Unsupervised Learning

Developer Student Clubs
Al-Azhar University

AGENDA

Unsupervised Learning

Clustering techniques

- K-means
- Hierarchical clustering
- DBSCAN
- Gaussian Mixture Model

Dimension Reduction

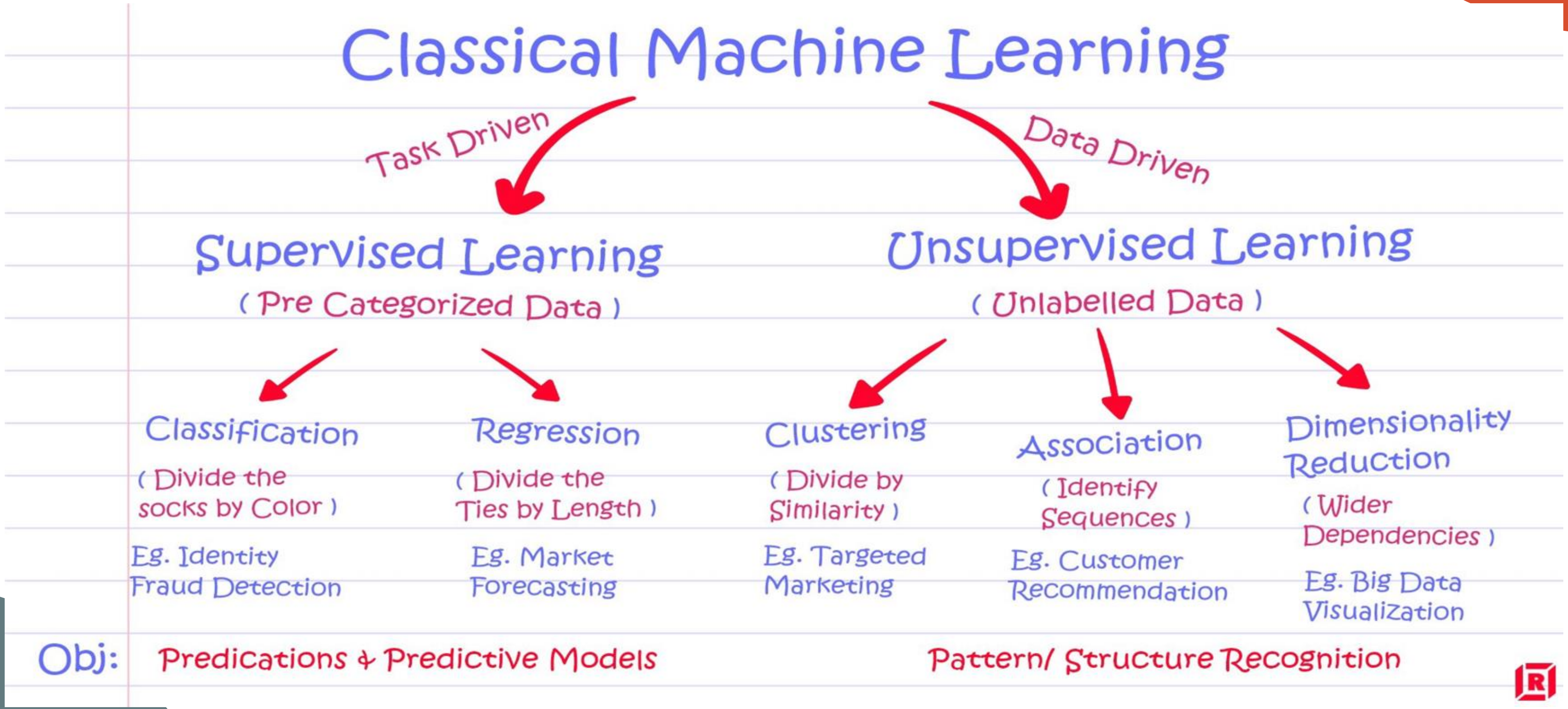
- Principal Components Analysis

01

Unsupervised Learning
VS
Supervised Learning

Developer Student Clubs
Al-Azhar University

Unsupervised Learning VS Supervised Learning



What is Supervised Learning?

Generally speaking, the model is trained on a labeled dataset, so it can predict the outcome of out-of-sample data

Feature

Attribute

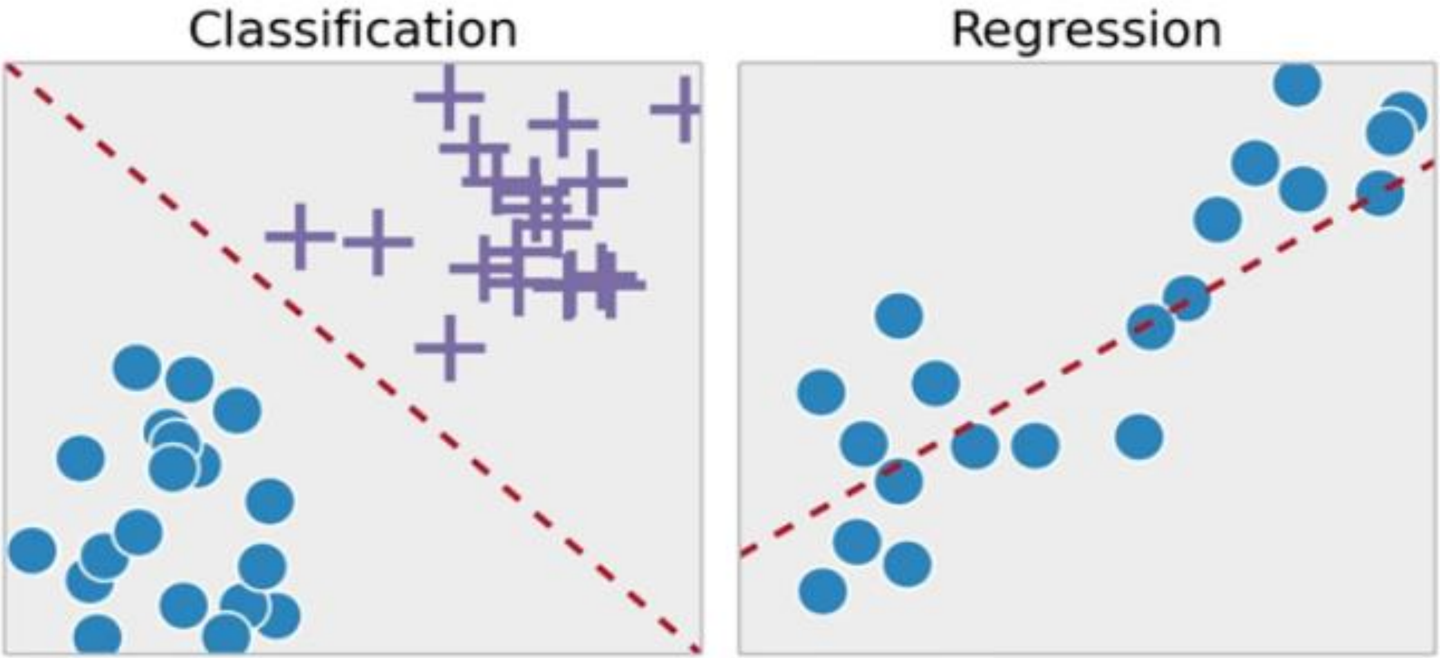
Observation

Numerical Value

Categorical Value

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.2	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
5.0	3.4	1.5	0.2	*I. setosa*
4.4	2.9	1.4	0.2	*I. setosa*
4.9	3.1	1.5	0.1	*I. setosa*
5.4	3.7	1.5	0.2	*I. setosa*
4.8	3.4	1.6	0.2	*I. setosa*
4.8	3.0	1.4	0.1	*I. setosa*

Classification and Regression



SUPERVISED LEARNING

- Target data exist

Training Data

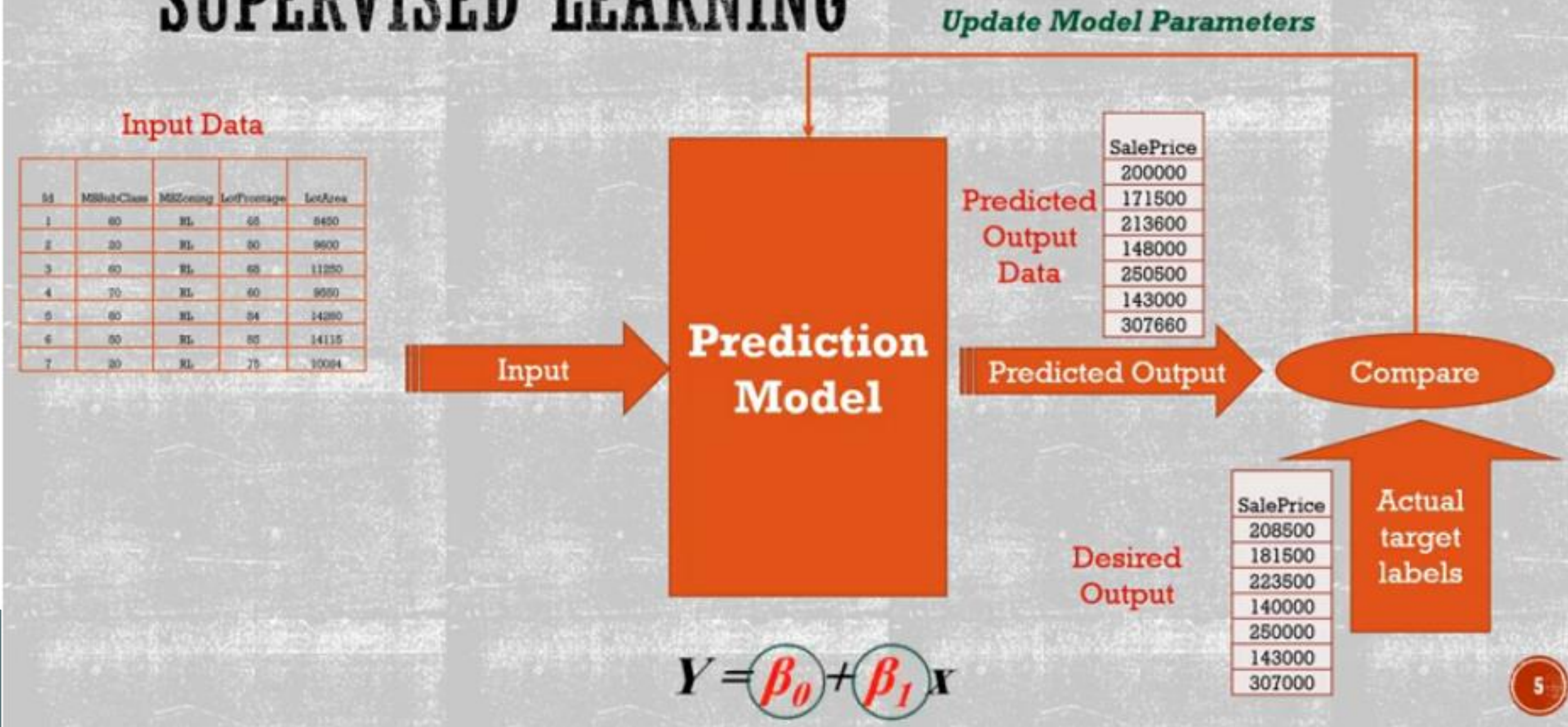
$$Y = \beta_0 + \beta_1 x$$

Input Data

Target Labels

Id	MSSubClass	MSZoning	LotFrontage	LotArea	SalePrice
1	60	RL	65	8450	208500
2	20	RL	80	9600	181500
3	60	RL	68	11250	223500
4	70	RL	60	9550	140000
5	60	RL	84	14260	250000
6	50	RL	85	14115	143000
7	20	RL	75	10084	307000

SUPERVISED LEARNING



Input Data

Id	MSSubClass	MSZoning	LotFrontage	LotArea
1	60	RL	65	8450
2	20	RL	80	9600
3	60	RL	68	11250
4	70	RL	60	9550
5	60	RL	84	14260
6	50	RL	85	14115
7	20	RL	75	10084

Predicted Output Data

SalePrice
200000
171500
213600
148000
250500
143000
307660

Desired Output

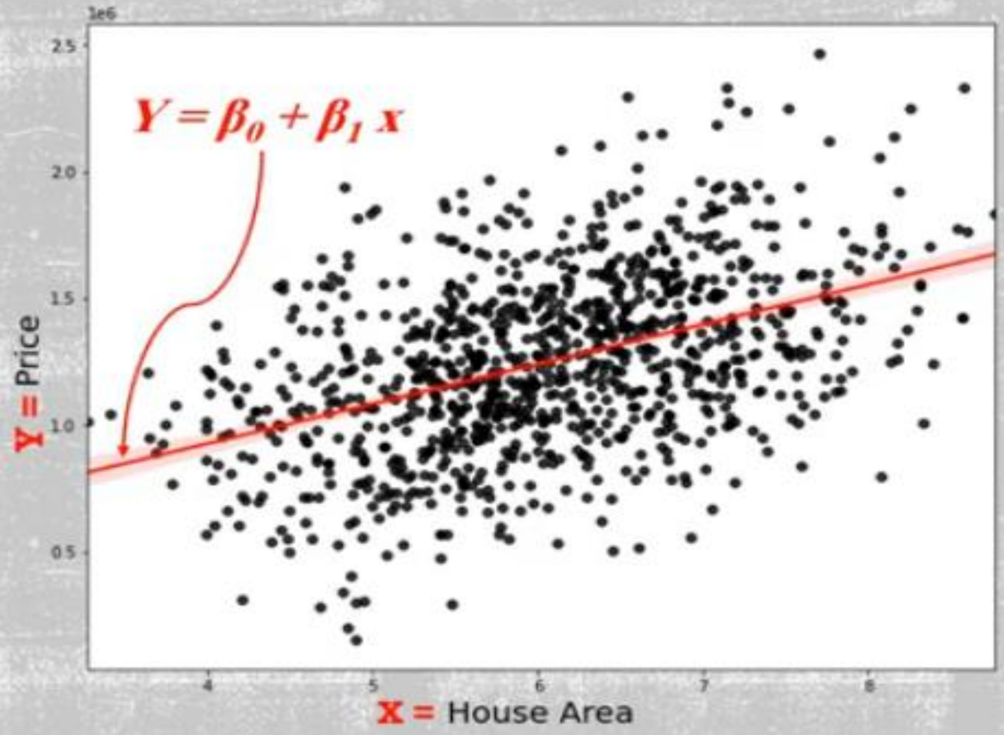
SalePrice
208500
181500
223500
140000
250000
143000
307000

$$Y = \beta_0 + \beta_1 x$$

SUPERVISED LEARNING

- Regression

$$Y = \beta_0 + \beta_1 x$$



SUPERVISED LEARNING

- Classification

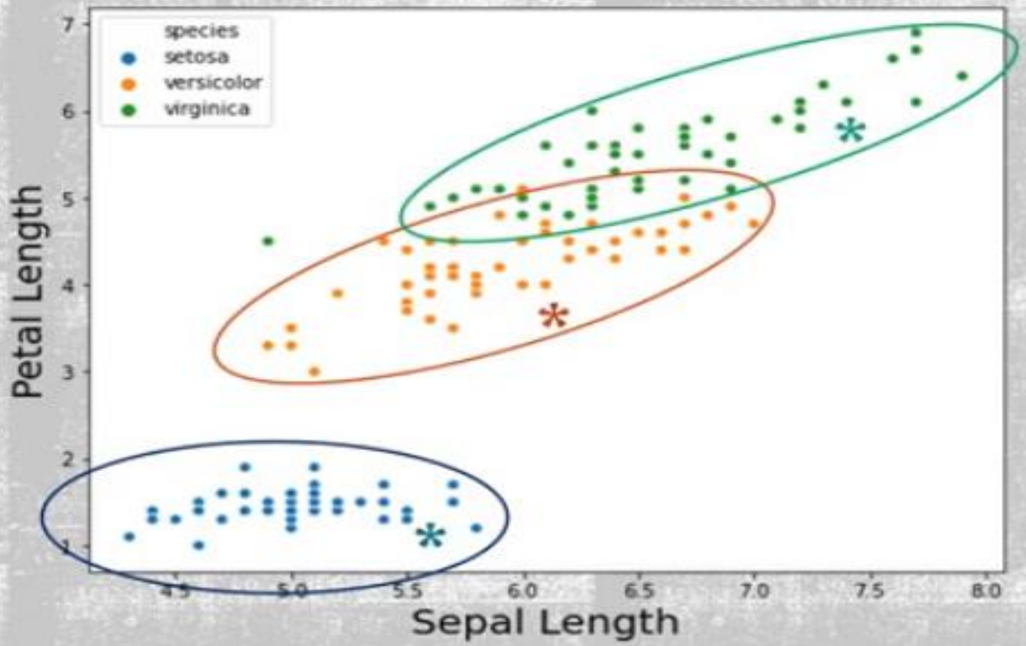
Input Data / Target Labels

sepal_length	sepal_width	petal_length	petal_width	species
6.1	2.8	4.7	1.2	versicolor
5.7	3.8	1.7	0.3	setosa
7.7	2.6	6.9	2.3	virginica
6.0	2.9	4.5	1.5	versicolor
6.8	2.8	4.8	1.4	versicolor
5.4	3.4	1.5	0.4	setosa
5.6	2.9	3.6	1.3	versicolor
6.9	3.1	5.1	2.3	virginica
6.2	2.2	4.5	1.5	versicolor
5.8	2.7	3.9	1.2	versicolor

From: RPubs – IRIS Flower EDA

SUPERVISED LEARNING

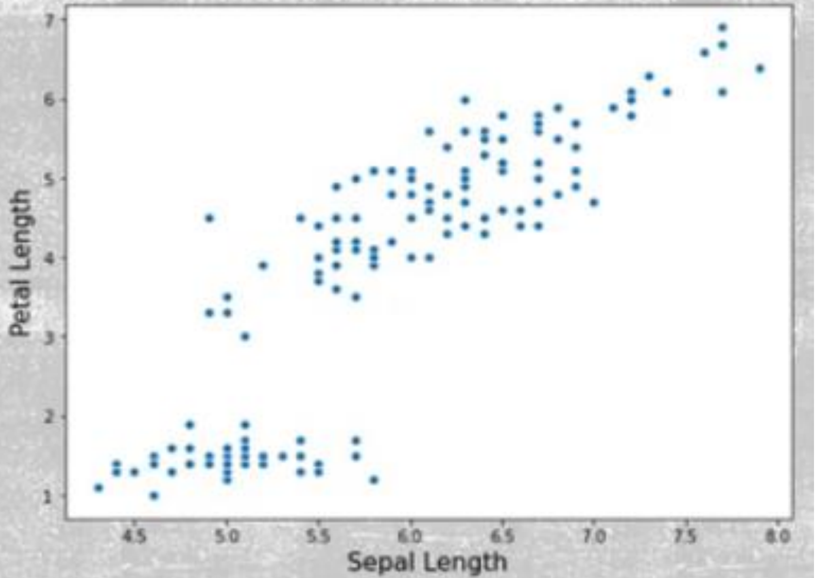
- Classification



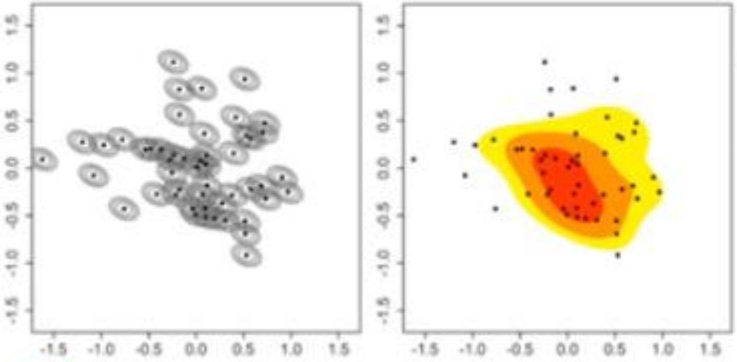
UNSUPERVISED LEARNING

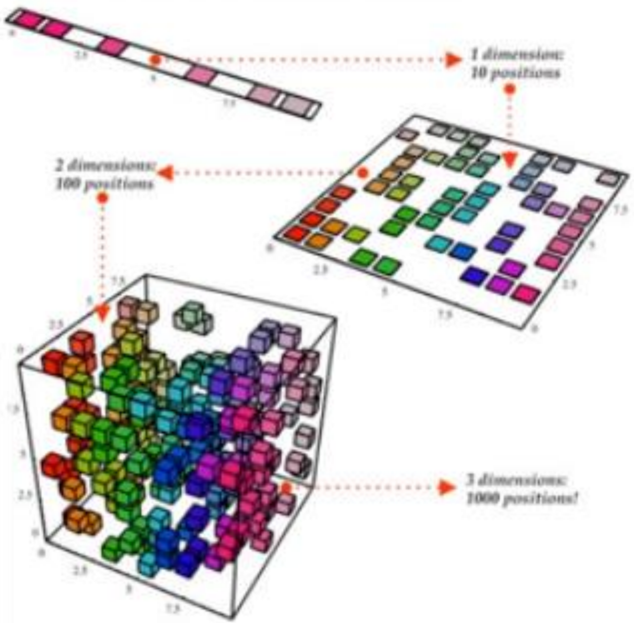
Input Data

sepal_length	sepal_width	petal_length	petal_width
6.1	2.8	4.7	1.2
5.7	3.8	1.7	0.3
7.7	2.6	6.9	2.3
6.0	2.9	4.5	1.5
6.8	2.8	4.8	1.4
5.4	3.4	1.5	0.4
5.6	2.9	3.6	1.3
6.9	3.1	5.1	2.3
6.2	2.2	4.5	1.5
5.8	2.7	3.9	1.2



Difficulties of Unsupervised Learning vs. Supervised Learning





Unsupervised Learning Cases

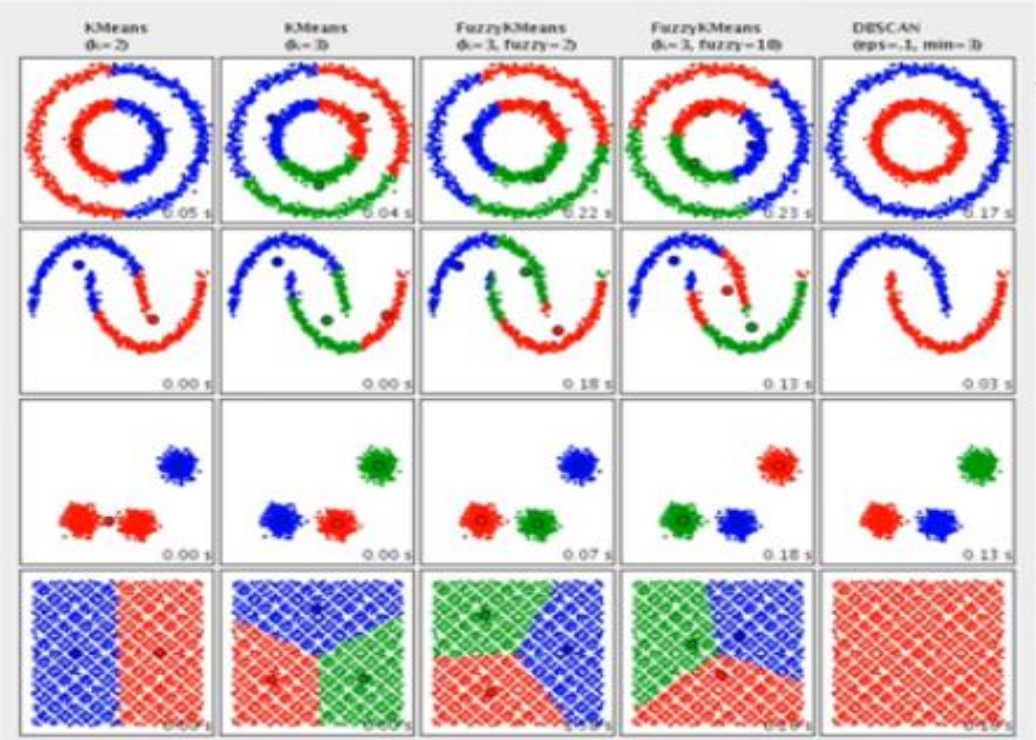
Unsupervised learning algorithms are appropriate for any situation where data is not grouped in advance, often because the features that define the groups aren't known. Examples of unsupervised learning tasks include:

Anomaly detection or fraud detection, as what events constitute an anomaly are unknown and discerned through the model's training process.

Customer segmentation is another unsupervised learning example. In this case, different customer groups are created based upon features like their responses to marketing strategies.

Recommendation systems, where the features of viewed media are analyzed to group users together based on similar tastes in media.

What is Unsupervised Learning?

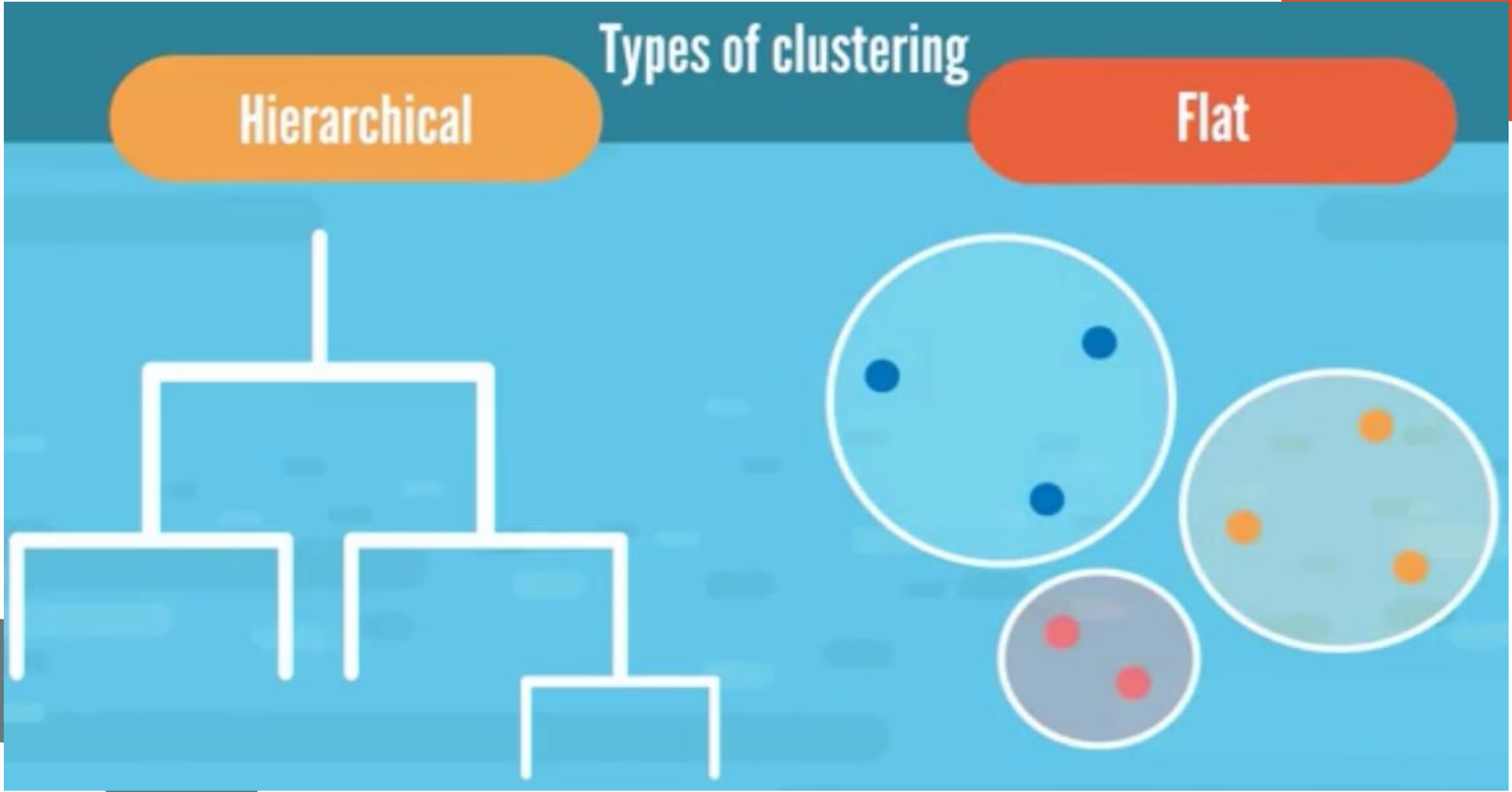


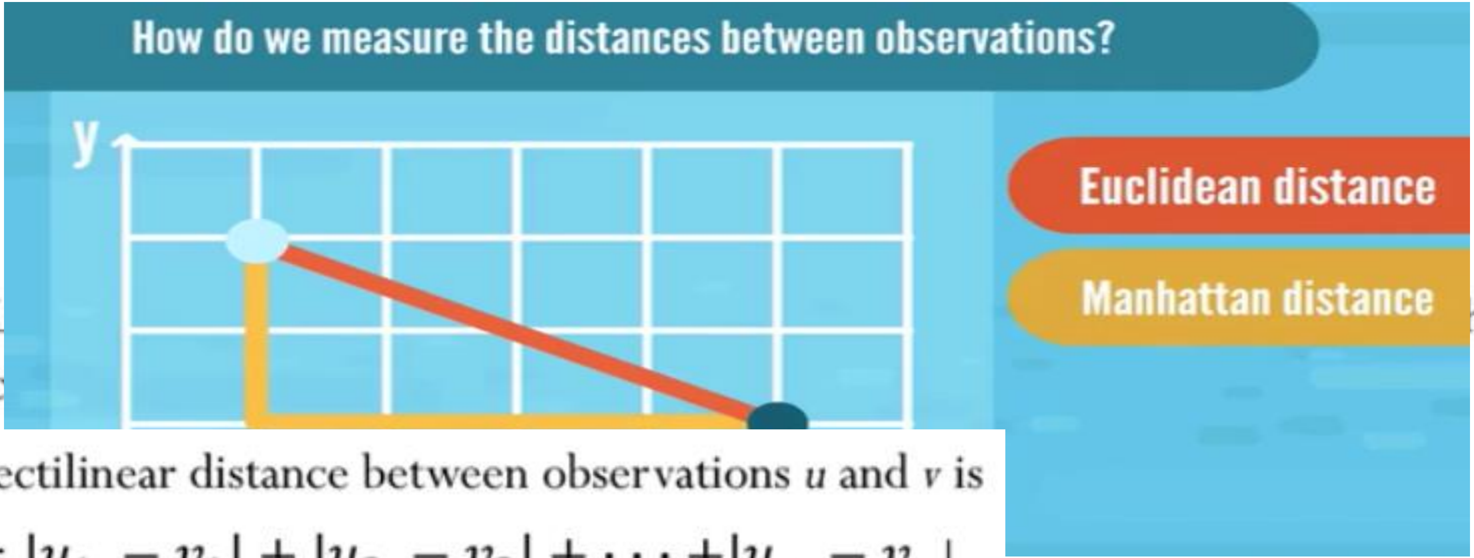
02

Clustering Techniques

Clustering

Clustering is similar to classification, but the basis is different. In Clustering you don't know what you are looking for, and you are trying to identify some segments or clusters in your data. When you use clustering algorithms on your dataset, unexpected things can suddenly pop up like structures, clusters and groupings you would have never thought of otherwise.


Types of clustering
Hierarchical
Flat



- The Rectilinear distance between observations *u* and *v* is
- $d_{u,v} = |u_1 - v_1| + |u_2 - v_2| + \cdots + |u_q - v_q|$

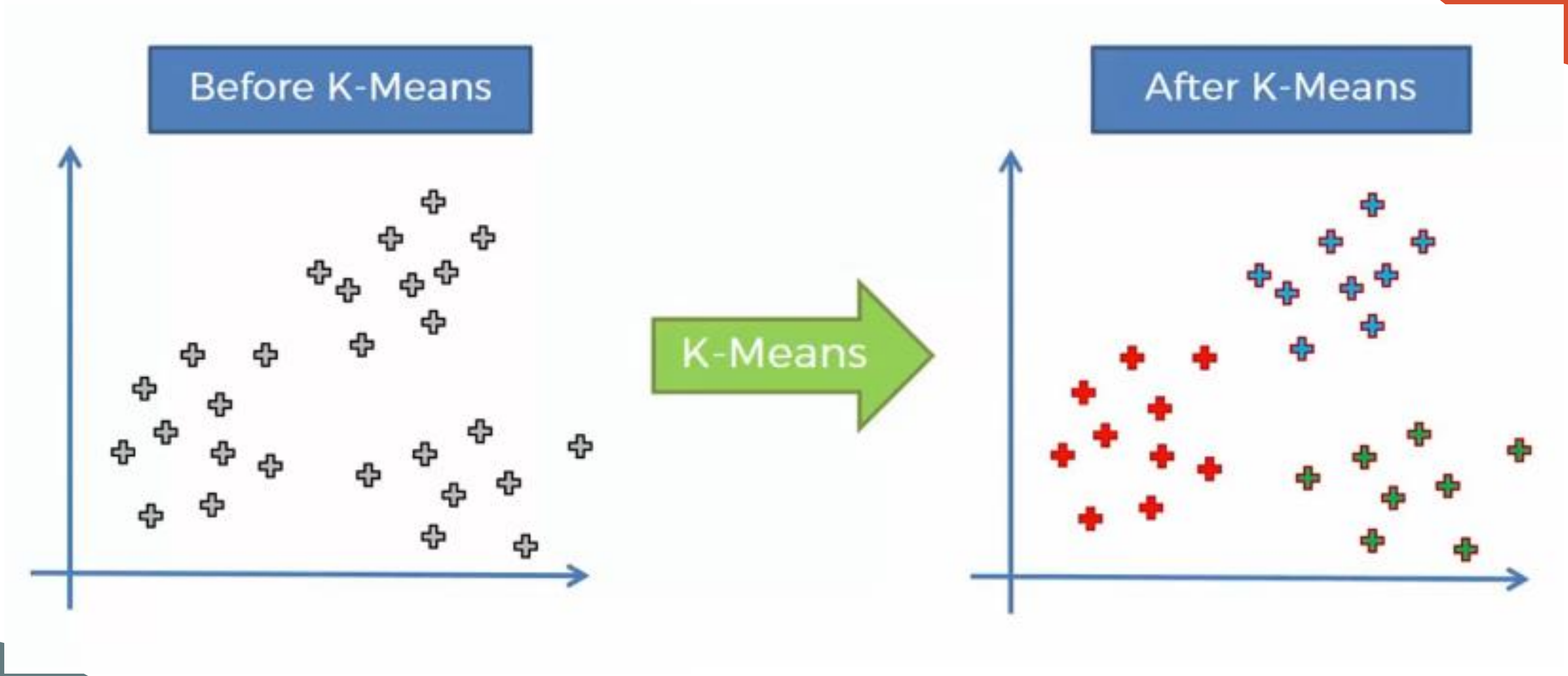
- The Euclidean distance between observations *u* and *v* is
- $d_{u,v} = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \cdots + (u_q - v_q)^2}$

Clustering techniques

- K-means
- Hierarchical clustering
- DBSCAN
- Gaussian Mixture Model

03

K-means


Before K-Means
K-Means
After K-Means

How does it work

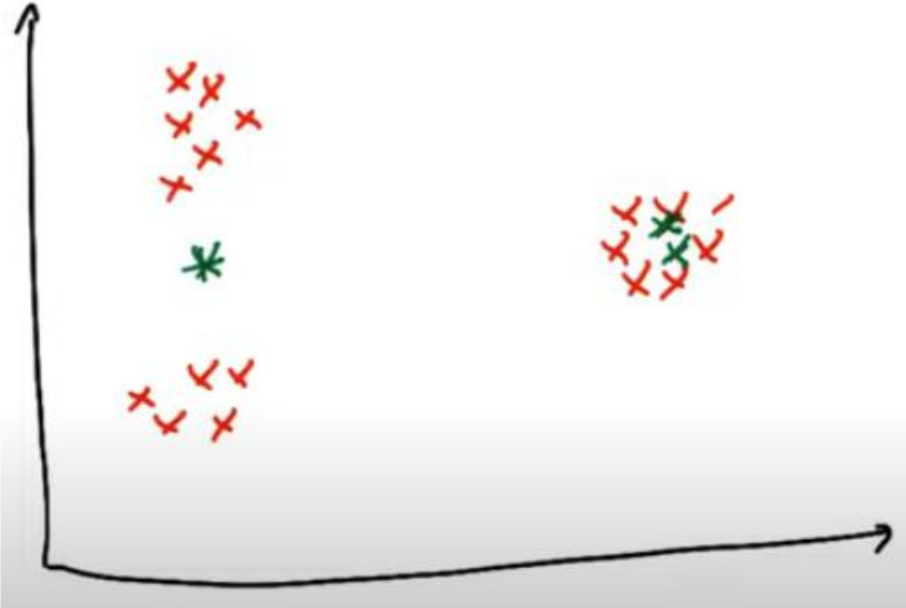
STEP 1: Choose the number K of clusters

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)

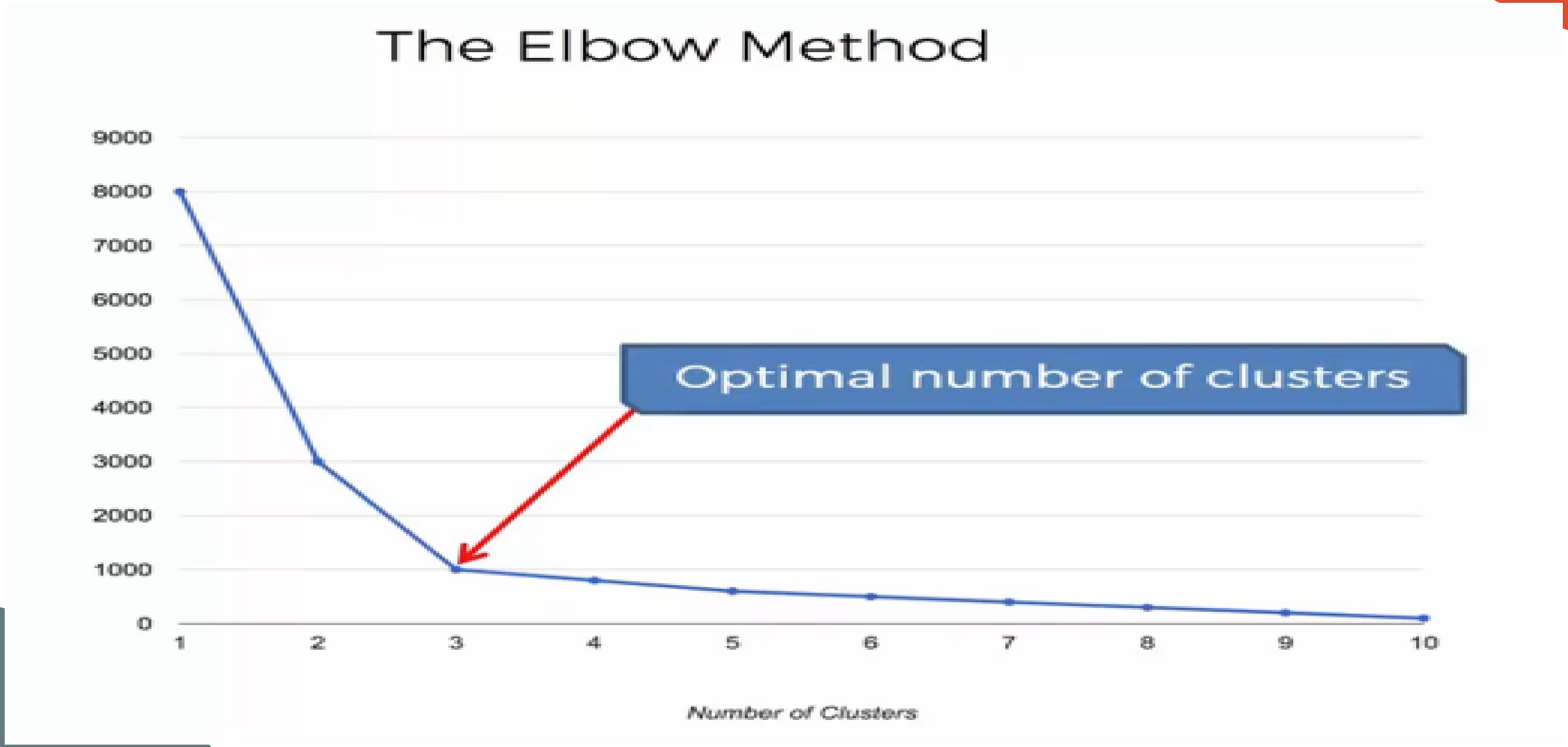
STEP 3: Assign each data point to the closest centroid ➡ That forms K clusters

STEP 4: Compute and place the new centroid of each cluster

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.

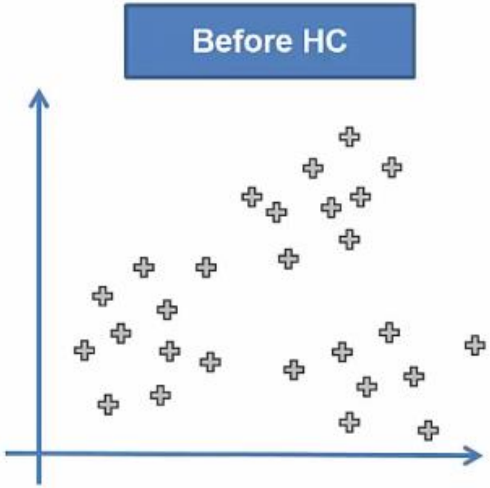


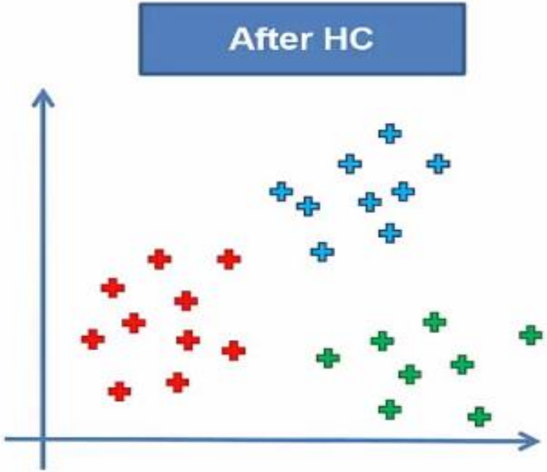
WCSS

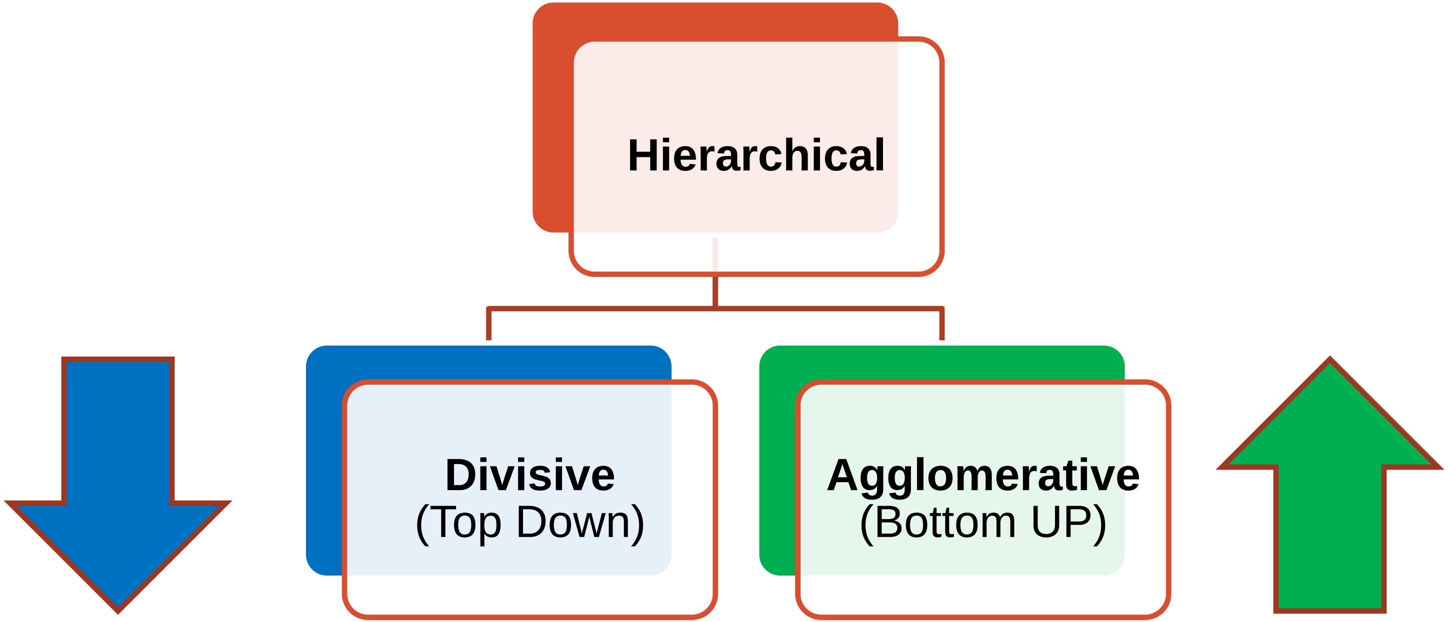

The Elbow Method
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
6
7
8
9
10
Optimal number of clusters
Number of Clusters

04

Hierarchical clustering


Before HC


After HC


Hierarchical
Divisive
(Top Down)
Agglomerative
(Bottom UP)

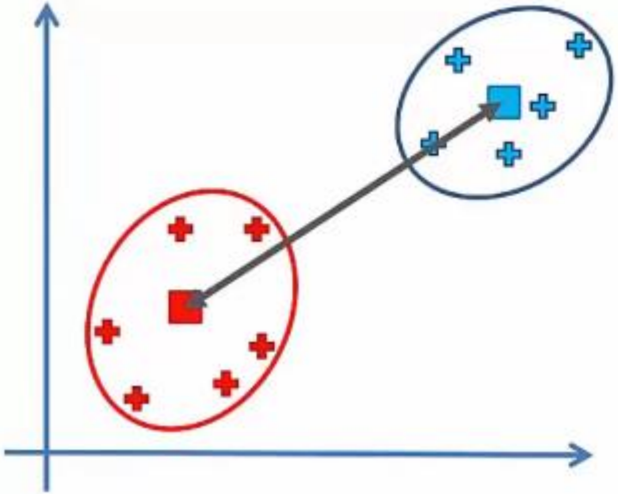
STEP 1: Make each data point a single-point cluster ➡ That forms N clusters

STEP 2: Take the two closest data points and make them one cluster ➡ That forms N-1 clusters

STEP 3: Take the two closest clusters and make them one cluster ➡ That forms N - 2 clusters

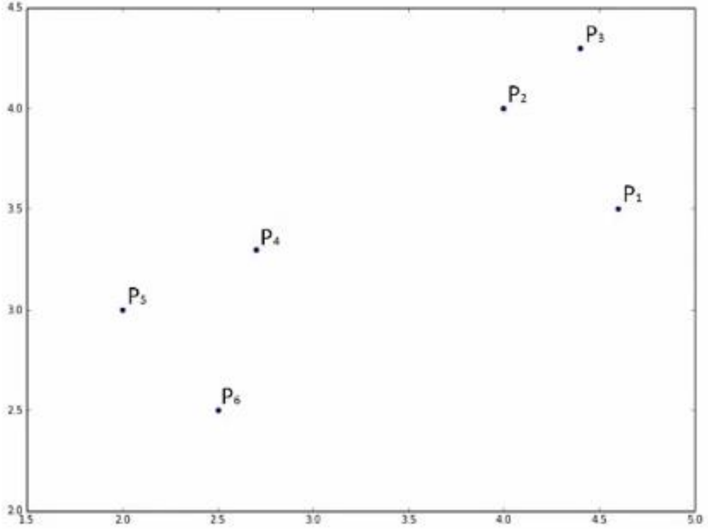
STEP 4: Repeat STEP 3 until there is only one cluster

FIN

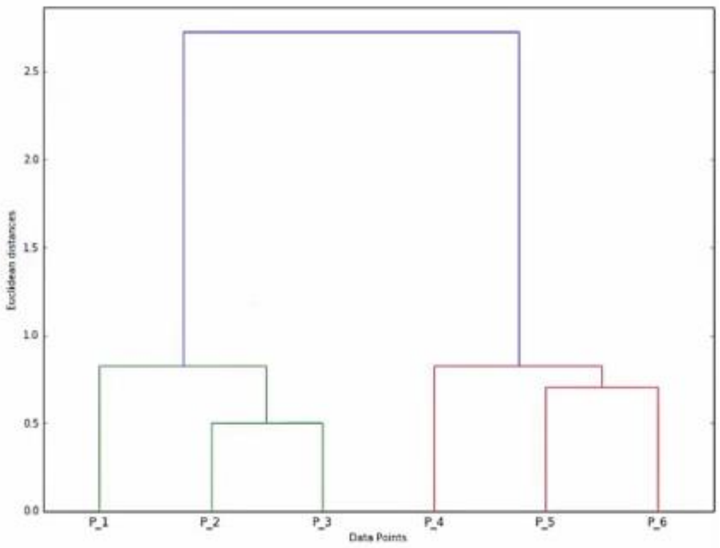


Distance Between Two Clusters:

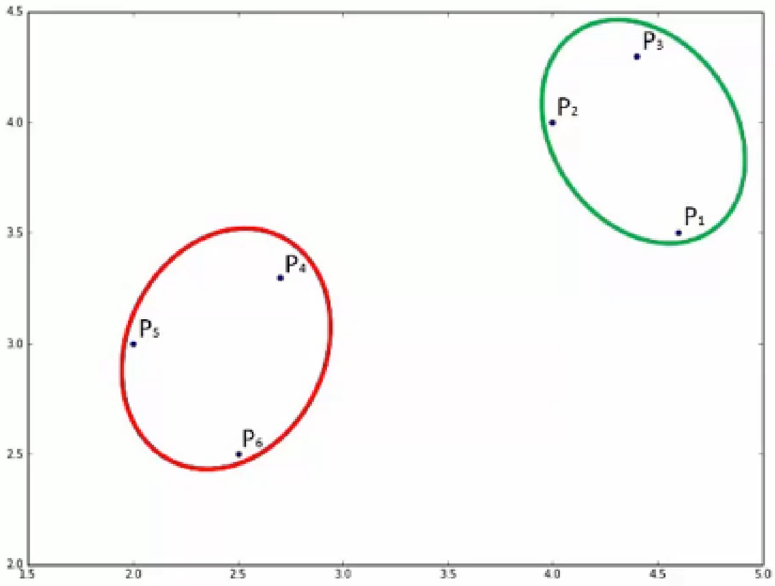
- Option 1: Closest Points
- Option 2: Furthest Points
- Option 3: Average Distance
- Option 4: Distance Between Centroids



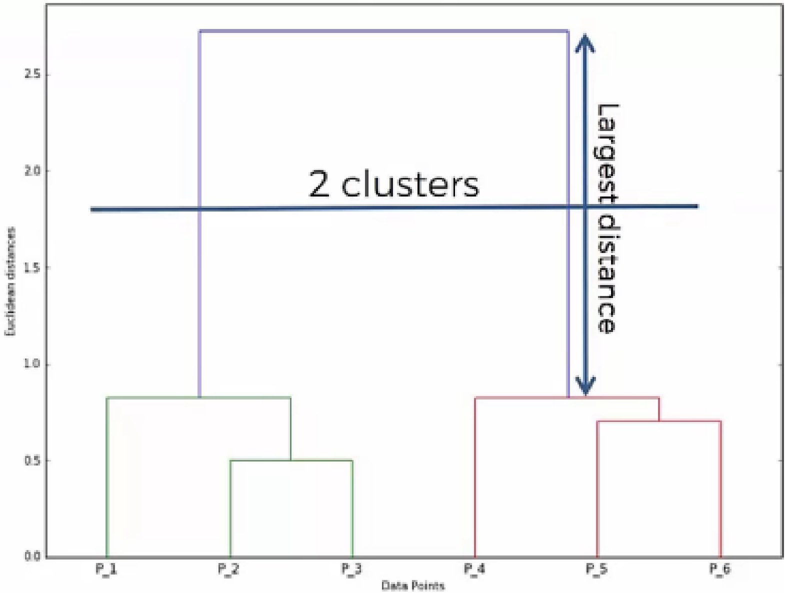
P1
P2
P3
P4
P5
P6



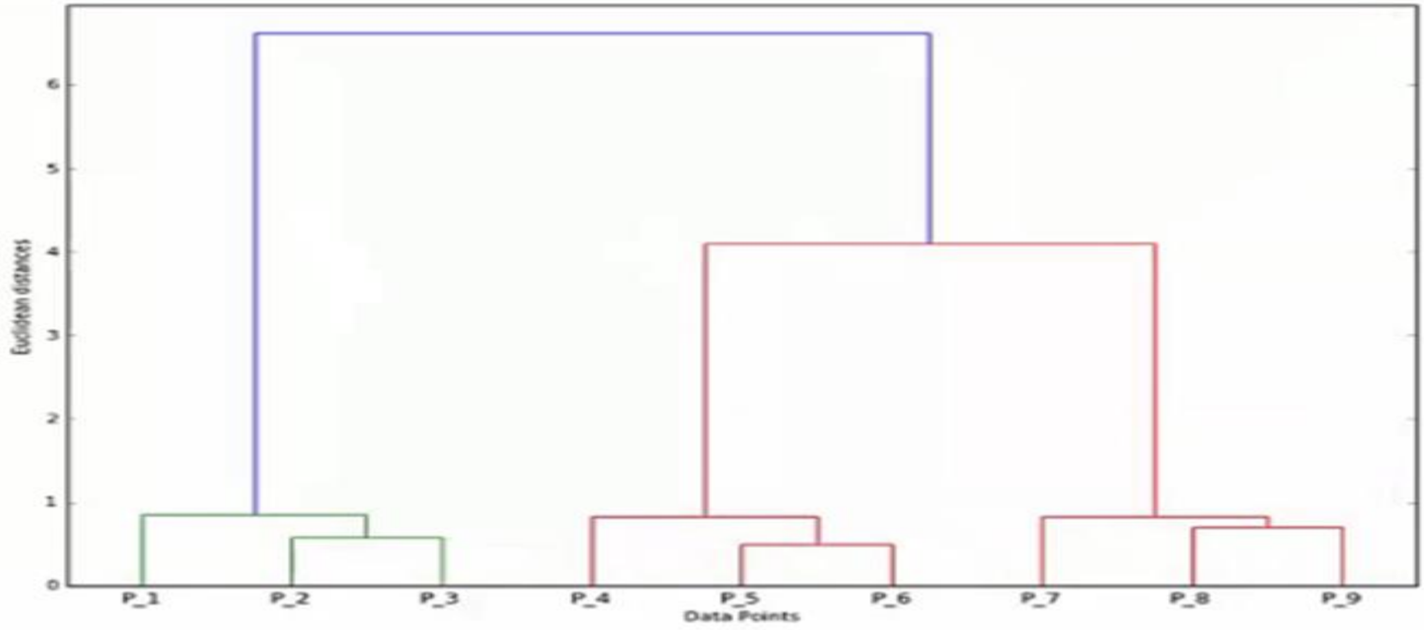
Euclidean distances
Data Points
P_1
P_2
P_3
P_4
P_5
P_6



P1
P2
P3
P4
P5
P6



2 clusters
Largest distance
Euclidean distances
Data Points
P_1
P_2
P_3
P_4
P_5
P_6



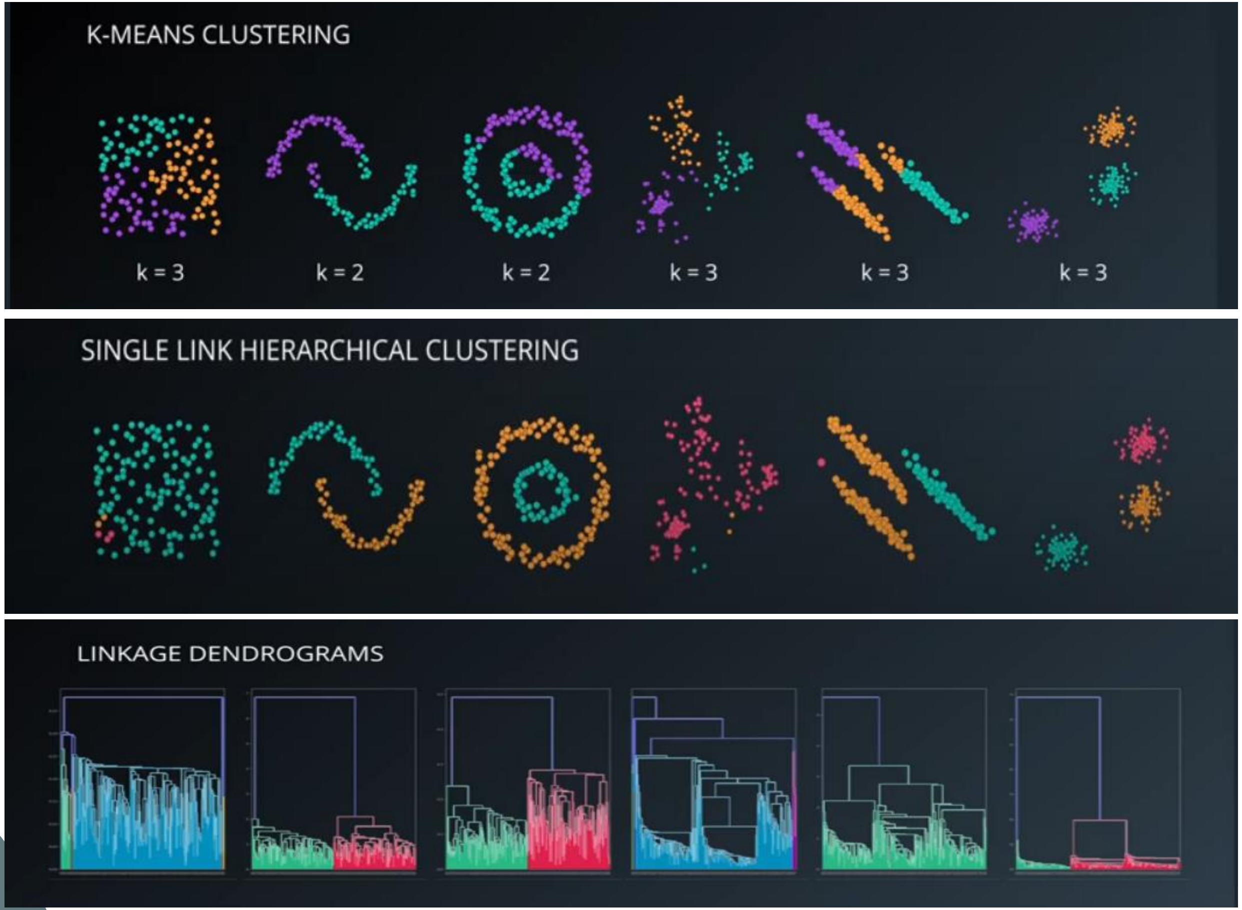
Euclidean distances
6
5
4
3
2
1
0
P_1
P_2
P_3
P_4
P_5
P_6
P_7
P_8
P_9
Data Points

ADVANTAGES:

- Resulting hierarchical representation can be very informative
- Provides an additional ability to visualize
- Especially potent when the dataset contains real hierarchical relationships (e.g. Evolutionary biology)

DISADVANTAGES:

- Sensitive to noise and outliers
- Computationally intensive $O(N^2)$


K-MEANS CLUSTERING
k = 3
k = 2
k = 2
k = 3
k = 3
k = 3
SINGLE LINK HIERARCHICAL CLUSTERING
LINKAGE DENDROGRAMS

05

Dimension Reduction

–Dimension Reduction

There are two types of Dimensionality Reduction techniques:

Feature Selection

Feature Extraction

06

Principal Components Analysis -PCA-

Principal Components Analysis
-PCA-

Dimensionality Reduction Method.

Transform a large set of variables into a smaller one that still contains most of the information in the large set.

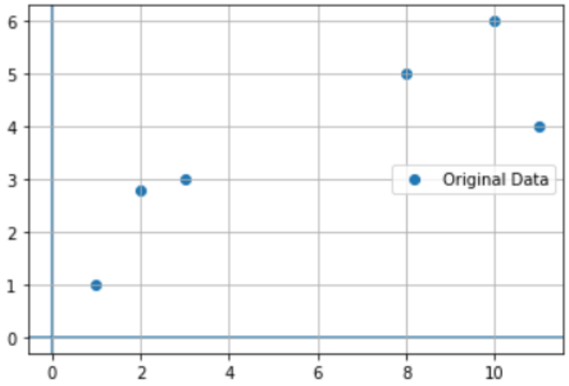
Used in Data compression:

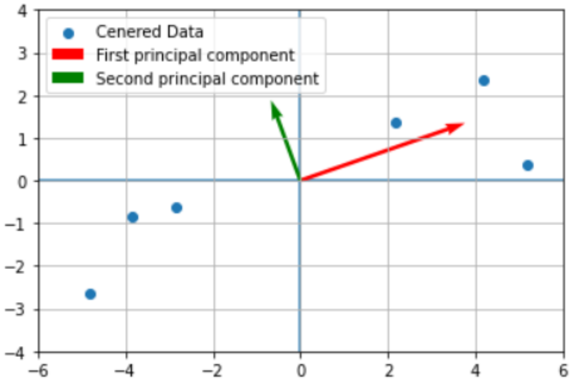
Save data.

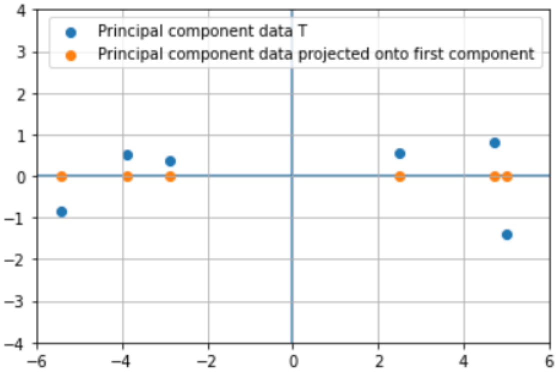
Speed up learning algorithm.

Data visualization (Reduce high dimension data to 3D or 2D).

Linear Algebra Notes
Transformation - Change Basis





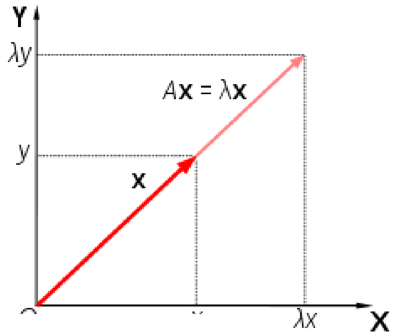


Eigenvalues and Eigen vectors of a matrix

- Suppose we have a transformation matrix A and we apply this transformation to vector x. This will be equivalent to stretching (or diminishing) the vector x by a scalar factor λ.

$$Ax = \lambda x$$

$$(A - \lambda I)x = 0$$



- A is n × n matrix and x is n dimensional vector
- The above equation has a non-zero solution iff the determinant of the matrix $(A - \lambda I)$ is zero i.e.

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

Applications: Eigenvalues & Eigen ve

It is used in PCA to reduce the dimensionality of data samples.

It is also used to do several matrix multiplications (e.g. n times) more computationally efficient (using diagonalization).

Changing to Eigen basis

$T = C\,D\,C^{-1}$ where,

T: transformation matrix.

C: matrix of eigenvectors.

D: diagonal matrix that contains eigenvalues.

$$T^n = C\,D^n\,C^{-1}$$

Principal Component Analysis (PCA)

Steps

1. Standardization.
2. Covariance matrix computation.
3. Eigenvectors and Eigenvalues of the covariance matrix.
4. Feature vector.

Principal Component Analysis (PCA): 1-Standardization

Given m observations and n number of features, X is the data matrix, and x_i the data from the i^{th} sample.

x_{ij} is the j^{th} reading from i^{th} sample

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij}$$

$$s_j^2 = \frac{1}{m-1}\sum_{i=1}^{m}\left(x_{ij} - \mu_j\right)^2$$

$$z_{ij} = \frac{x_{ij} - \mu_j}{s_j}$$

Z is the scaled data matrix, each feature has mean equal to zero and standard deviation equal to one.

$$X = \begin{bmatrix} - & x_1^T & - \\ - & x_2^T & - \\ - & \vdots & - \\ - & x_m^T & - \end{bmatrix}$$

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{in} \end{bmatrix}$$

$$Z = \begin{bmatrix} - & z_1 & - \\ - & z_2 & - \\ - & \vdots & - \\ - & z_m & - \end{bmatrix}$$

Principal Component Analysis (PCA): 2-Covariance matrix computation.

Covariance matrix is a $n \times n$ symmetric matrix.

Capture the relationship between the features of the input data set.

Correlations between all the possible pairs of variables.

Sometimes, features are highly correlated in such a way that they contain redundant information.

$$cov(a,b) = cov(b,a)$$

Positive number: increase or decrease together. (Correlated)

Negative number: One increase other decrease (Inversely correlated).

$$C = Z^T Z = \begin{bmatrix} cov(z_{i1}, z_{i1}) & cov(z_{i1}, z_{i2}) & \cdots & cov(z_{i1}, z_{in}) \\ cov(z_{i2}, z_{i1}) & cov(z_{i2}, z_{i2}) & \cdots & cov(z_{i2}, z_{in}) \\ \vdots & \vdots & \ddots & \vdots \\ cov(z_{in}, z_{i1}) & \cdots & \cdots & cov(z_{in}, z_{in}) \end{bmatrix}$$

PCA:
3-Eigenvectors and Eigenvalues of the covariance matrix.

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

The principal components are artificial features

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

To compute how much variance captured in the j^{th}component: $\frac{\lambda_j}{\sum_{k=1}^{n} \lambda_k} = \frac{\lambda_j}{trace(D)}$

$$Cv_1 = \lambda_1 v_1, Cv_2 = \lambda_2 v_2, \dots, Cv_n = \lambda_n v_n$$

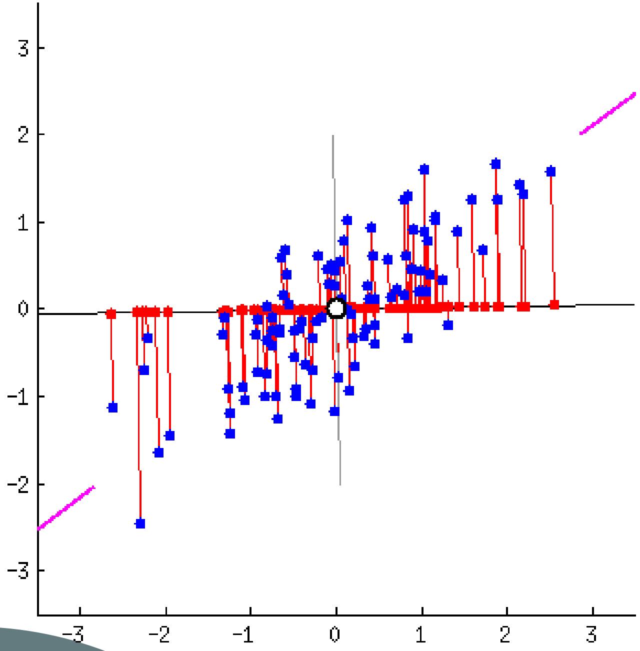
$$\lambda_1 > \lambda_2 > \dots > \lambda_n$$

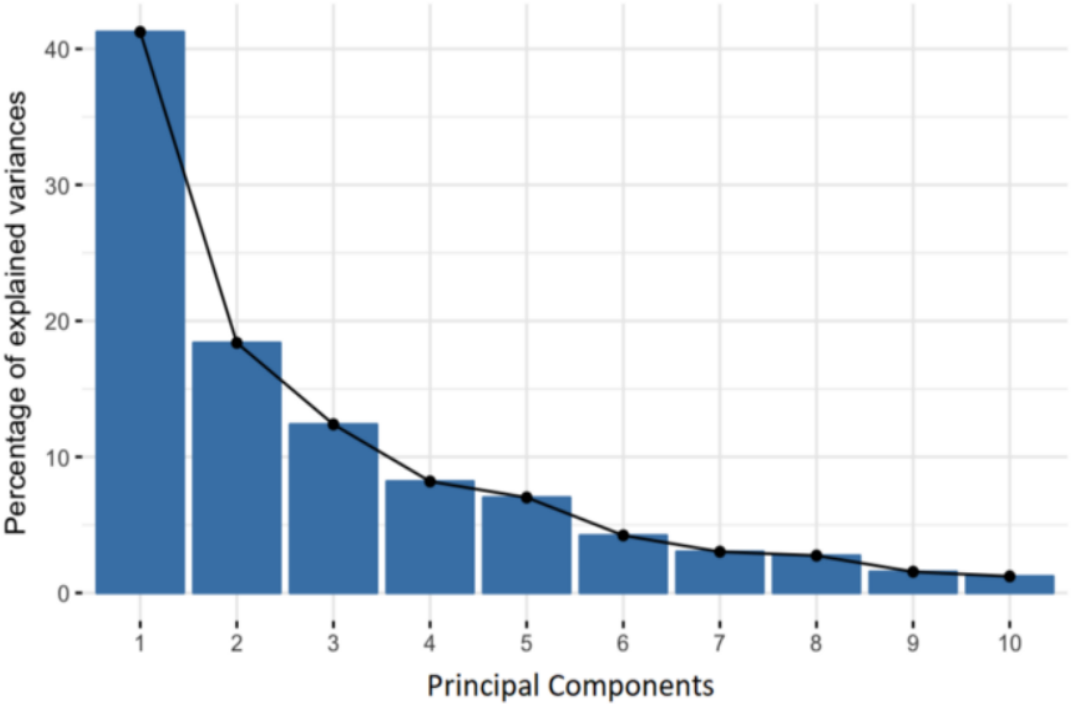
$$D = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}$$

$$CV = VD$$

$$T = ZV$$

Principal Component Analysis (PCA): 4-Feature vector





References

1-Artificial Intelligence Prognostics in Engineering
https://www.youtube.com/channel/UCyGxExVxl3prnzGXjJIwzIQ

2-Udactiy-------------Intro to Machine Learning
3-udemy -------------- Machine Learning Nanodegree

THANKS

Any questions?

Developer Student Clubs
Al-Azhar University