	1.23	6.01	15.0	5
1	7.2	13.0	0.0	9.5	12.0	0
2	1.23	4.0	9.0	9.0	15.0	2
3	9.5	12.0	7.2	7.2	0.0	9
4	8.1	0.0	13.0	15.0	5.2	1

What does PCA do?



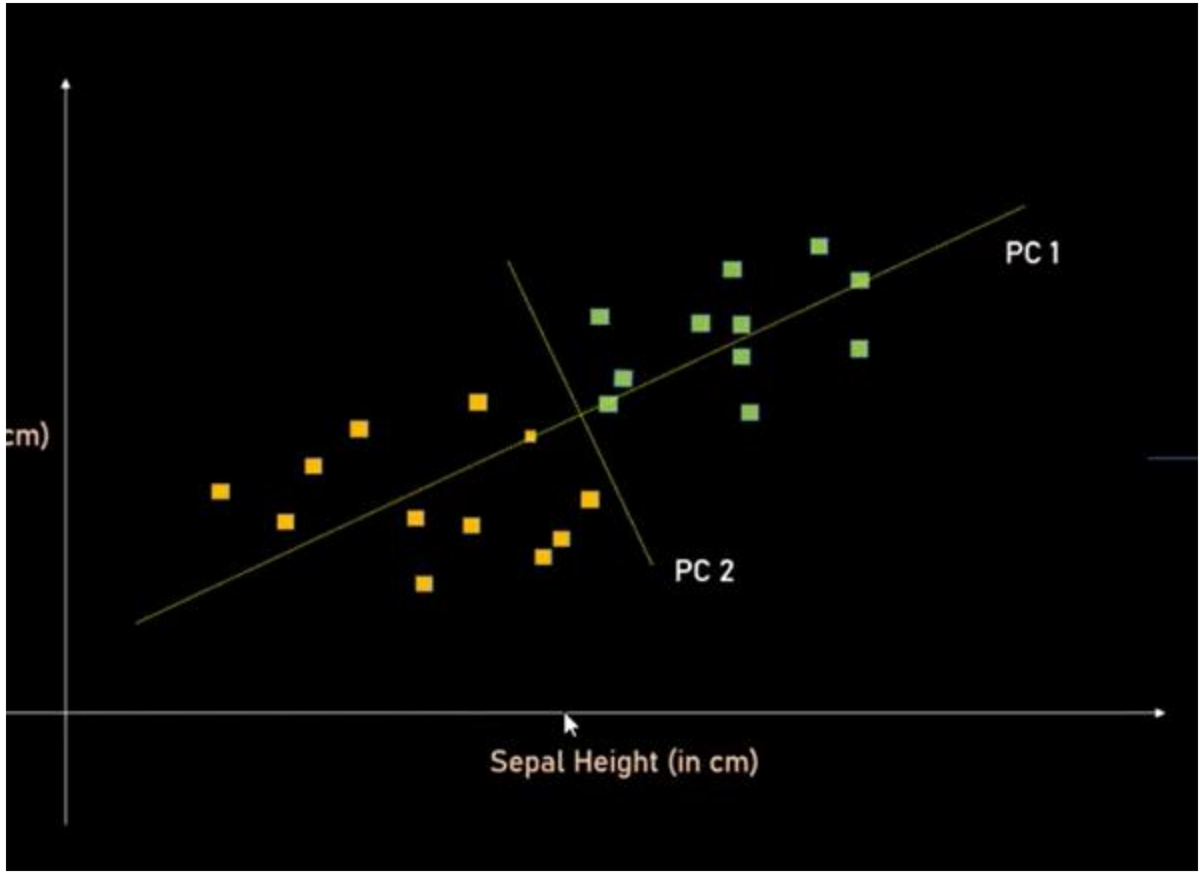
- To address overfitting, reduce the dimension, without loosing the information.
- In this example two dimension is reduced to single dimension.
- But in general, their can be multiple dimensions… and will be reduced.
- When the data is viewed from one angle, it will be reduced to single dimension and the same is shown at the bottom right corner, and this will be Principal Component 1.

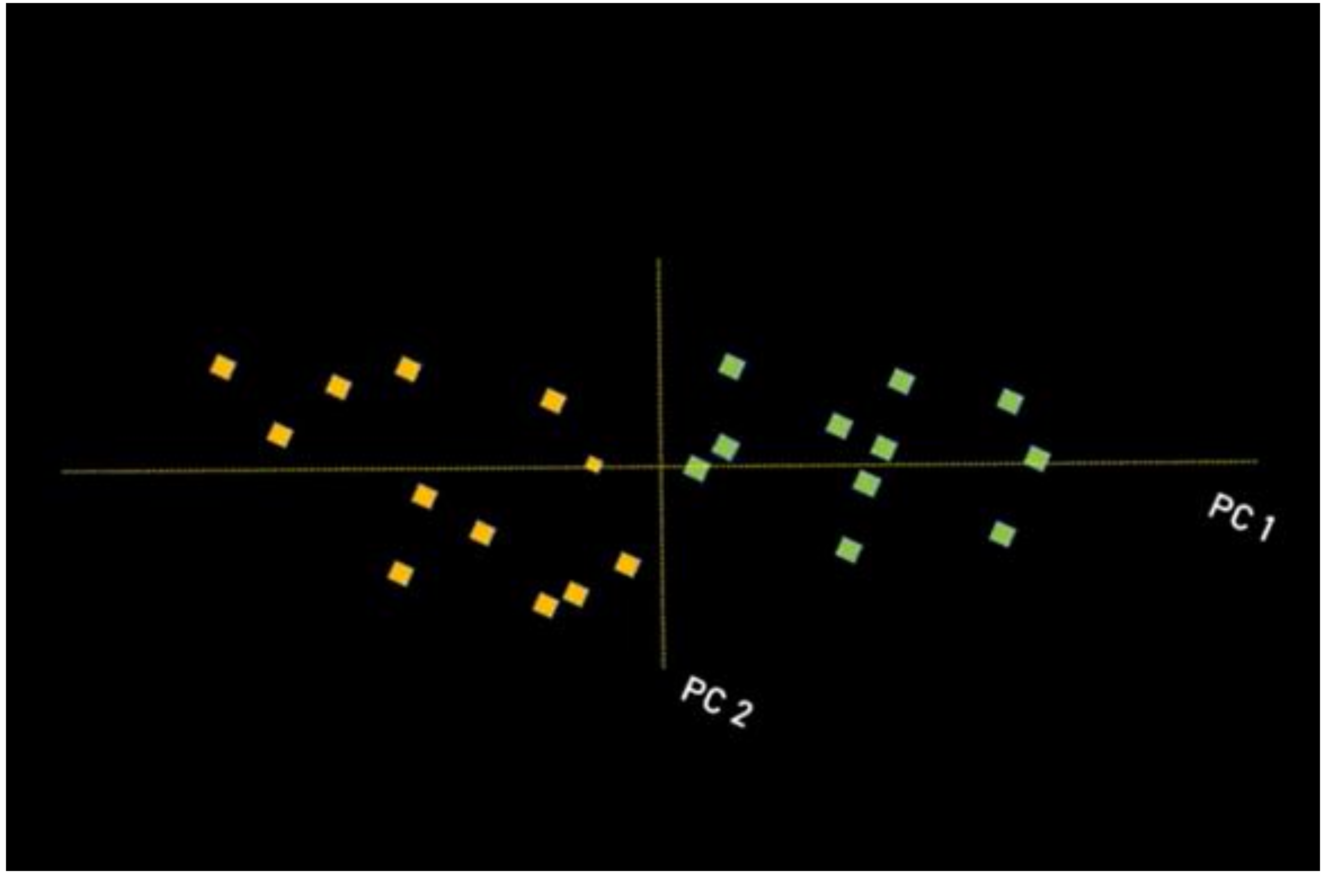
Similarly compute PC2



- Figure shows the representation of PC1 and PC2.
- Like this we have several principal components…
- Say PC1,PC2, PC3… and so on..
- In that PC1 will be of top priority.
- Each Principal Components are independent and are orthogonal. It means one PC does not depends on another…all of them are independent.

Another Example

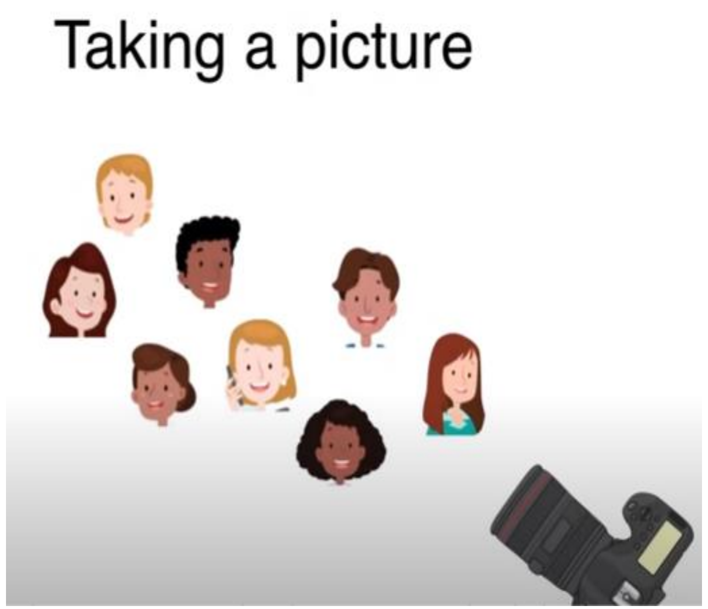


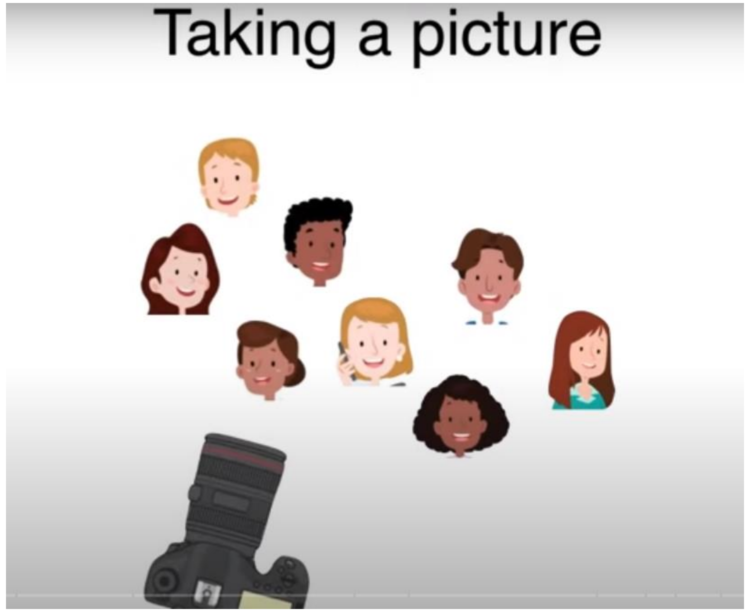


Example to illustrate the PC

Taking a picture

Multiple angles in which picture can be captured



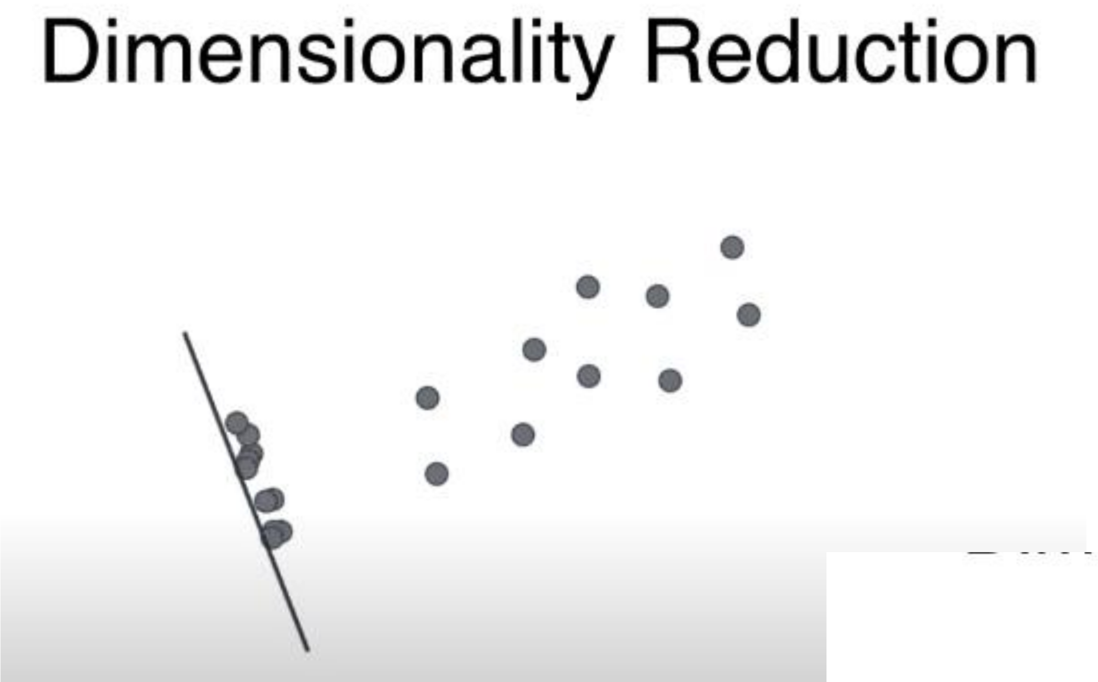


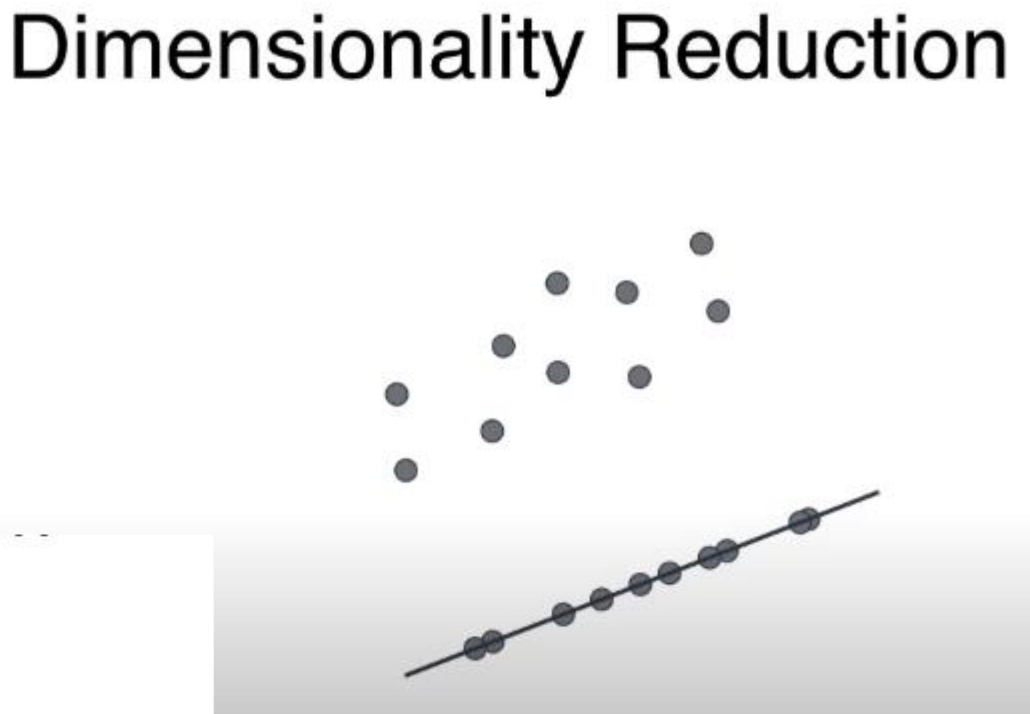
- In previous slide, the last picture gives the right angle to take the picture.
- It means, you have to identify a better angle to collect the data without loosing much information.
- The angle shown in the last picture will capture all the faces, without much overlapping and without loosing information.

In this example the second one is the best angle to project

https://www.youtube.com/watch?v=g-Hb26agBFg (reference video)

https://www.youtube.com/watch?v=MLaJbA82nzk

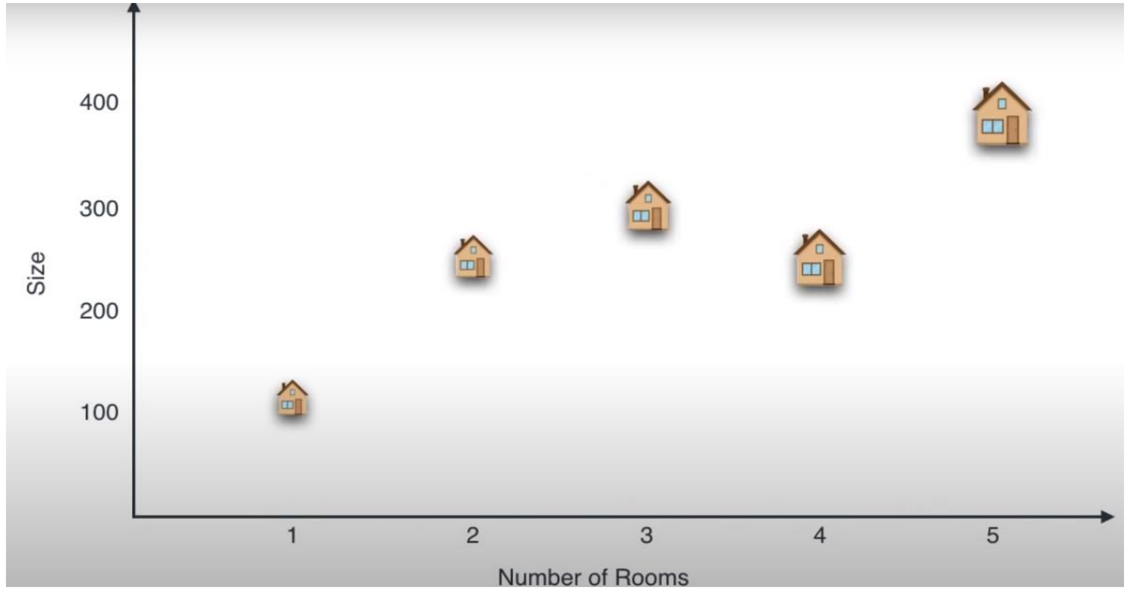




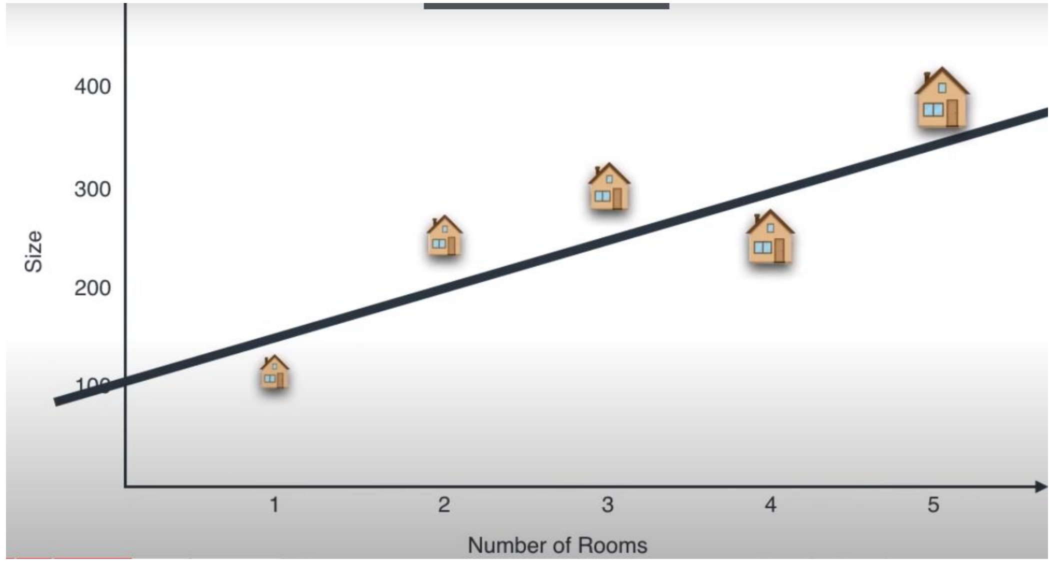
Housing Example: More rooms..more the size

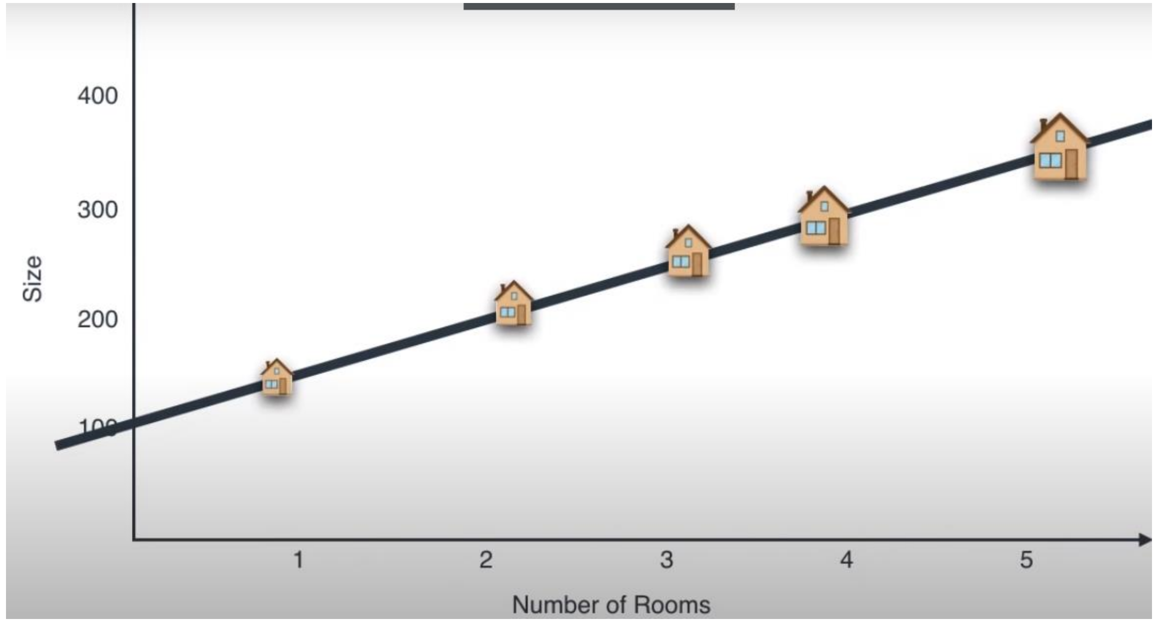
Housing Data

Size
Number of rooms $\longrightarrow$ Size feature



Two dimension is reduced to single dimension





- PCA is a method of dimensionality reduction.
- Example shows how to convert a two dimension to one dimension.

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

How to compute PCA?

- Consider the Samples given in the table (10 Samples).
- Compute the mean of X and mean of Y independently. Similar computation has to be done for each features. (In this example only two features).
- Mean of X = 1.81 and Mean of Y = 1.91

Next Step is to compute Co-Variance Matrix.

- Covariance between (x, y) is computed as given below:

$$Cov(x,y) = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

- The following covariance Matrix to be computed is:

$$C = \begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix}$$

Covariance between (x and x)

X	Y	(X- Mean(X))		(x-mean(x) * (x-Mean(x)
2.5	2.4	0.69		0.476
0.5	0.7	-1.31		1.716
2.2	2.9	0.39		0.152
1.9	2.2	0.09		0.008
3.1	3	1.29		1.664
2.3	2.7	0.49		0.24
2	1.6	0.19		0.036
1	1.1	-0.81		0.656
1.5	1.6	-0.31		0.096
1.1	0.9	-0.71		0.504
			Total=	5.549
			Total/9	0.617

- Similarly compute co variance between (x,y),(y,x) and (y,y).
- Computed Co-Variance matrix is given in next slide

Final co-variance matrix

$$C = \begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix}$$

$$= \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix}$$

Next Step is to Compute Eigen Values using the Co-variance matrix

If A is the given matrix (in this case co-variance matrix)

We can calculate eigenvalues from the following equation:

$$|A - \lambda I| = 0$$

Where A is the given matrix

λ is the eigen value

I is the identity Matrix

$$|A - \lambda I| = 0$$

$$\Rightarrow \left| \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{bmatrix} 0.6165-\lambda & 0.6154 \\ 0.6154 & 0.7165-\lambda \end{bmatrix} \right| = 0$$

Determinant computation and finally Eigen values

$$\left[(0.6165-\lambda)*(0.7165-\lambda)-(0.6154)*(0.6154)\right]=0$$

$$\Rightarrow (0.6165*0.7165)-(0.6165\lambda)-(0.7165\lambda)+\lambda^2-(0.6154)*(0.6154)=0$$

$$\Rightarrow \boxed{\lambda^2-1.333\lambda+0.0630=0}$$

$$\boxed{\begin{aligned} a &= 1 \\ b &= -1.33 \\ c &= 0.0630 \end{aligned}}$$

Quadratic Formula Calculator

$ax^2+bx+c=0$

a = 1

b = -1.33

c = 0.063

Clear | Calculate

Answer:

$x = 1.28081$

$x = 0.0491875$

$$\boxed{\begin{aligned} \lambda_1 &= 1.2840 \\ \lambda_2 &= 0.490 \end{aligned}}$$

- Compute Eigen vector for each of the eigen value.

$$\boxed{CV = \lambda V}$$

- Consider the first eigen value **λ1** = 1.284
- C is the covariance matrix
- V is the eigen vector to be computed.

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = 1.2840 \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = 1.9840 \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$0.6165x_1 + 0.6154y_1 = 1.9840x_1$$
$$0.6154x_1 + 0.7169y_1 = 1.9840y_1$$

$$0.6154y_1 = 1.9840x_1 - 0.6165x_1$$
$$0.6154y_1 = 0.6675x_1$$

$$0.6675x_1 = 0.6154y_1$$

$$x_1 = \frac{0.6154}{0.6675} y_1$$

$$\boxed{x_1 = 0.9219\, y_1}$$

$$y_1 = 1$$
$$x_1 = 0.9219$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0.9219 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.9219/p \\ 1/p \end{bmatrix} =$$

$$p = \sqrt{(0.9219)^2 + 1^2}$$

$$\boxed{p = 1.360}$$

$$= \begin{bmatrix} 0.6952 \\ 0.7541 \end{bmatrix}$$

Now convert the two dimension data to single dimension

Sample 1:- $x = 2.5$

$y = 2.4$

$\bar{x} = 1.81$

$\bar{y} = 1.91$

$$= e_1^T * \begin{bmatrix} x - \text{mean}(x) \\ y - \text{mean}(y) \end{bmatrix}$$ => new transformed data

Final step

$$\Rightarrow \begin{pmatrix} 0.6952 & 0.7541 \end{pmatrix} \begin{bmatrix} 0.69 \\ 0.49 \end{bmatrix}$$

$$\Rightarrow 0.4796 + 0.3695$$

$$\Rightarrow 0.8491$$

- Compute Eigen vector for the second eigen value.

$$\boxed{CV = \lambda V}$$

- Consider the first eigen value **λ2** = 0.0490
- C is the covariance matrix
- V is the eigen vector to be computed.

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = 0.0490 \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$$

- Using this we can have two linear equation:

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = 0.0490 \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$$

$$0.6165 X_1 + 0.6154 Y_1 = 0.0490 X_1$$
$$0.6154 X_1 + 0.7165 Y_1 = 0.0490 Y_1$$

$$0.5674 X_1 = -0.6154 Y_1$$
$$0.6154 X_1 = -0.6674 Y_1$$

- Use any one of the following equation… final result remains same.

$$0.5674\, X_1 = -0.6154\, Y_1$$

$$0.6154\, X_1 = -0.6674\, Y_1$$

- 0.5674 x1 = -0.6154 y1
- Divide both side by 0.5674.
- You will get :   x1 = -1.0845 y1

- **$x1 = -1.0845\ y1$**
- **If y1=1, then x1 will be -1.0845**
- So in that case (x1, y1) will be (-1.0845,1). This will be the initial eigen vector. Needs normalization to get the final value.
- To normalize, take square-root of sum of square of each eigen vector values, and consider this as ‘x’
- Finally divide each eigen vector values by ‘x’ to get the final eigen vector.

eigen vectors are generated for the eigen value : 0.490

$X_1 = -1.0845\, Y_1$

$$\begin{bmatrix} -1.0845 \\ 1 \end{bmatrix} = 1.17614 + 1 = \sqrt{2.17614} = 1.47517$$

$$\Rightarrow \begin{bmatrix} -0.7351 \\ 0.6778 \end{bmatrix}$$

$X_2 = 0.92194\, Y_2$

$$\begin{bmatrix} 0.92194 \\ 1 \end{bmatrix} = 0.8499 + 1 = \sqrt{1.8499} = 1.3601$$

$$\Rightarrow \begin{bmatrix} 0.6778 \\ 0.7351 \end{bmatrix}$$

Describe the algorithm with an example:

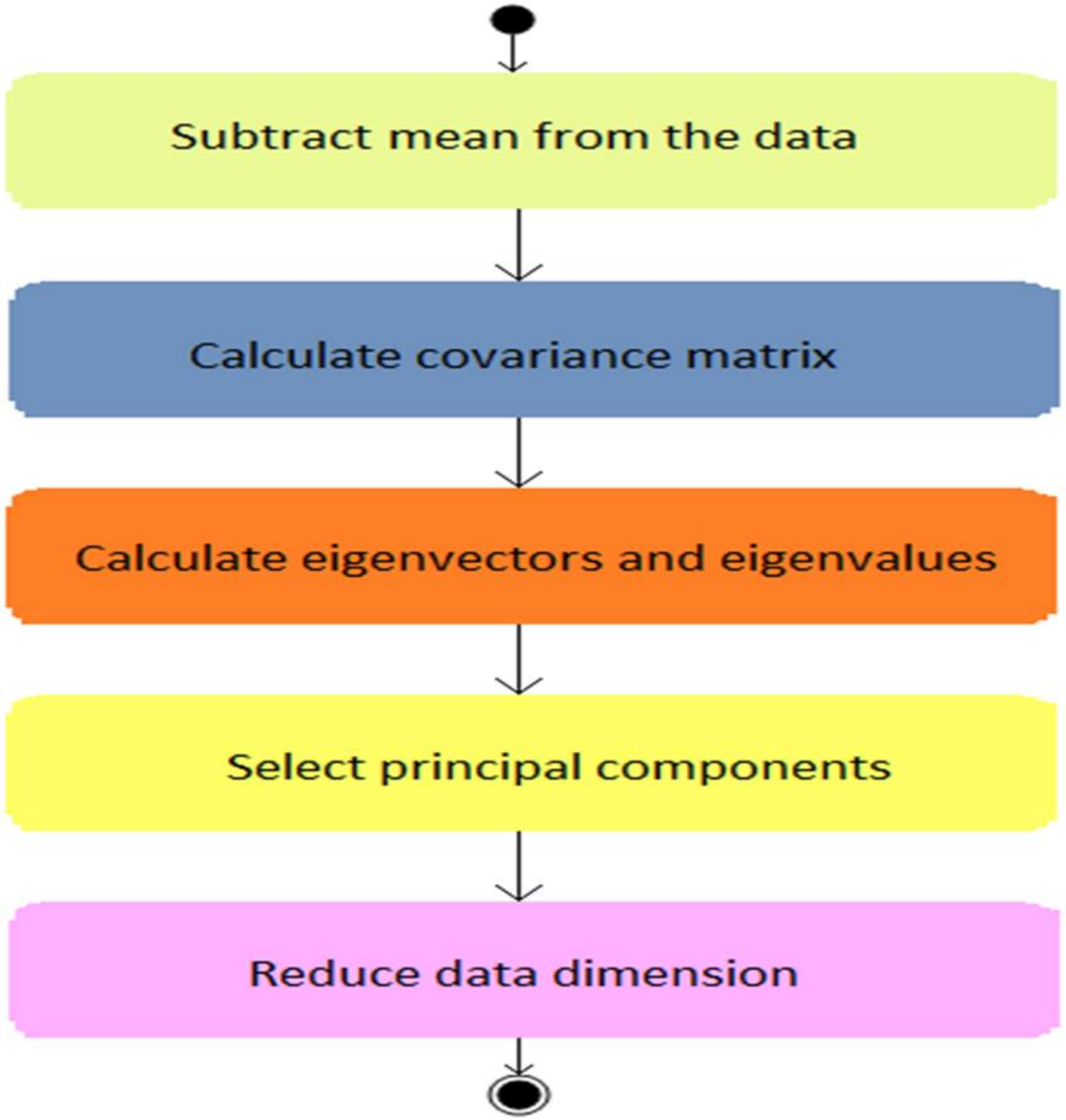
- Consider a 2-D dataset
- Cl =X1 =(x1,x2) ={(4,1),(2,4),(2,3),(3,6), (4,4)}
- C2=X2=(x1,x2) = {(9,10),(6,8),(9,5),(8,7),(10,8)}

PCA

Theory – Algorithms – steps explained

Steps/ Functions to perform PCA

- Subtract mean.
- Calculate the covariance matrix.
- Calculate eigenvectors and eigenvalues.
- Select principal components.
- Reduce the data dimension.



Eigenvectors and eigenvalues are computed from the covariance matrix in order to determine the ***principal components*** of the data.

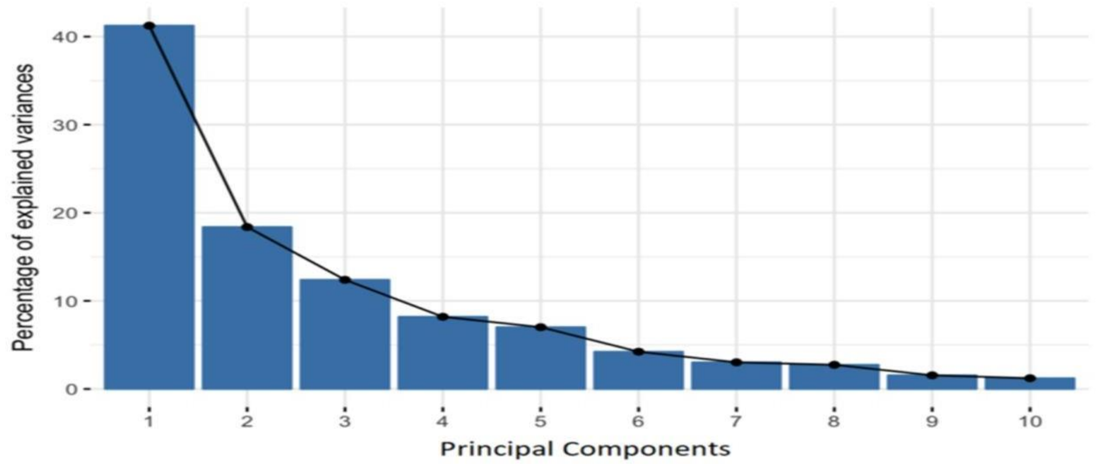
Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables.

These combinations are done in such a way that the new variables (i.e., principal components) are **uncorrelated** and most of the information within the initial variables is squeezed or compressed into the first components.

So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component.

Then next maximum of the remaining information in the second and so on, until having something like shown in the scree plot below.

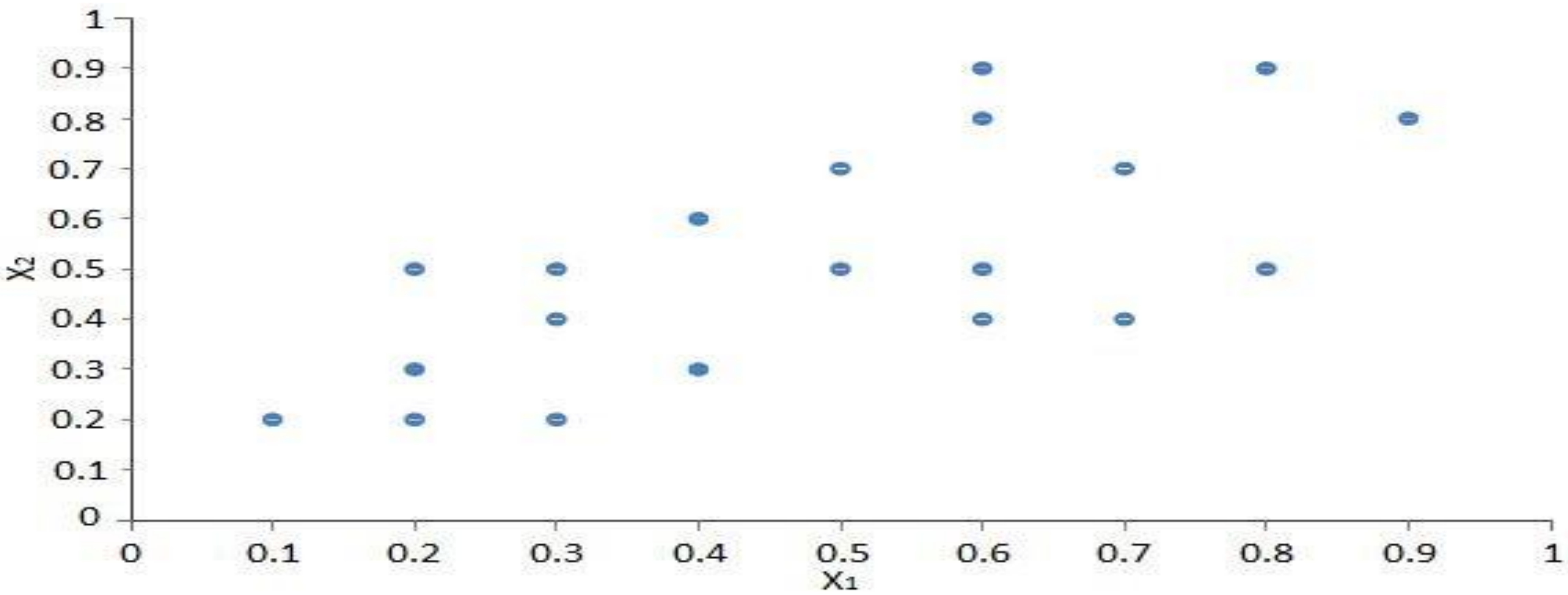
- There are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the **largest possible variance** in the data set.



- Organizing information in principal components this way, will allow us to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as your new variables.
- An important thing to realize here is that, the **principal components are less interpretable and don't have any real meaning** since they are constructed as linear combinations of the initial variables.

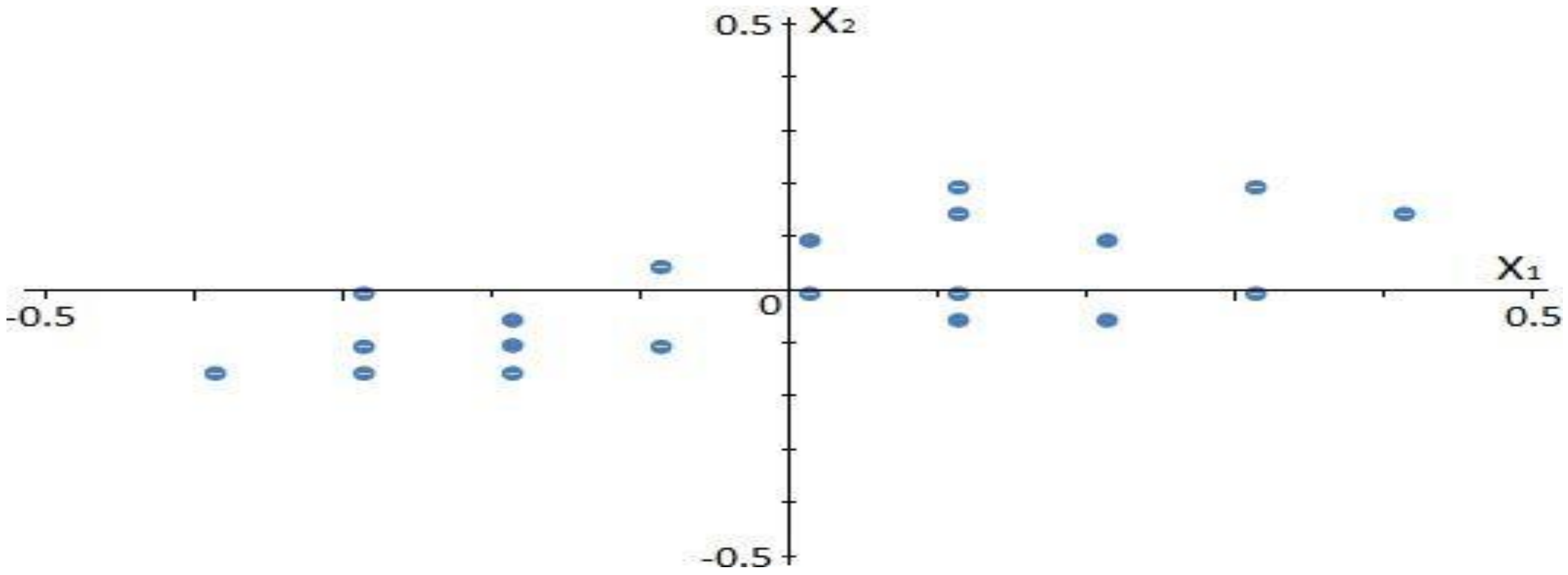
Instance	x1	x2
1	0.3	0.5
2	0.4	0.3
3	0.7	0.4
4	0.5	0.7
5	0.3	0.2
6	0.9	0.8
7	0.1	0.2
8	0.2	0.5
9	0.6	0.9
10	0.2	0.2

Instance	x1	x2
11	0.6	0.8
12	0.4	0.6
13	0.3	0.4
14	0.6	0.5
15	0.8	0.5
16	0.8	0.9
17	0.2	0.3
18	0.7	0.7
19	0.5	0.5
20	0.6	0.4



X₂
X₁
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

After subtracting the mean



The covariance of two random variables measures the degree of variation from their means for each other.

The sign of the covariance provides us with information about the relation between them:

If the covariance is positive, then the two variables increase and decrease together.

If the covariance is negative, then when one variable increases, the other decreases, and vice versa.

These values determine the linear dependencies between the variables, which will be used to reduce the data set's dimension.

x1 x2 x1

0.33 0.25

x2 0.25 0.41

Let's suppose that our data set is 2-dimensional with 2 variables ***x,y*** and that the eigenvectors and eigenvalues of the covariance matrix are as follows:

$$v1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix} \qquad \lambda_1 = 1.284028$$

$$v2 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix} \qquad \lambda_2 = 0.04908323$$

If we rank the eigenvalues in descending order, we get $\lambda 1 > \lambda 2$, which means that the eigenvector that corresponds to the first principal component (PC1) is v1 and the one that corresponds to the second component (PC2) is v2.

After having the principal components, to compute the percentage of variance (information) accounted for by each component, we divide the eigenvalue of each component by the sum of eigenvalues. If we apply this on the example above, we find that PC1 and PC2 carry respectively 96% and 4% of the variance of the data.

- As we saw in the previous step, computing the eigenvectors and ordering them by their eigenvalues in descending order, allow us to find the principal components in order of significance. In this step, what we do is, to choose whether to keep all these components or discard those of lesser significance (of low eigenvalues), and form with the remaining ones a matrix of vectors that we call *Feature vector*.

- So, the feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep. This makes it the first step towards dimensionality reduction, because if we choose to keep only ***p*** eigenvectors (components) out of ***n***, the final data set will have only ***p*** dimensions.

Continuing with the example from the previous step, we can either form a feature vector with both of the eigenvectors *v*1 and *v*2:

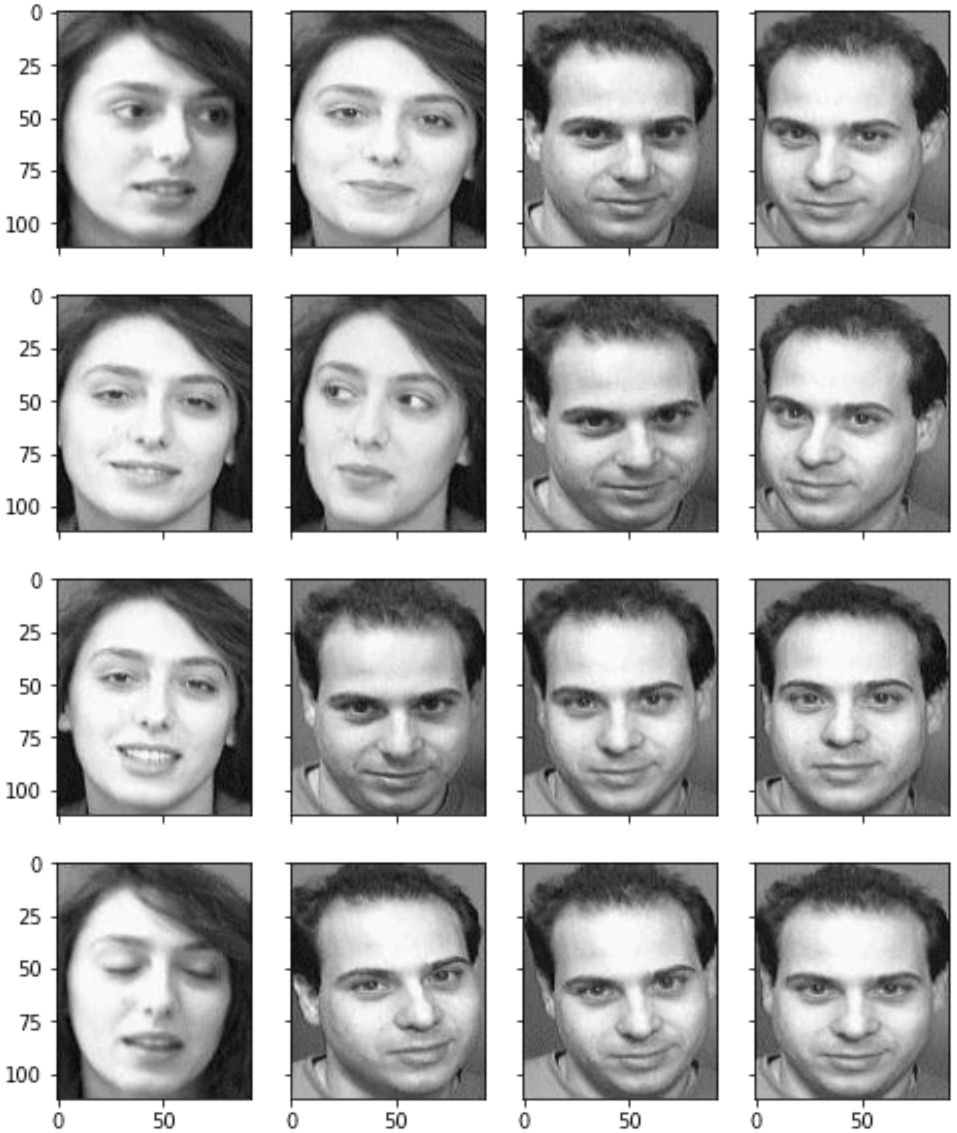
$$\begin{bmatrix} 0.6778736 & -0.7351785 \\ 0.7351785 & 0.6778736 \end{bmatrix}$$

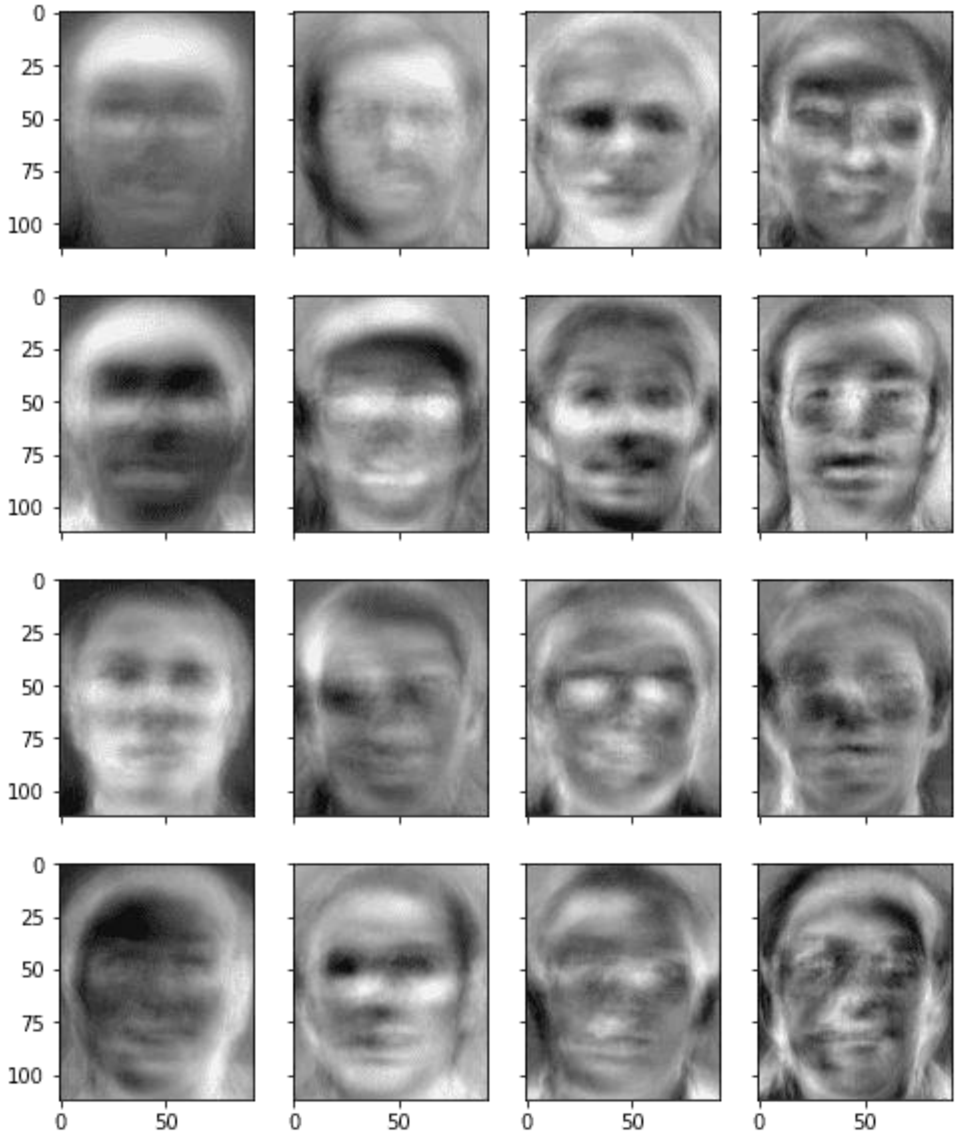
Or discard the eigenvector *v*2, which is the one of lesser significance, and form a feature vector with *v*1 only:

$$\begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix}$$

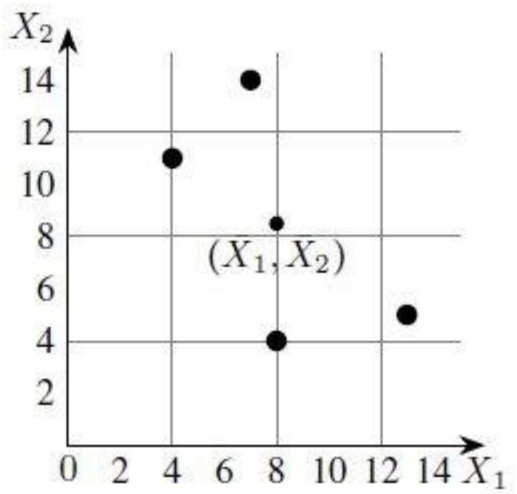
Discarding the eigenvector v2 will reduce dimensionality by 1, and will consequently cause a loss of information in the final data set. But given that v2 was carrying only 4% of the information, the loss will be therefore not important and we will still have 96% of the information that is carried by v1.

- Principal Components in PCA
- As described above, the transformed new features or the output of PCA are the Principal Components. The number of these PCs are either equal to or less than the original features present in the dataset. Some properties of these principal components are given below:
- The principal component must be the linear combination of the original features.
- These components are orthogonal, i.e., the correlation between a pair of variables is zero.
- The importance of each component decreases when going to 1 to n, it means the 1 PC has the most importance, and n PC will have the least importance.


0
25
50
75
100
0
50


0
25
50
75
100
0
50

PCA-Example



Feature	Sample 1	Sample 2	Sample 3	Sample 4
X_1	4	8	13	7
X_2	11	4	5	14

Calculate the mean of X1 and X2 as shown below.

$$\bar{X}_1 = \tfrac{1}{4}(4+8+13+7) = 8,$$
$$\bar{X}_2 = \tfrac{1}{4}(11+4+5+14) = 8.5.$$

The covariances are calculated as follows:

$$\begin{aligned}
\text{Cov}(X_1, X_2) &= \frac{1}{N-1}\sum_{k=1}^{N}(X_{1k} - \bar{X}_1)^2 \\
&= \tfrac{1}{3}\left((4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\right) \\
&= 14
\end{aligned}$$

$$\begin{aligned}
\text{Cov}(X_1, X_2) &= \frac{1}{N-1}\sum_{k=1}^{N}(X_{1k} - \bar{X}_1)(X_{2k} - \bar{X}_2) \\
&= \tfrac{1}{3}((4-8)(11-8.5) + (8-8)(4-8.5) \\
&\quad + (13-8)(5-8.5) + (7-8)(14-8.5) \\
&= -11
\end{aligned}$$

$$\begin{aligned}
\text{Cov}(X_2, X_1) &= \text{Cov}(X_1, X_2) \\
&= -11
\end{aligned}$$

$$\begin{aligned}
\text{Cov}(X_2, X_2) &= \frac{1}{N-1}\sum_{k=1}^{N}(X_{2k} - \bar{X}_2)^2 \\
&= \tfrac{1}{3}\left((11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2\right) \\
&= 23
\end{aligned}$$

The covariance matrix is,

$$S = \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) \end{bmatrix}$$

$$= \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Eigenvalues of the covariance matrix

The characteristic equation of the covariance matrix is,

$$\begin{aligned} 0 &= \det(S - \lambda I) \\ &= \begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix} \\ &= (14-\lambda)(23-\lambda) - (-11) \times (-11) \\ &= \lambda^2 - 37\lambda + 201 \end{aligned}$$

$$\begin{aligned} \lambda &= \tfrac{1}{2}(37 \pm \sqrt{565}) \\ &= 30.3849,\ 6.6151 \\ &= \lambda_1,\ \lambda_2 \quad \text{(say)} \end{aligned}$$

Linear Discriminant Analysis (LDA)

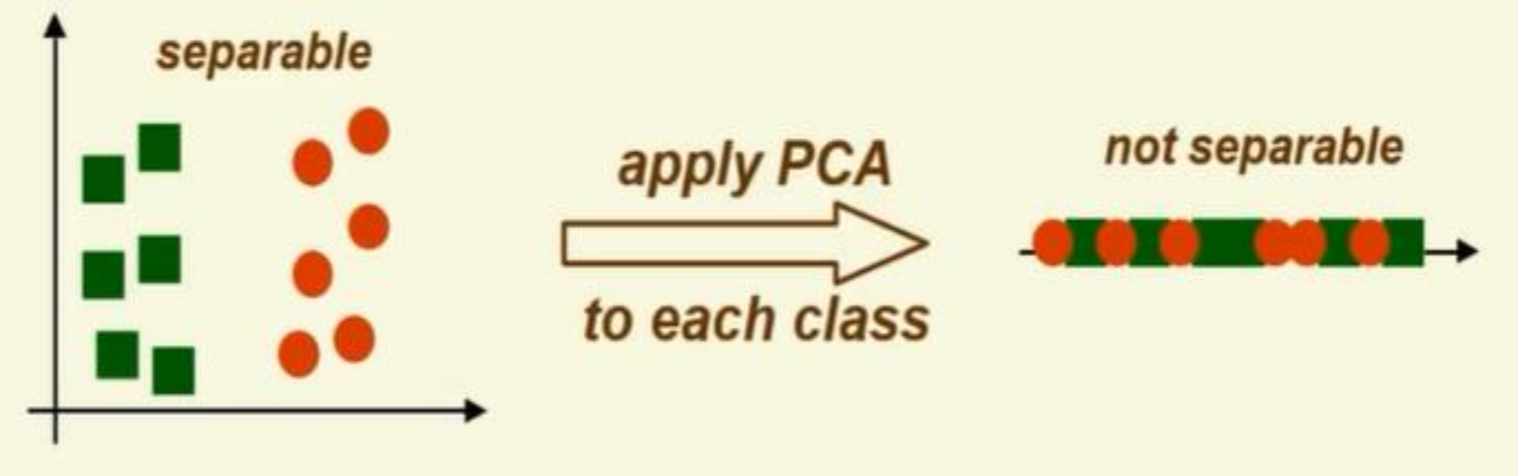
Linear Discriminant Analysis (LDA) is one of the commonly used dimensionality reduction techniques in machine learning to solve more than two-class classification problems. LDA is also a dimensionality reduction technique. It is used as a pre-processing step in Machine Learning and applications of pattern classification.

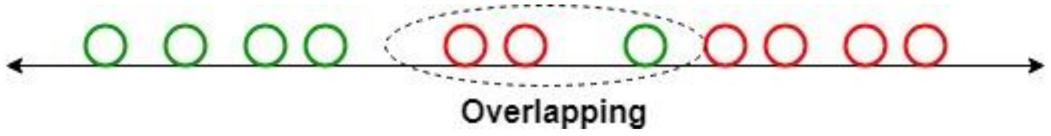
The goal of LDA is to project the features in higher dimensional space onto a lower dimensional space in order to avoid the curse of dimensionality and also reduce resources and dimensional costs.

The original technique was developed in the year 1936 by Ronald A. Fisher and was named Linear Discriminant or Fisher's Discriminant Analysis. The original Linear Discriminant was described as a two-class technique. The multi-class version was later generalized by C.R Rao as Multiple Discriminant Analysis. They are all simply referred to as the Linear Discriminant Analysis.

LDA Algorithm

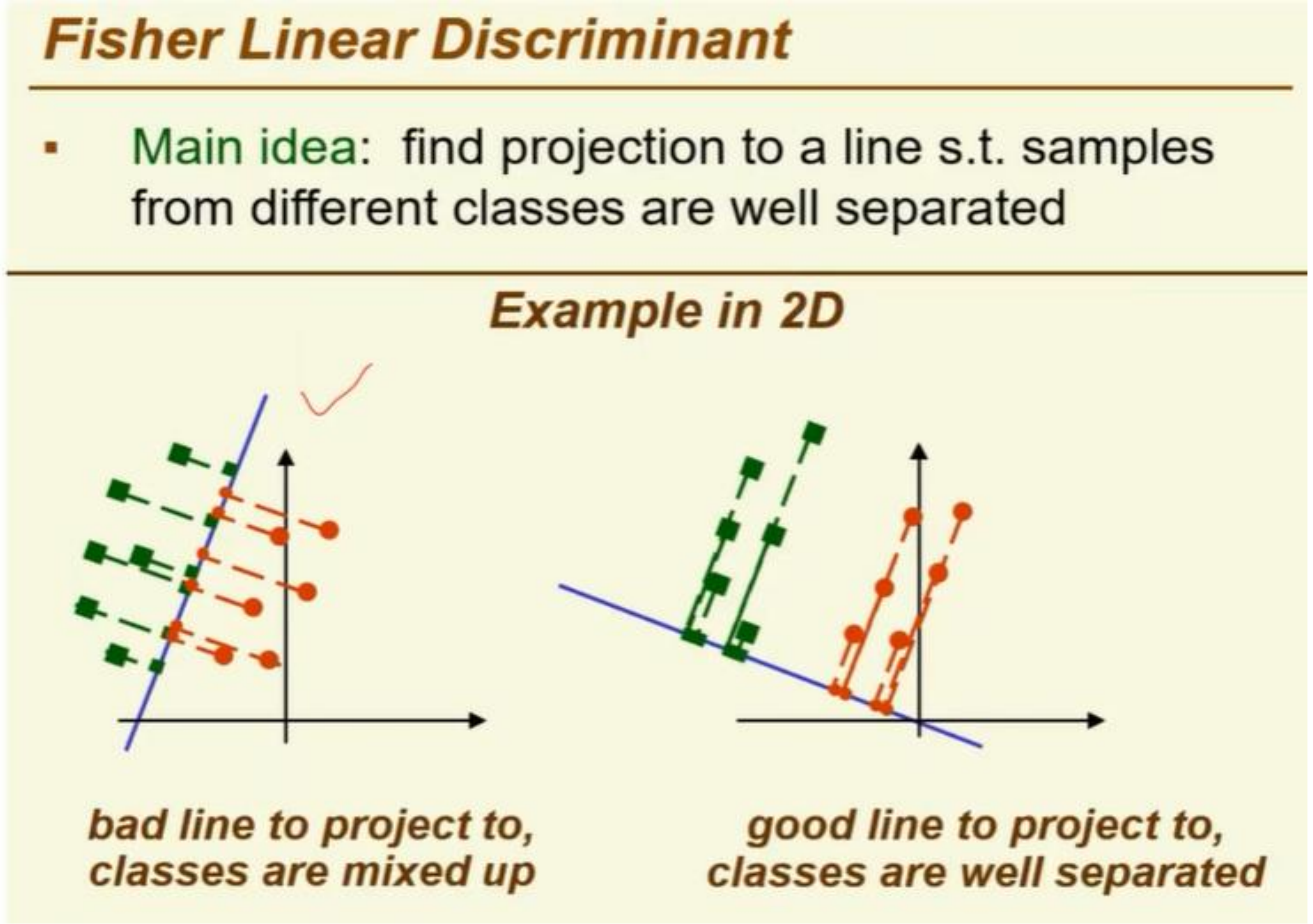
- PCA is good for dimensionality reduction.
- However Figure shows how PCA fails to classify. (because it will try to project this points which maximizes variance and minimizes the error)





- Fisher Linear Discriminant Project to a line which reduces the dimension and also maintains the class discriminating information.

Projection of the samples in the second picture is the best:



LDA vs. PCA : Linear discriminant analysis is very similar to PCA both look for linear combinations of the features which best explain the data.

The main difference is that the **Linear discriminant analysis** is a **supervised** dimensionality reduction technique that also achieves classification of the data simultaneously.

LDA focuses on finding a feature subspace that **maximizes the separability** between the groups.

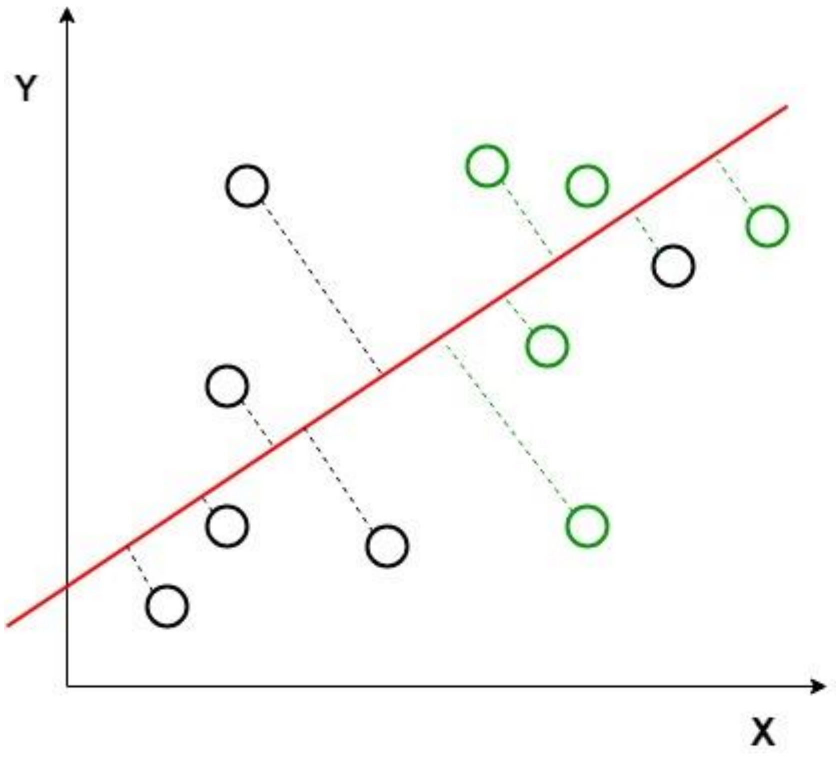
While **Principal component analysis** is an **unsupervised** Dimensionality reduction technique, it ignores the class label.

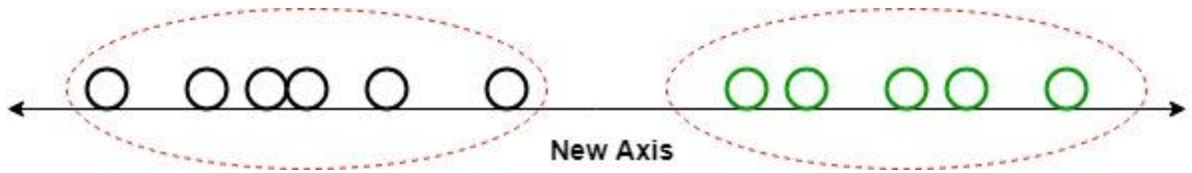
PCA focuses on capturing the direction of **maximum variation** in the data set.

LDA and PCA both form a new set of components.

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

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



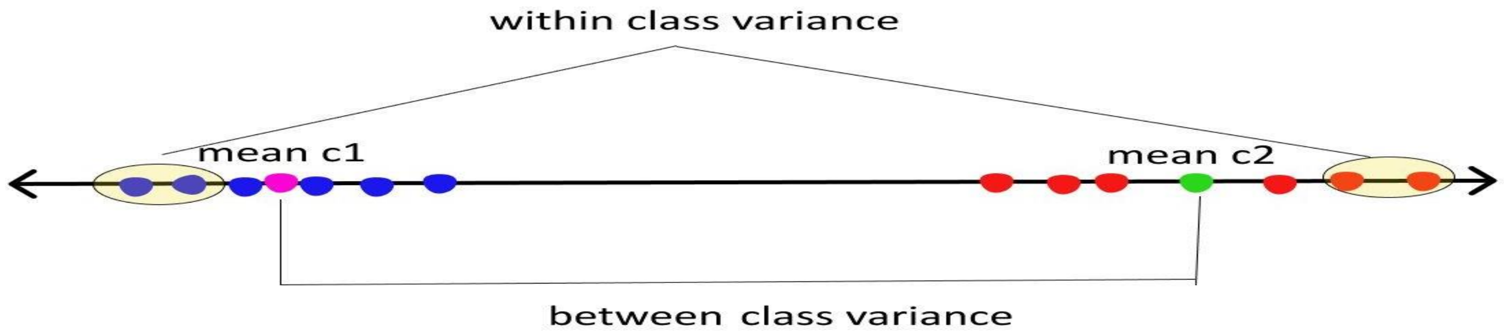


A Brief Introduction to Linear Discriminant Analysis

- **Scatter matrix:** Used to make estimates of the covariance matrix. It is a m X m positive semi-definite matrix. Given by: sample variance * no. of samples.
- Note: Scatter and variance measure the same thing but on different scales. So, we might use both words interchangeably. So, do not get confused.

Here we will be dealing with two types of scatter matrices

- Between class scatter = $\mathbf{S_b}$ = measures the distance between class means
- Within class scatter = $\mathbf{S_w}$ = measures the spread around means of each class
-



Within-class scatter matrix

To calculate the within-class scatter matrix, you can use the following mathematical expression:

$$S_W = \sum_{i=1}^{c} S_i$$

- where, c = total number of distinct classes and

$$S_i = \sum_{x \in D_i}^{n} (\boldsymbol{x} - \boldsymbol{m}_i)\,(\boldsymbol{x} - \boldsymbol{m}_i)^T$$

$$\boldsymbol{m}_i = \frac{1}{n_i} \sum_{x \in D_i}^{n} \boldsymbol{x}_k$$

- where, x = a sample (i.e. a row).

  n = total number of samples within a given class.
- Now we create a vector with the mean values of each feature:

Between-class scatter matrix

- We can calculate the between-class scatter matrix using the following mathematical expression:

Where

$$S_B = \sum_{i=1}^{c} N_i(\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$$

and

$$\boldsymbol{m}_i = \frac{1}{n_i} \sum_{\boldsymbol{x} \in D_i}^{n} \boldsymbol{x}_k$$

Then solve the generalized eigenvalue problen he linear discriminants for:

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

We will sort the eigenvalues from the highest to the lowest since the eigenvalues with the highest values carry the most information about the distribution of data is done. Next, we will first k eigenvectors.

$$S_W^{-1} S_B$$

Finally, we will place the eigenvalues in a temporary array to make sure the eigenvalues map to the same eigenvectors after the sorting is done:

Describe the algorithm with an example:

- Consider a 2-D dataset
- Cl =X1 =(x1,x2) ={(4,1),(2,4),(2,3),(3,6), (4,4)}
- C2=X2=(x1,x2) = {(9,10),(6,8),(9,5),(8,7),(10,8)}

Step 1: Compute within class scatter matrix(Sw)

- Sw= = s1+s2

- s1 is the covariance matrix for class 1 and
- s2 is the covariance matrix for class 2.

- Note : Covariance matrix is to be computed on the Mean Cantered data
- For the given example: mean of C1= (3, 3.6) and
- mean of C2=(8,4, 7.6)
- S1=Transpose of mean centred data * Mean centred data

Computed values s1,s2 and Sw

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

$$S_2 = \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$$

$$S_W = S_1 + S_2$$

$$S_W = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.28 \end{bmatrix}$$

Step 2: Compute between class scatter Matrix(Sb)

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

- Mean 1 (M1) =(3,3.6)
- Mean 2 (M2)=(8,4,7.6)

- (M1-M2) = (3-8.4, 3.6-7.6) = (-5.4, 4.0)

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

$$= \begin{pmatrix} -5.4 \\ -4 \end{pmatrix} \begin{pmatrix} -5.4 & -4 \end{pmatrix} = \begin{pmatrix} 29.16 & 21.6 \\ 21.6 & 16.00 \end{pmatrix}$$

Step 3: Find the best LDA projection vector

- To do this ..compute the Eigen values and eigen vector for the largest eigen value, on the matrix which is the product of : $S_W^{-1} S_B$ = $\begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.76 \end{bmatrix}$

- In this example, highest eigen value is : 15.65 ($\lambda = 15.65$)

Compute inverse of S_W^{-1}

- = $S_W = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.28 \end{bmatrix}$

S_W^{-1} is found by using the formula

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

S_w^{-1}

So, $S_W = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.28 \end{bmatrix}$

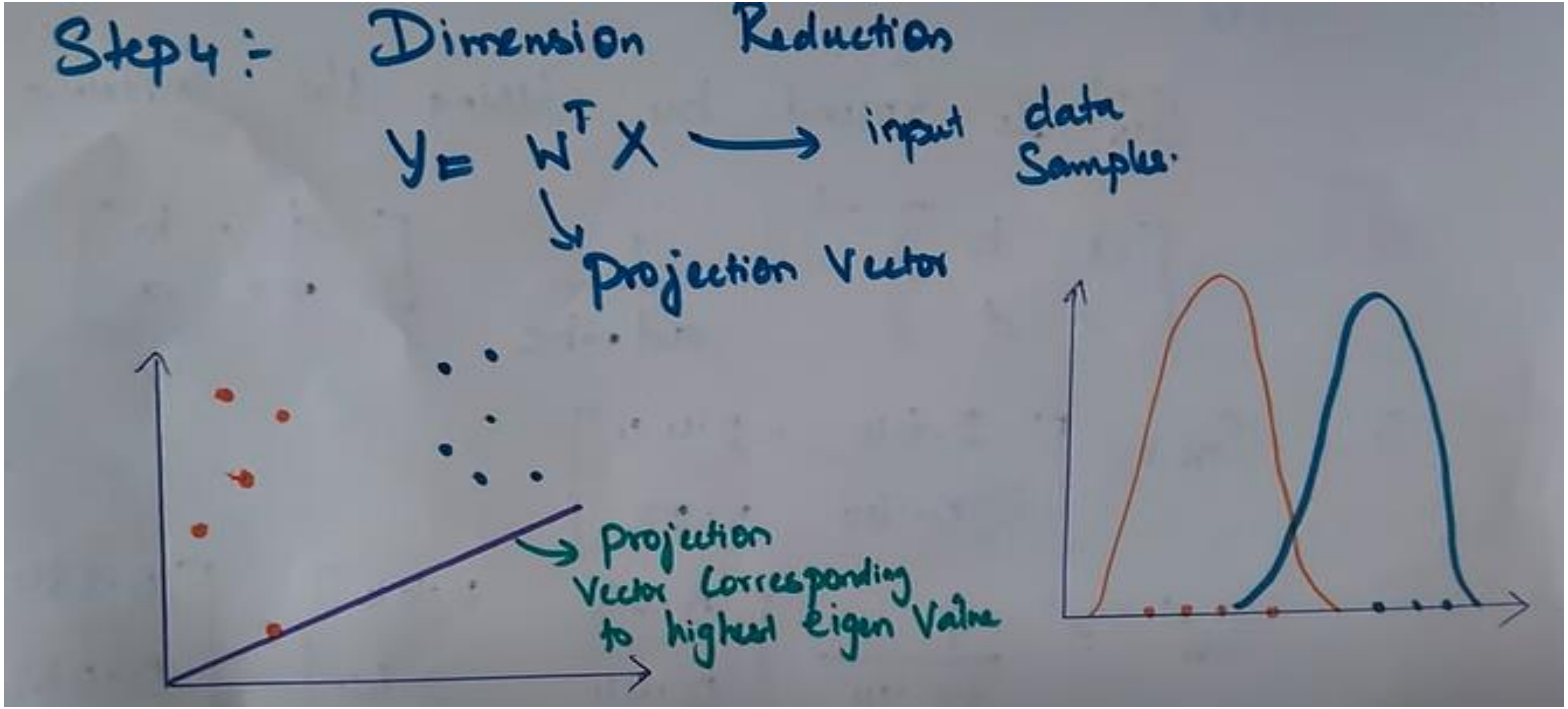
$$S_W^{-1} = \frac{1}{13.74}\begin{bmatrix} 5.28 & 0.44 \\ 0.44 & 2.64 \end{bmatrix} = \begin{bmatrix} 0.384 & 0.032 \\ 0.032 & 0.192 \end{bmatrix}$$

Eigen vector computed for Eigen value: 15.65

$$\begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.76 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = 15.65 \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

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

Step 4: Dimension Reduction



Summary of the Steps

- Step 1 - Computing the within-class and between-class scatter matrices.
- Step 2 - Computing the eigenvectors and their corresponding eigenvalues for the scatter matrices.
- Step 3 - Sorting the eigenvalues and selecting the top k.
- Step 4 - Creating a new matrix that will contain the eigenvectors mapped to the k eigenvalues.
- Step 5 - Obtaining new features by taking the dot product of the data and the matrix from Step 4.

Drawbacks of Linear Discriminant Analysis (LDA)

Although, LDA is specifically used to solve supervised classification problems for two or more classes which are not possible using logistic regression in machine learning. But LDA also fails in some cases where the Mean of the distributions is shared. In this case, LDA fails to create a new axis that makes both the classes linearly separable.

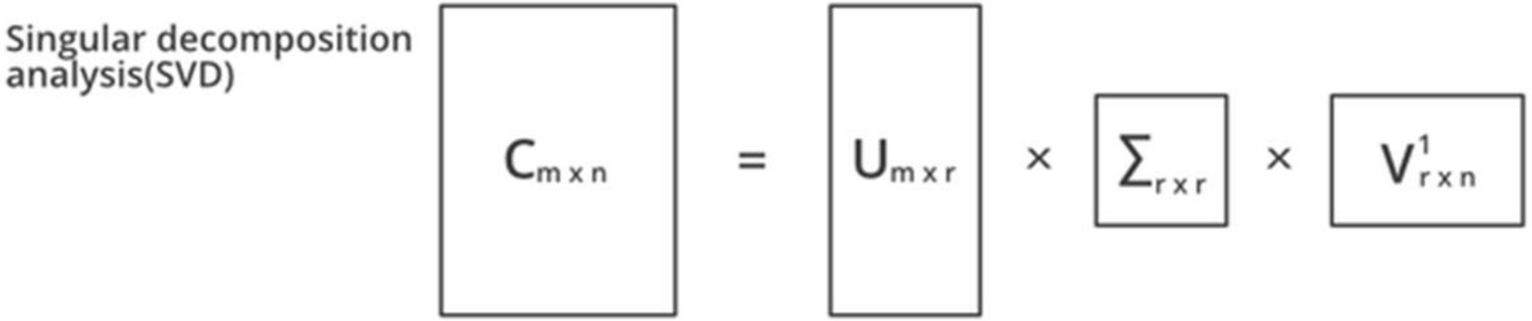
Singular Value Decomposition (SVD)

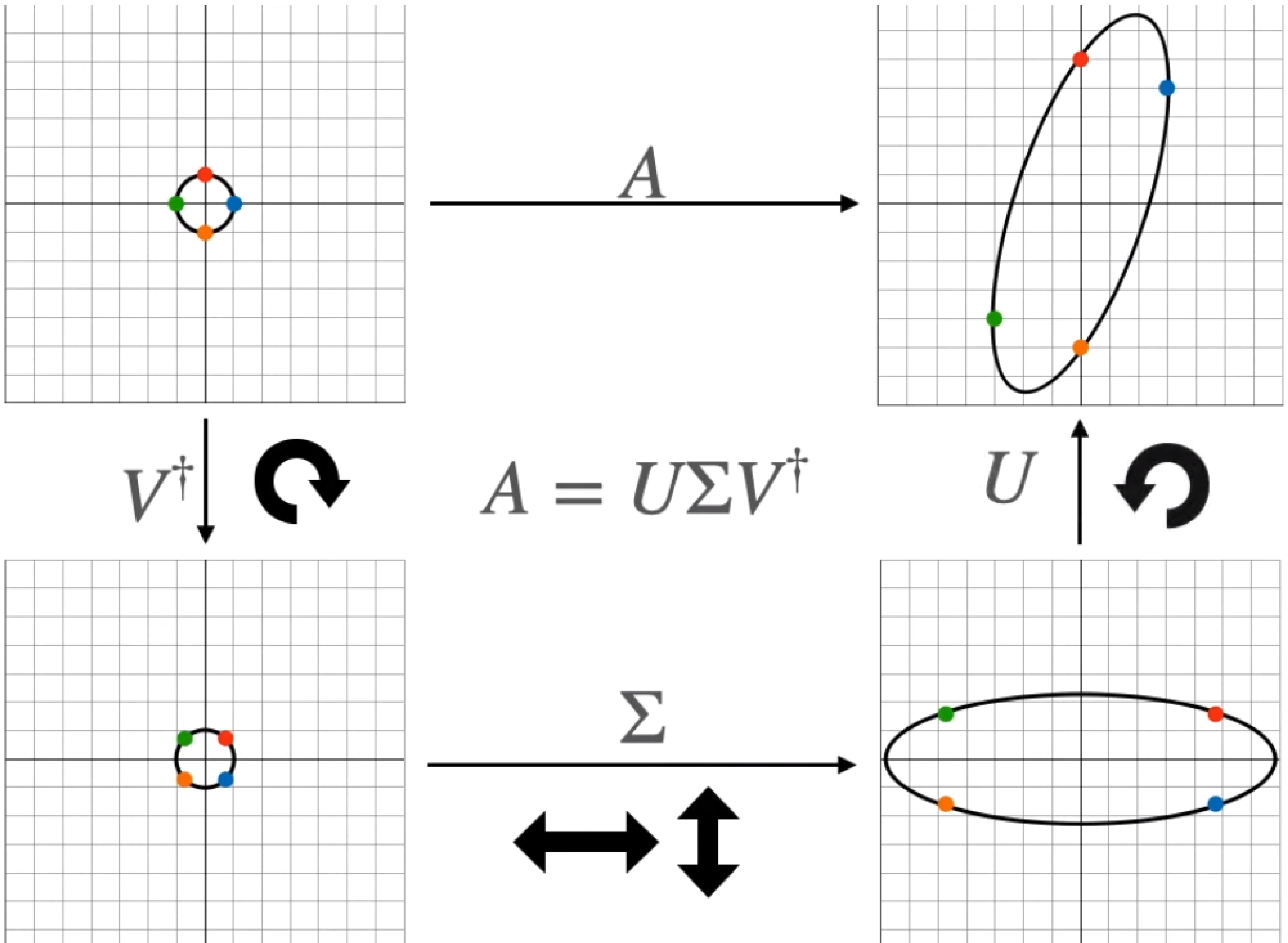
What is singular value decomposition explain with example?

- The singular value decomposition of a matrix A is **the factorization of A into the product of three matrices A = UDV^T where the columns of U and V^T are orthonormal and the matrix D is diagonal with positive real entries**. The SVD is useful in many tasks.
- Calculating the SVD consists of finding the eigenvalues and eigenvectors of AA^T and A^TA.
- The eigenvectors of A^TA make up the columns of V , the eigenvectors of AA^T make up the columns of U.
- Also, the singular values in **S** are square roots of eigenvalues from AA^T or A^TA.
- The singular values are the diagonal entries of the S matrix and are arranged in descending order. The singular values are always real numbers.
- If the matrix A is a real matrix, then U and V are also real.

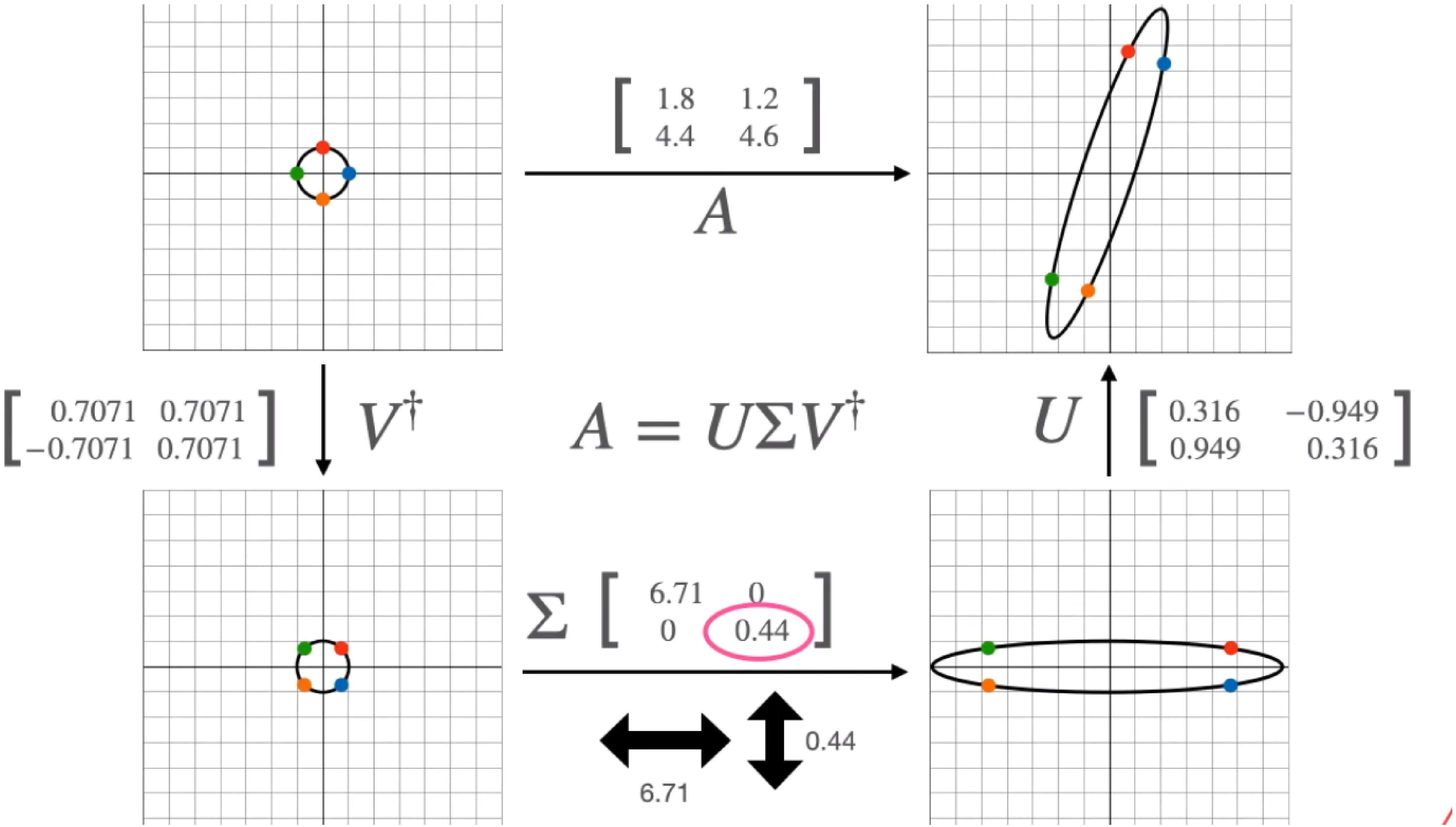
where:

- U: ***mxr*** matrix of the orthonormal eigenvectors of AA^T.
- V^T: transpose of a ***rxn*** matrix containing the orthonormal eigenvectors of A^TA.
- W: a ***rxr*** diagonal matrix of the singular values which are the square roots of the eigenvalues of AA^T and A^TA .





A
$V^\dagger$
$A = U\Sigma V^\dagger$
U
Σ



[1.8 1.2 ; 4.4 4.6]
A
[0.7071 0.7071 ; −0.7071 0.7071]
V†
A = UΣV†
U
[0.316 −0.949 ; 0.949 0.316]
Σ
[6.71 0 ; 0 0.44]
6.71
0.44

$$C = \begin{pmatrix} 5 & 5 \\ -1 & 7 \end{pmatrix}$$

SVD of $C = U \Sigma V^T$

$$C^T C = \begin{pmatrix} 5 & -1 \\ 5 & 7 \end{pmatrix} \begin{pmatrix} 5 & 5 \\ -1 & 7 \end{pmatrix} = \begin{pmatrix} 26 & 18 \\ 18 & 74 \end{pmatrix}$$

COMPUTE EIGEN VALUES

$$|C^T C - \lambda I| = \begin{vmatrix} 26-\lambda & 18 \\ 18 & 74-\lambda \end{vmatrix}$$

$$= \lambda^2 - 100\lambda + 1600$$

$a = 1$
$b = 100$
$c = 1600$

$$\boxed{\begin{matrix} \lambda_1 = 20 \\ \lambda_2 = 80 \end{matrix}}$$

EIGEN VECTORS

$\lambda_1 = 20$

$$(C^T C - 20 I)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 6 & 18 \\ 18 & 54 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$v_1 = \begin{pmatrix} -3/\sqrt{10} \\ 1/\sqrt{10} \end{pmatrix} \qquad v_2 = \begin{pmatrix} 1/\sqrt{10} \\ 3/\sqrt{10} \end{pmatrix}$$

So $V = \begin{pmatrix} -3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & 3/\sqrt{10} \end{pmatrix}$

Σ = square roots of eigen values of $C^T C$ in the diagonal positn

$$= \begin{pmatrix} \sqrt{20} & 0 \\ 0 & \sqrt{80} \end{pmatrix} = \begin{pmatrix} 2\sqrt{5} & 0 \\ 0 & 4\sqrt{5} \end{pmatrix}$$

$$C = U \Sigma V^T$$

$$CV = U \Sigma V^T V$$

$$\boxed{CV = U\Sigma}$$

$$\begin{pmatrix} 5 & 5 \\ -1 & 7 \end{pmatrix} \begin{pmatrix} -3/\sqrt{10} & \frac{1}{\sqrt{10}} \\ 1/\sqrt{10} & 3/\sqrt{10} \end{pmatrix} = U\Sigma$$

$$\Rightarrow \begin{pmatrix} -\sqrt{10} & 2\sqrt{10} \\ \sqrt{10} & 2\sqrt{10} \end{pmatrix} = U \begin{pmatrix} 2\sqrt{5} & 0 \\ 0 & 4\sqrt{5} \end{pmatrix}$$

Hence

$$\begin{pmatrix} 5 & 5 \\ -1 & 7 \end{pmatrix} = \underbrace{\begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}}_{U} \underbrace{\begin{pmatrix} 4\sqrt{5} & 0 \\ 0 & 2\sqrt{5} \end{pmatrix}}_{\Sigma} \underbrace{\begin{pmatrix} 1/\sqrt{10} & -3/\sqrt{10} \\ 3/\sqrt{10} & 1/\sqrt{10} \end{pmatrix}^{T}}_{V^{T}}$$

End of unit 4