Machine Learning

Dr Eman Ruslan CLS

Machine Learning

Unsupervised Learning

Types of Unsupervised Learning Algorithm

- Unsupervised Learning algorithms are classified into two categories.
  - **Clustering:** Clustering is a technique of grouping objects into clusters.
  - **Association:** helps in finding the relationships between variables in a large database.

Clustering

C 1 C 2

Class 1 Class 3

Class 2 Class 4

Classification

Clustering Application

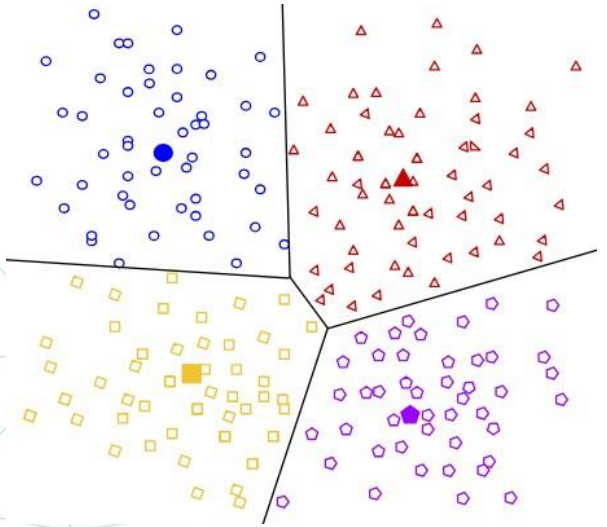
- Market Segmentation
- Image segmentation

Types of Clustering

- **Centroid-based Clustering**
- **Density-based Clustering**
- **Distribution-based Clustering**
- **Hierarchical Clustering**

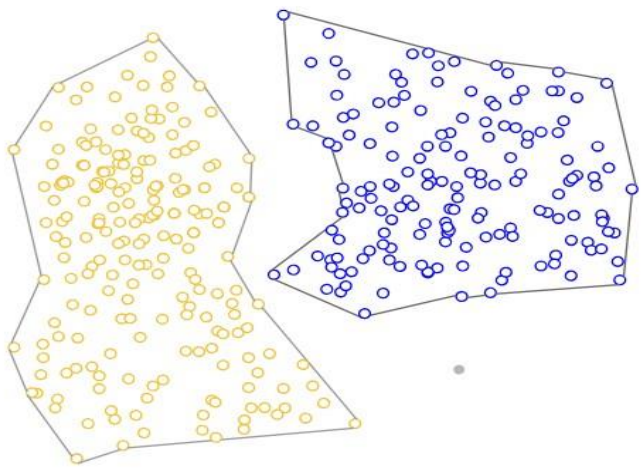
Centroid-based clustering

- **Centroid-based clustering** organizes the data into non-hierarchical clusters, in contrast to hierarchical clustering defined below.
- k-means is the most widely-used centroid-based clustering algorithm. Centroid-based algorithms are efficient but sensitive to initial conditions and outliers.



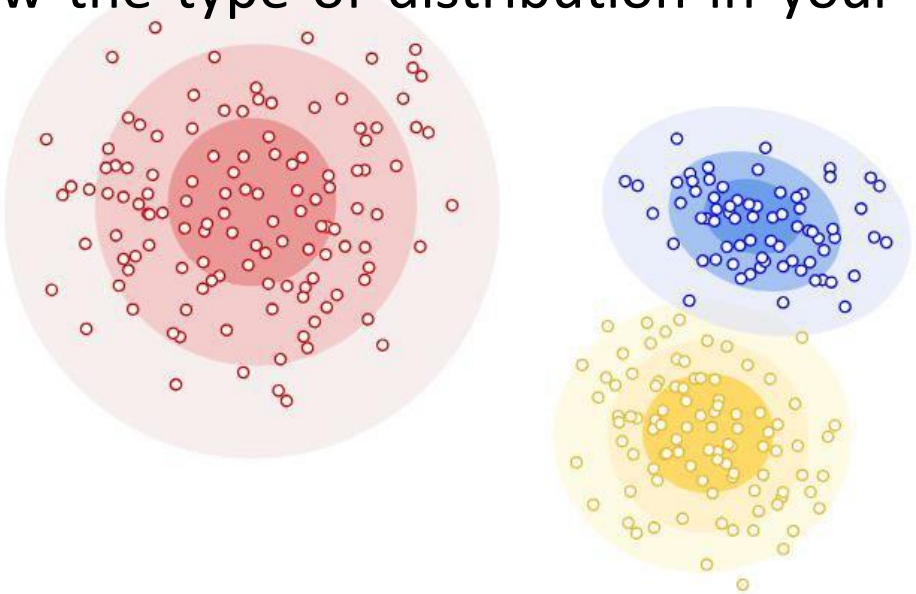
Density-based clustering

- Density-based clustering connects areas of high example density into clusters.
- This allows for arbitrary-shaped distributions as long as dense areas can be connected.
- These algorithms have difficulty with data of varying densities and high dimensions. Further, by design, these algorithms do not assign outliers to clusters.



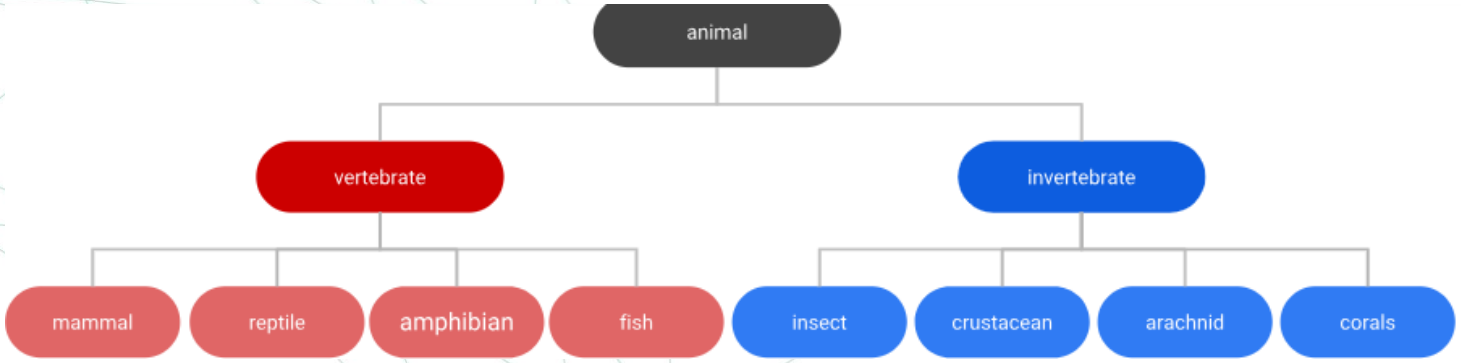
Distribution-based Clustering

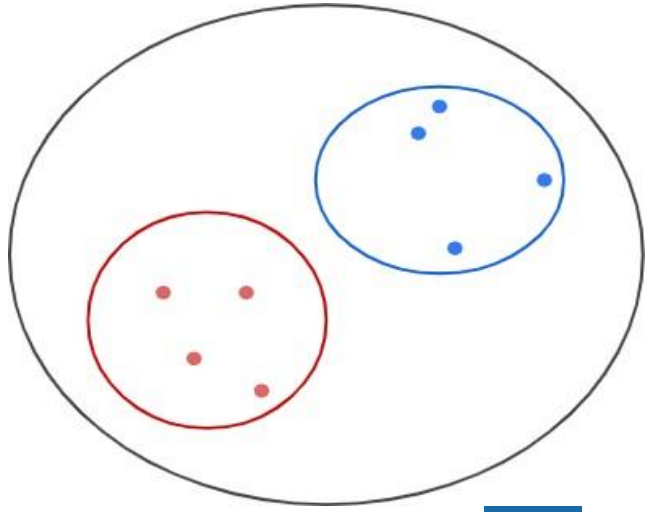
- This clustering approach assumes data is composed of distributions, such as **Gaussian distributions**.
- As distance from the distribution's center increases, the probability that a point belongs to the distribution decreases. The bands show that decrease in probability.
- When you do not know the type of distribution in your data, you should use a different algorithm.



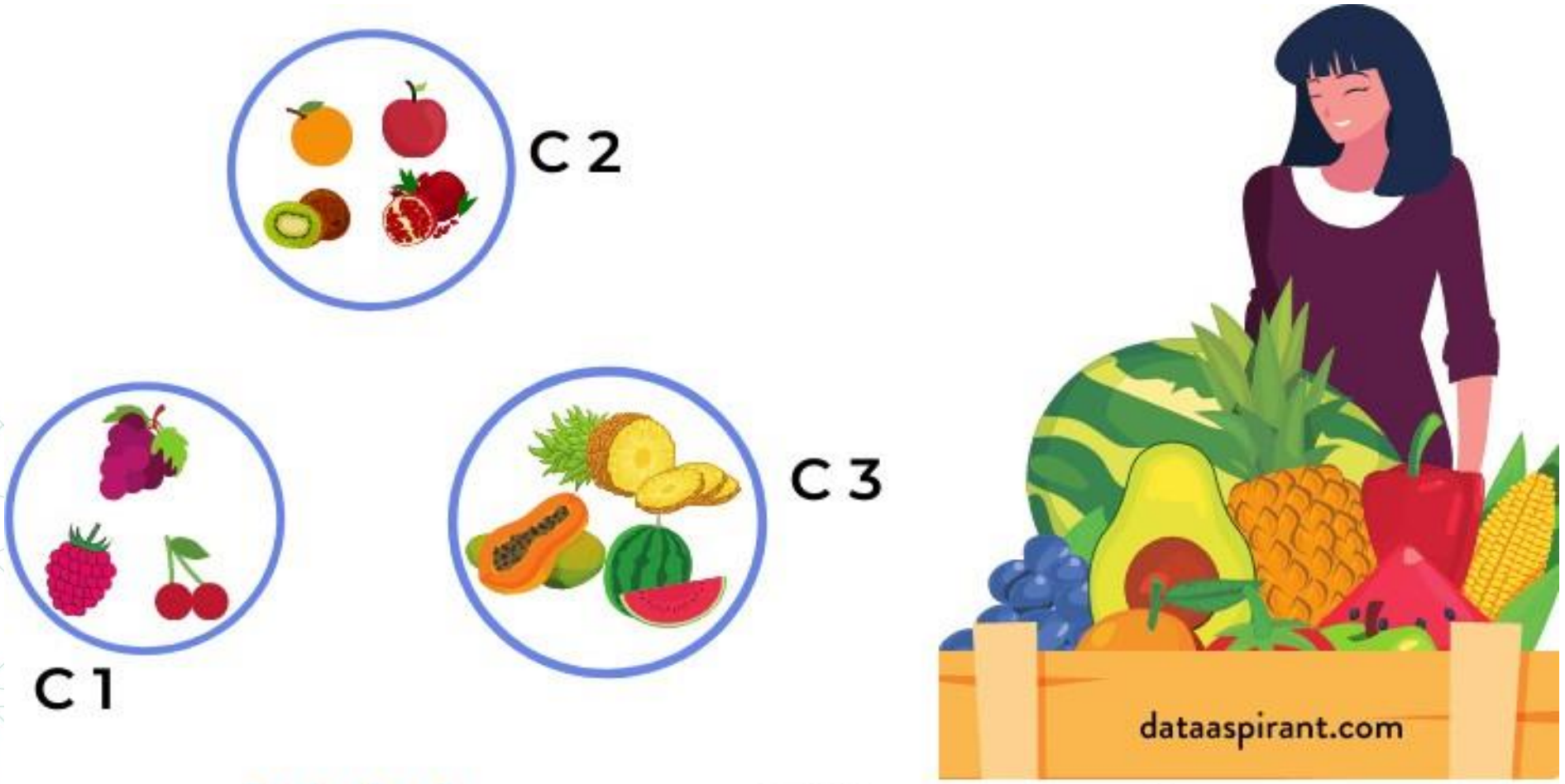
Hierarchical clustering

- **Hierarchical clustering** creates a tree of clusters.
- Hierarchical clustering, not surprisingly, is well suited to hierarchical data, such as taxonomies.
- In addition, another advantage is that any number of clusters can be chosen by cutting the tree at the right level.





K-means Clustering



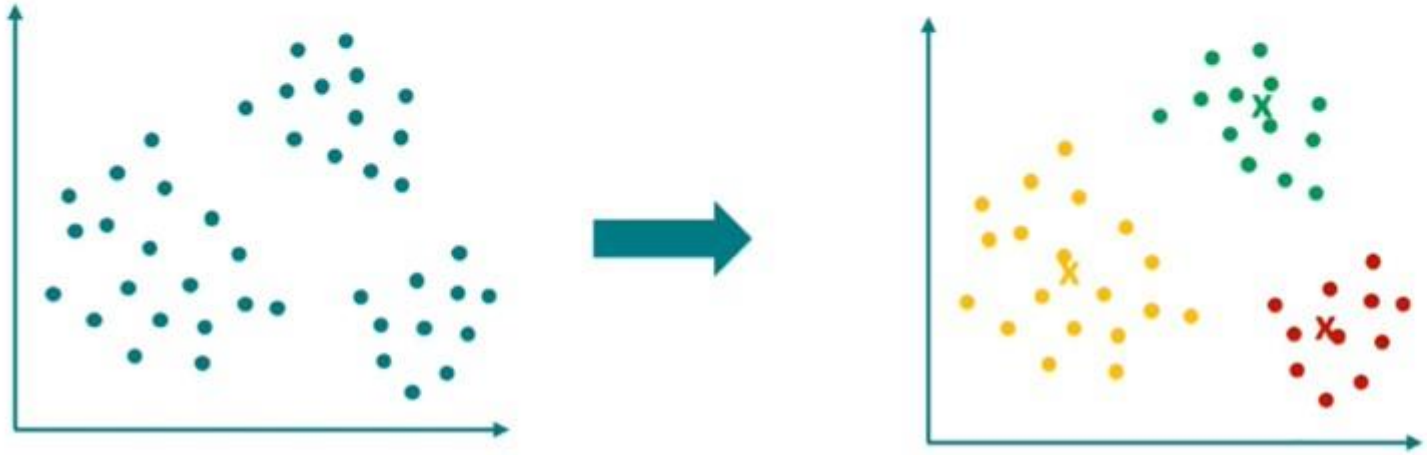
K Means Clustering

K-means

- One of the simplest and most common methods for cluster analysis.
- The k-Means method clusters your data points on a given number of clusters.

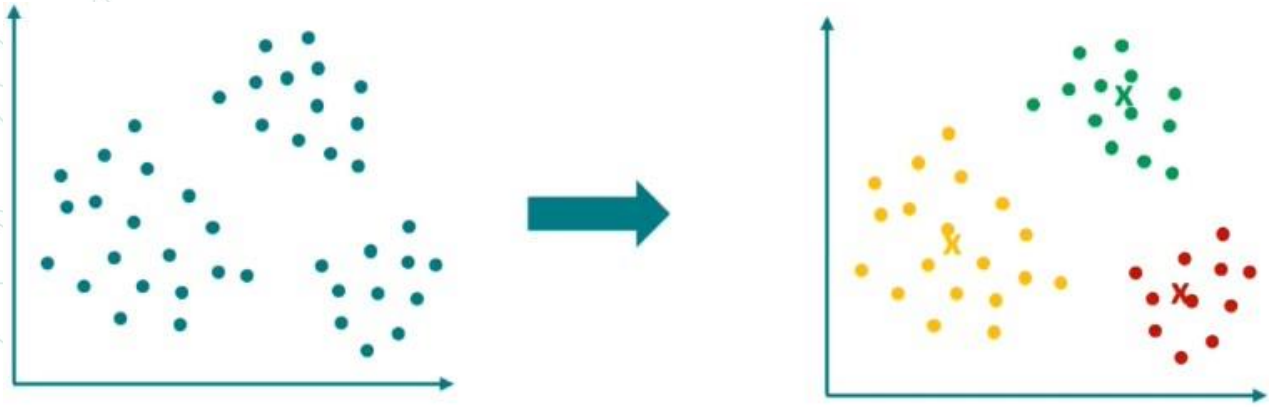
→ We have a data set and would like to divide it into clusters.

You must define the number of clusters!!!

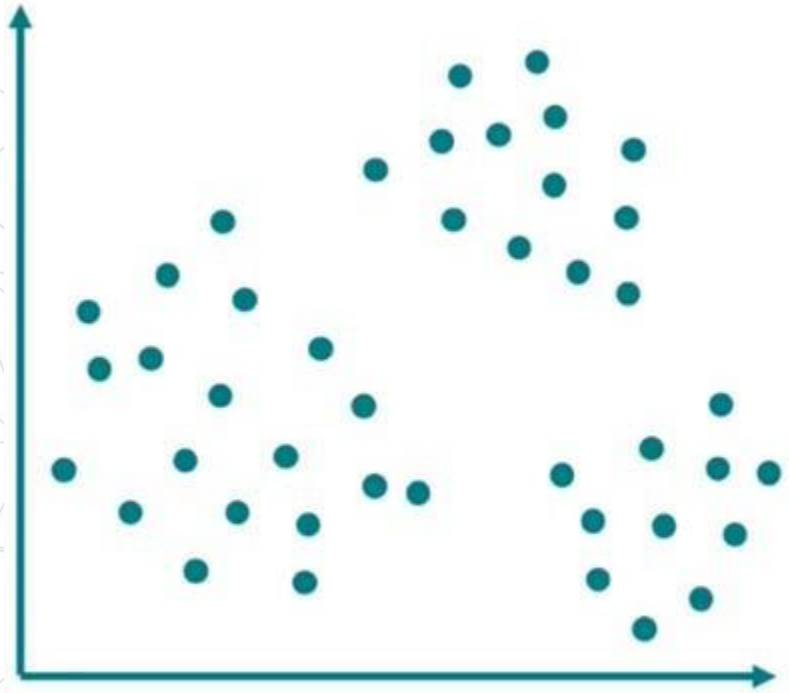


How does the K-means work?

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.
4. Compute the sum of the squared distance between data points and all centroids.
5. Assign each data point to the closest cluster (centroid).
6. Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

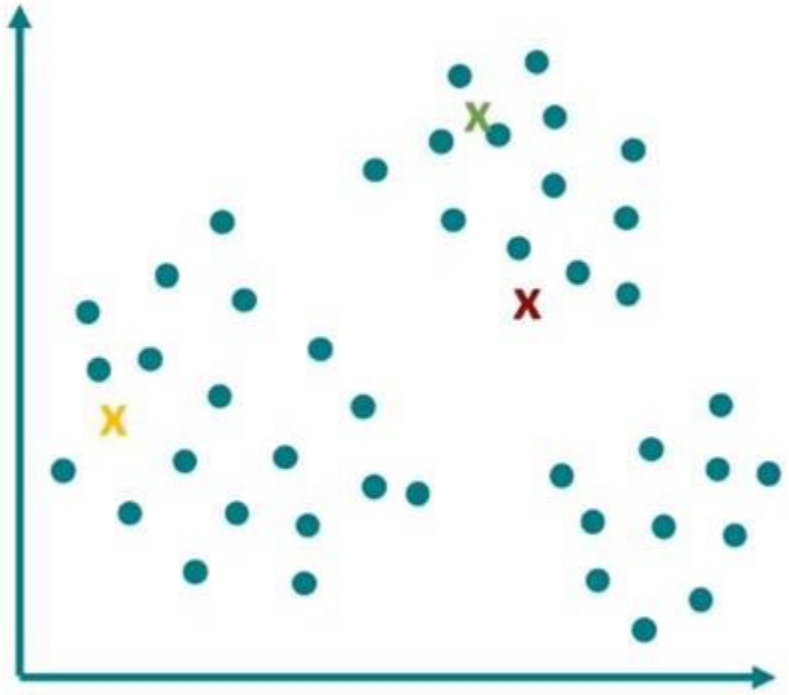


Step 1: Define number of clusters



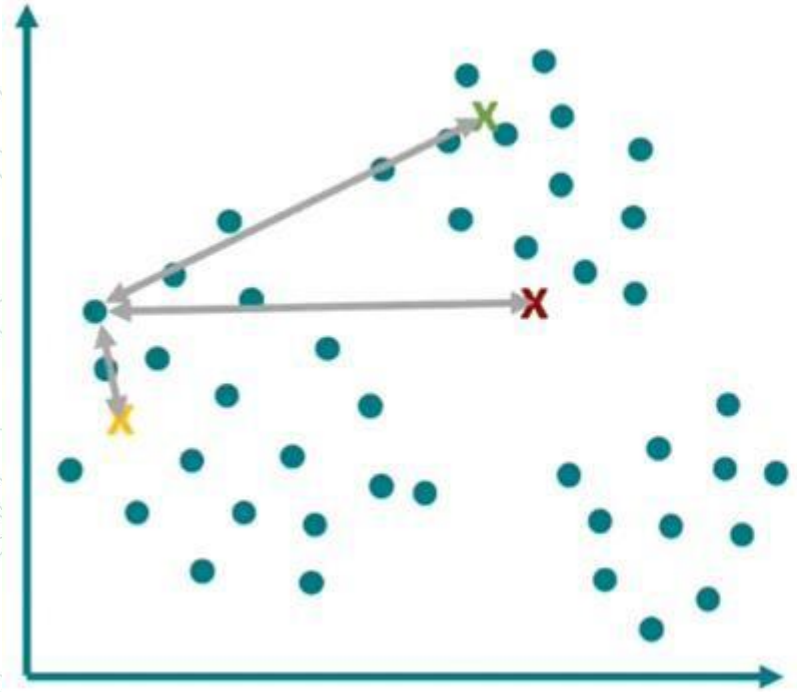
- To find the groups or clusters, the number of clusters must first be defined.
- The number of clusters is the "k" in k-means.
- In this example, k was selected equal to 3.

Step 2: Set cluster centers randomly



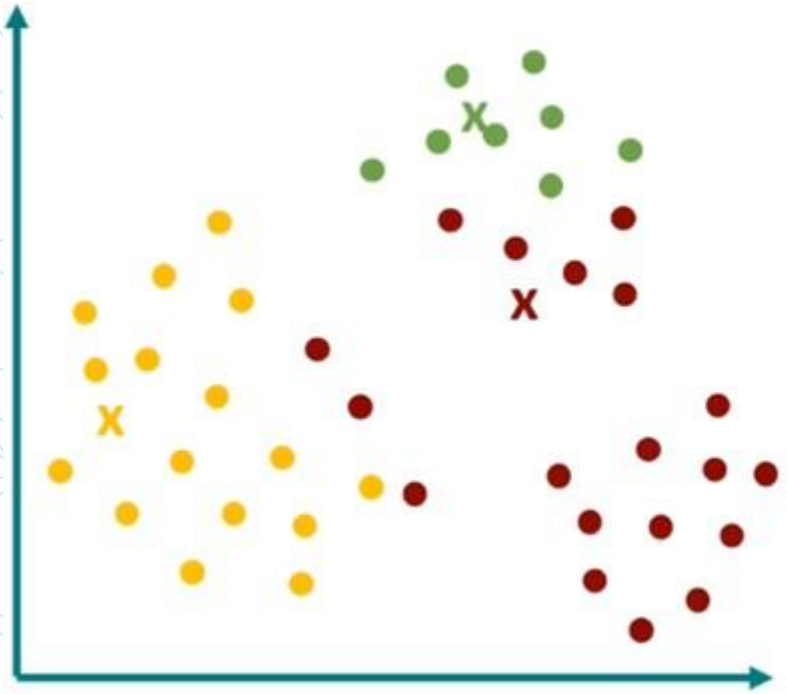
- The initial Cluster Centroids are defined.
- This usually happens randomly.
- We have selected 3 clusters, so three centroids are positioned randomly.
- Each of the centroids now represents a cluster.

Step 3: Assign points to clusters



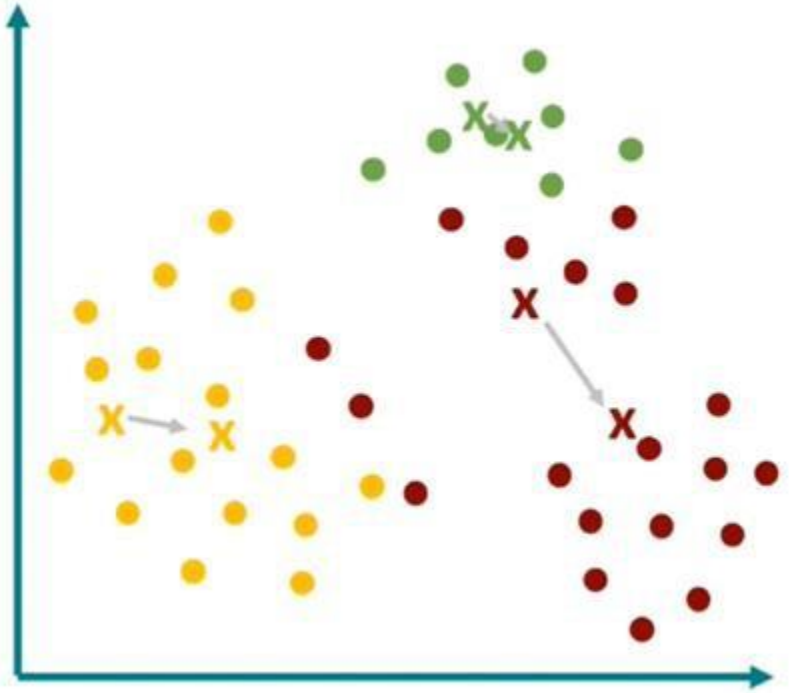
- Now the distance from the first point to each of the cluster centroids is measured.

Step 3: Assign points to clusters



- Now the distance from the first point to each of the cluster centroids is measured.
- The point is then assigned to the cluster that is closest to it.
- This is now repeated for all further points.
- Then all points are initially assigned to a cluster

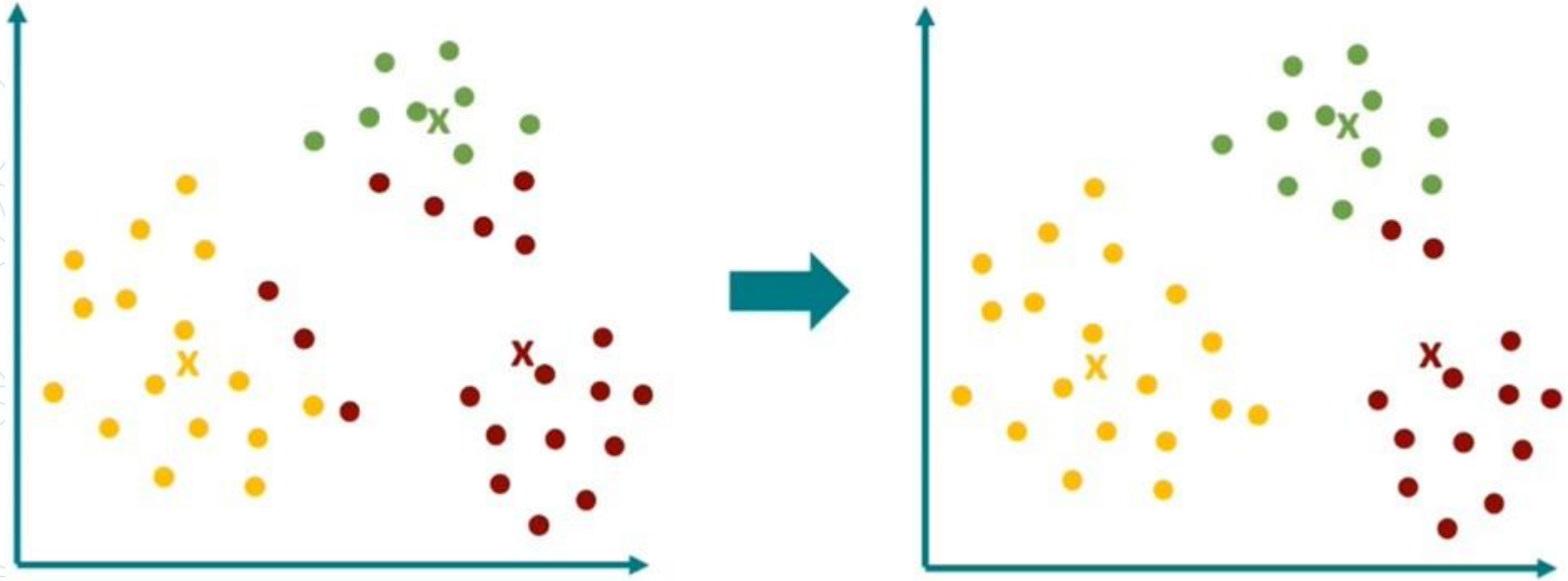
Step 4: Calculate the center of each cluster



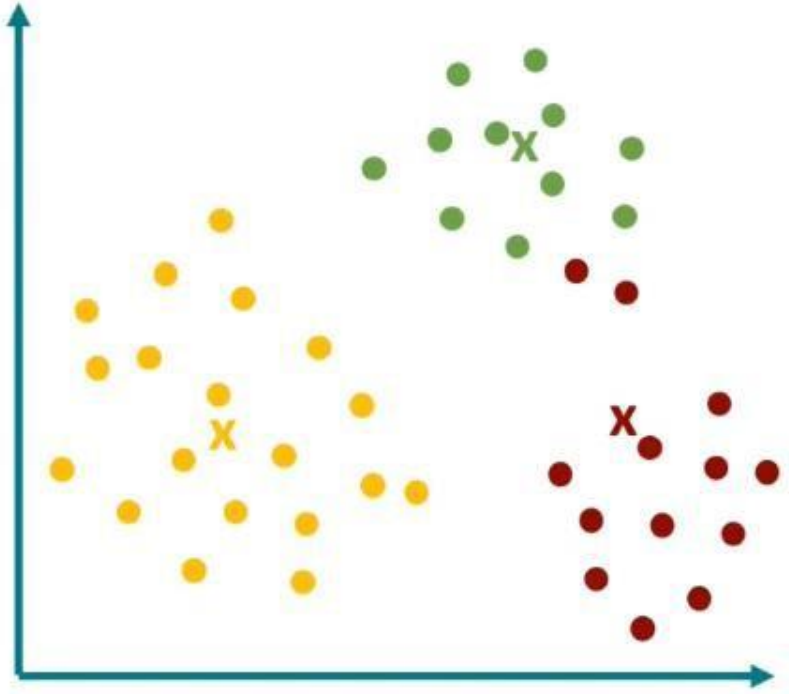
- The center of each cluster is calculated.
- These centers are the new Centroids of the clusters.
- So the cluster centroids are moved to the cluster centers.

Step 5: Assign points to the new clusters

Since the centroids can now be located at a different point, each point is again assigned the cluster that is closest to it.

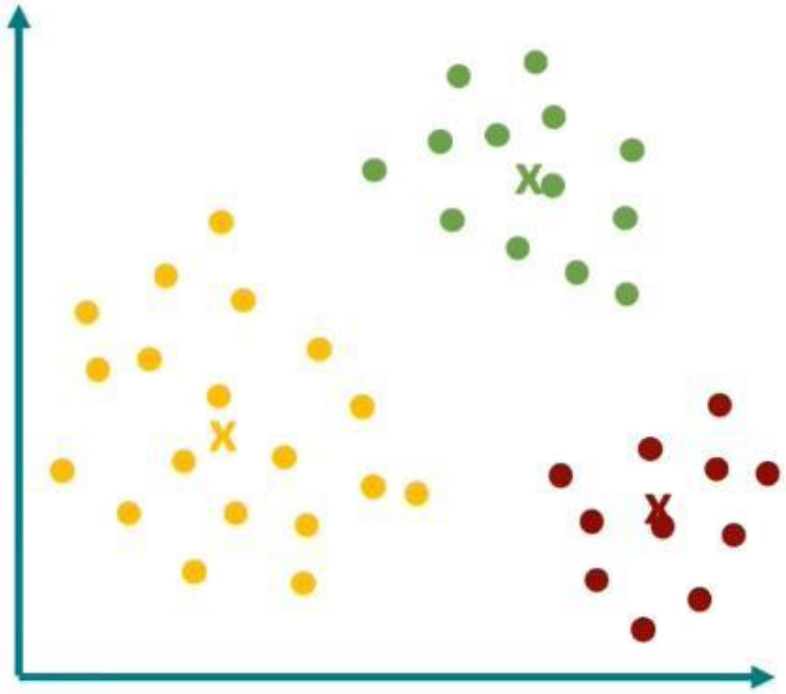


Repeat step 4 and step 5



- Now steps 4) and 5) are repeated until the cluster distribution does not change anymore.
  - Calculate the center of each cluster
  - Assign cluster centroid to the center
  - Assign points to the new clusters
- If the clusters do not change in one iteration, the procedure is over!

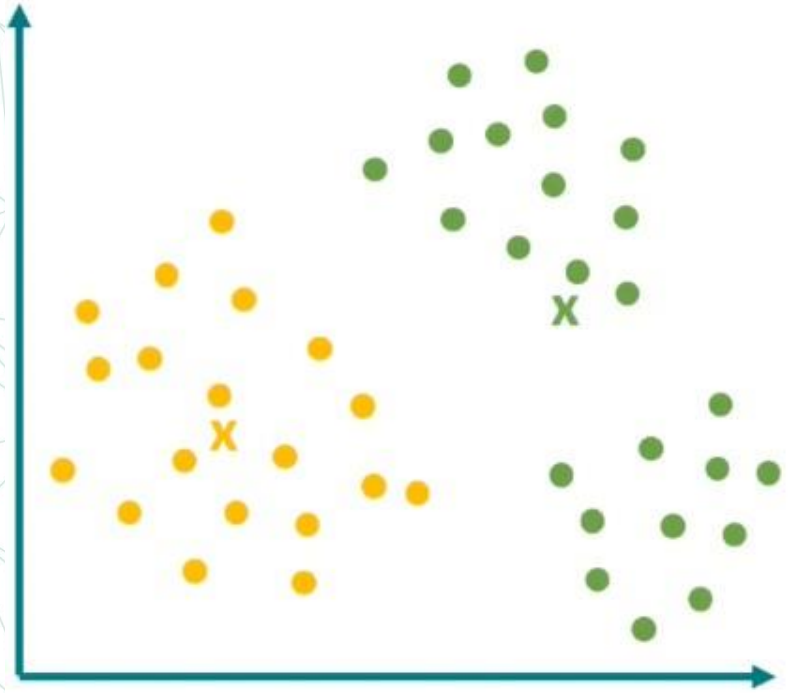
K-means



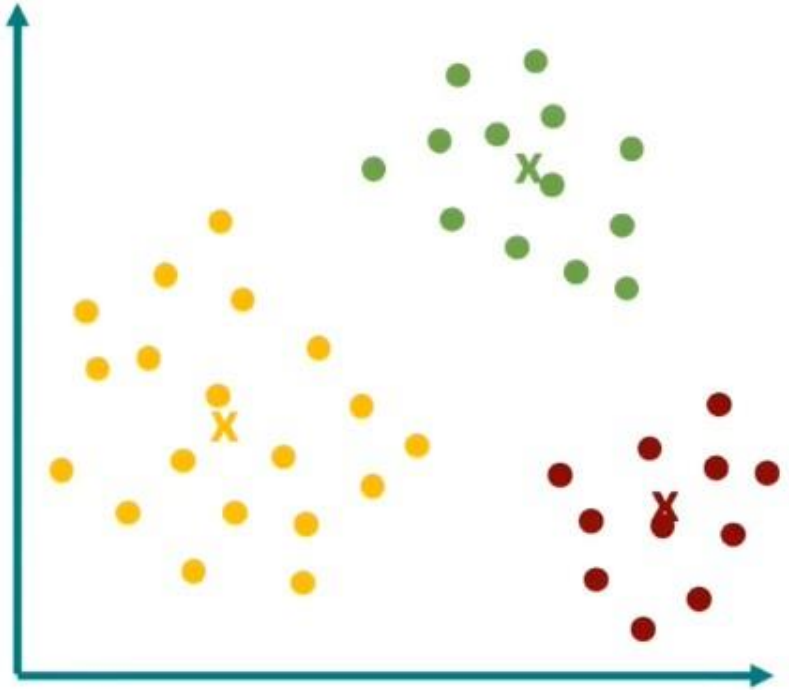
- A big disadvantage of the k-means method is that the final result depends very much on which initial cluster was used.
- To take this into account, the whole procedure is carried out several times.
- Different randomly chosen starting points are then used for each of the calculations.
- Then the cluster is used, which has the smallest sum of the distances between the cluster center and the points.

Optimal cluster number

2 Cluster



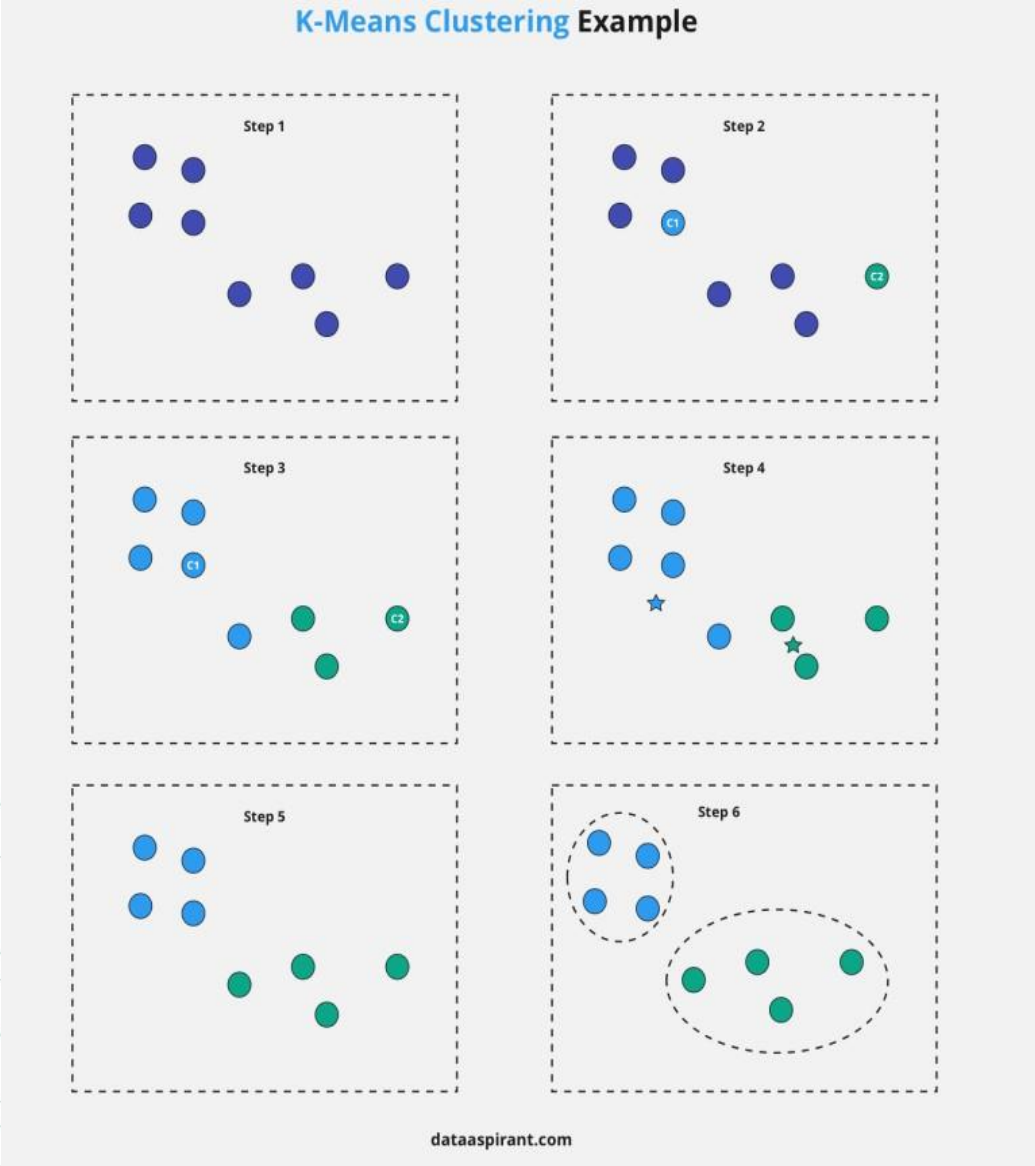
3 Cluster



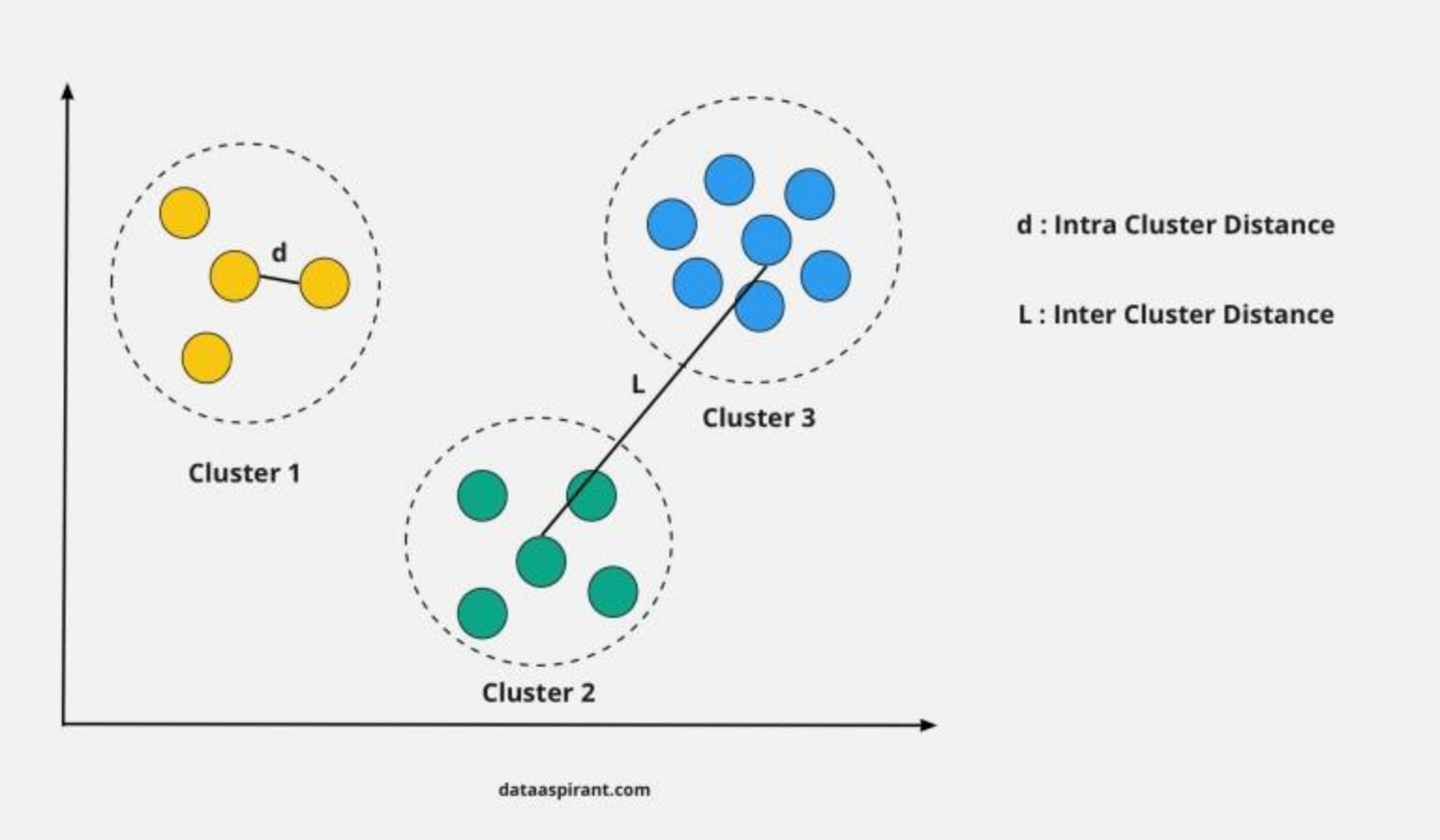
- With each new cluster the summed distance in the clusters becomes smaller and smaller.
- If there are as many clusters as points, zero comes out
- How many clusters should be used?

How to identify the best “K” ?

- Elbow method: It is to calculate the sum the distances from data points to centroids and aims at minimising the sum to an optimal value.
- Silhouette analysis: it measures how similar a point is to its own cluster compared to other clusters.

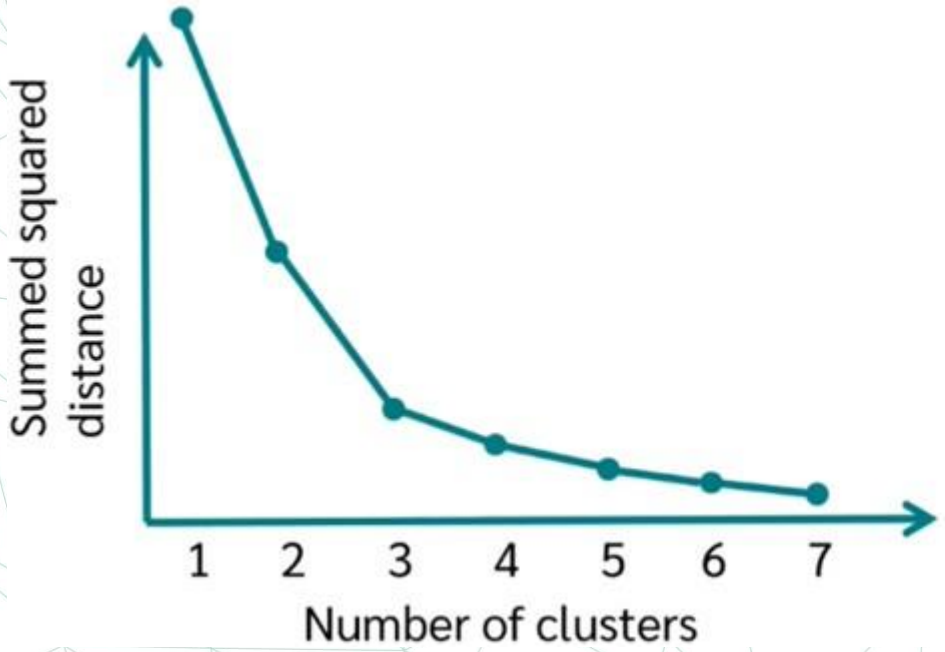

K-Means Clustering Example
Step 1
Step 2
C1
C2
Step 3
C1
C2
Step 4
Step 5
Step 6
dataaspirant.com

How To Evaluate Clusters



Elbow Method

Elbow method gives us an idea on what a good *k* number of clusters would be based on the sum of squared distance (SSE) between data points and their assigned clusters' centroids.

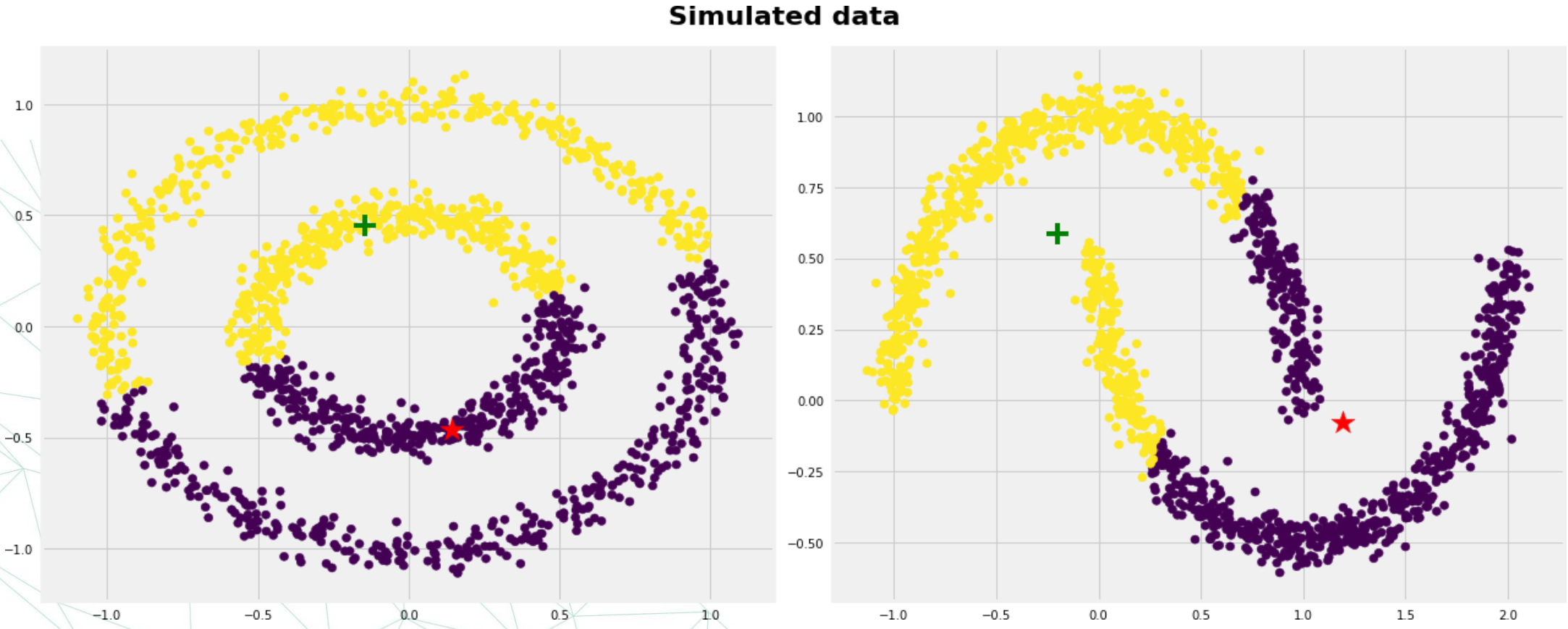


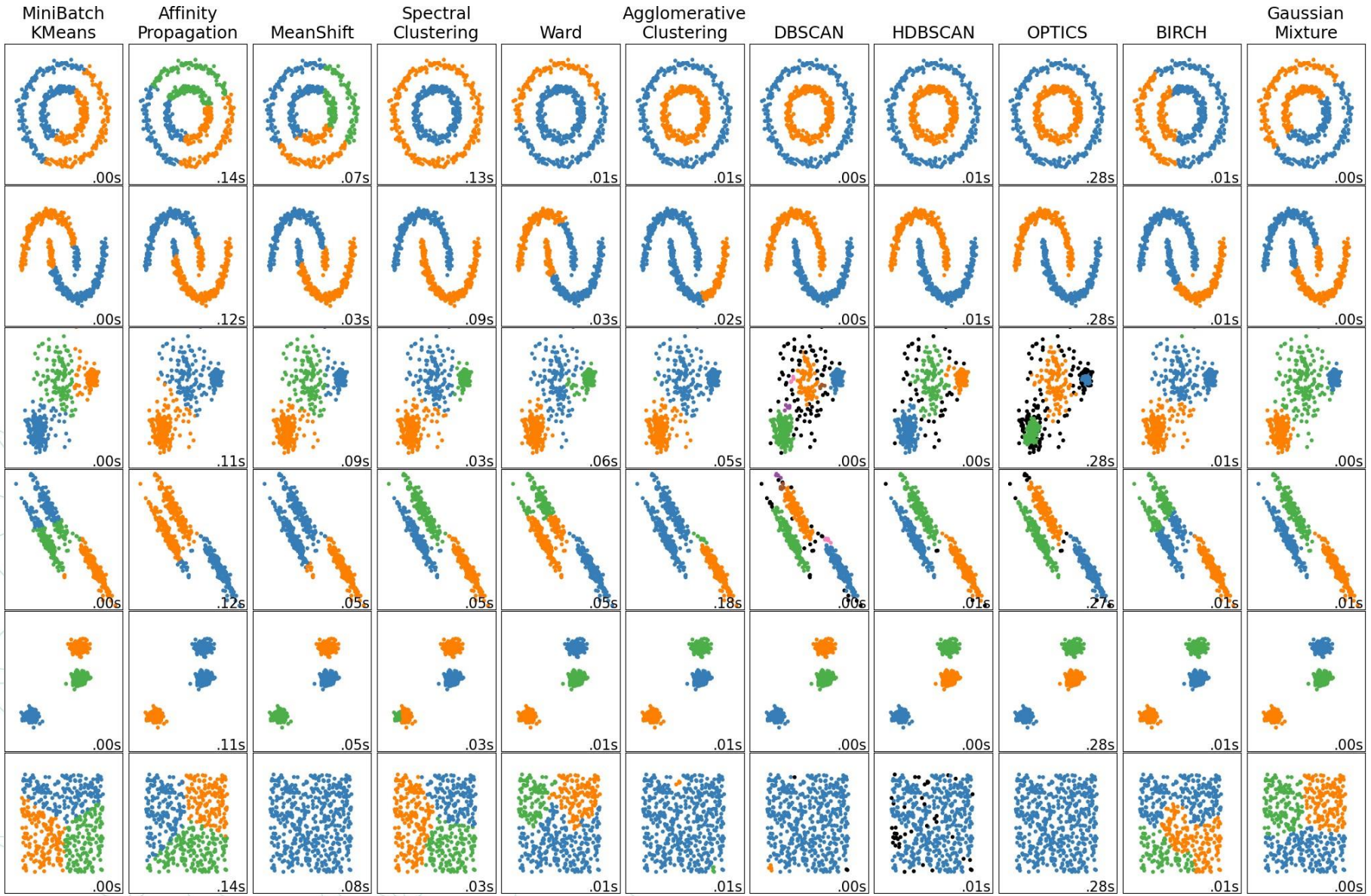
- With each additional cluster the summed distance between the points and the cluster center becomes smaller and smaller
- However, there is a cluster number from which each additional cluster reduces the summed distance only slightly.
- This point is used as number for the clusters.

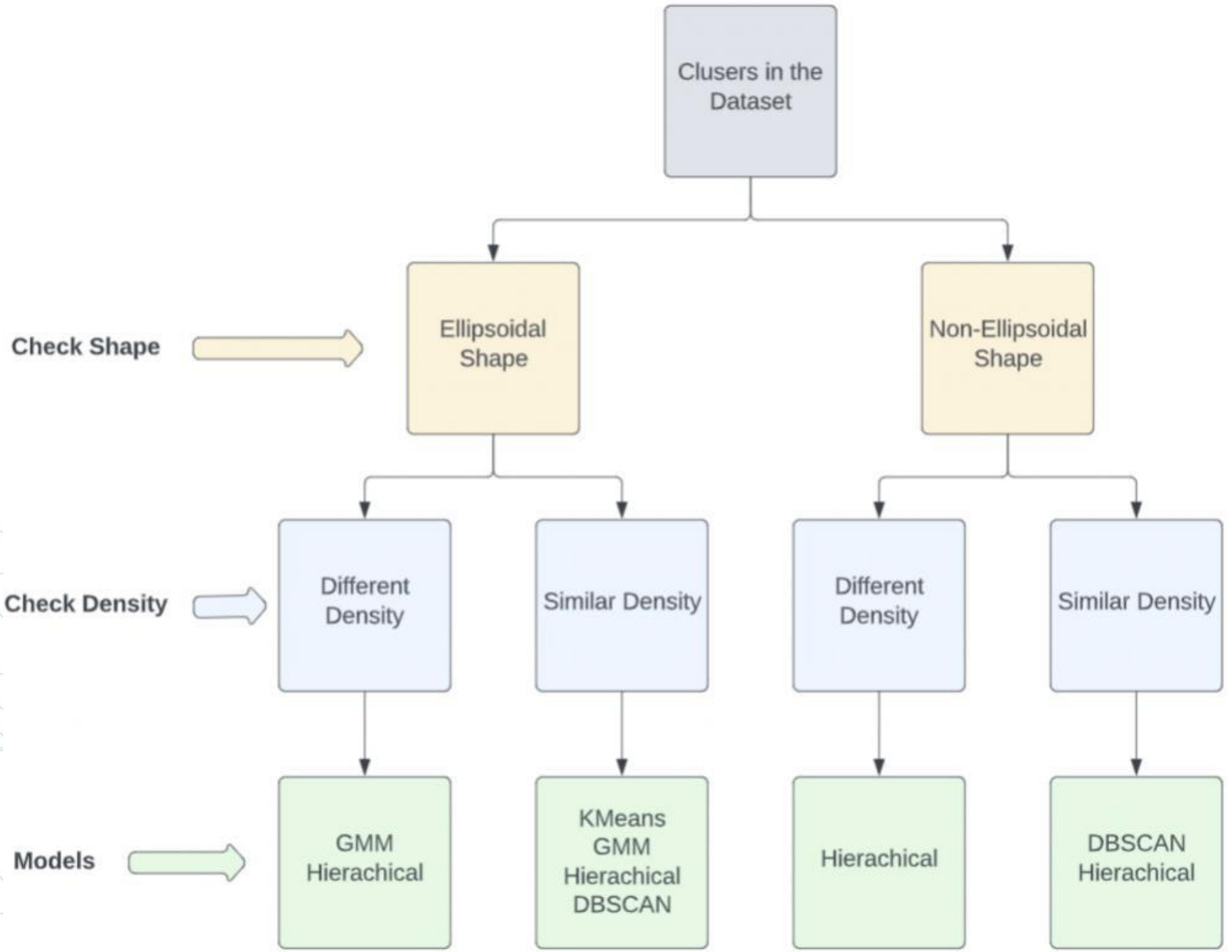
Evaluation Methods

- **Inertia:** it does calculates the sum of distances of all the entities present in the cluster.

Drawbacks


Simulated data
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
HDBSCAN
OPTICS
BIRCH
Gaussian Mixture
.00s .14s .07s .13s .01s .01s .00s .01s .28s .01s .00s
.00s .12s .03s .09s .03s .02s .00s .01s .28s .01s .00s
.00s .11s .09s .03s .06s .05s .00s .00s .28s .01s .00s
.00s .12s .05s .05s .05s .18s .00s .01s .27s .01s .01s
.00s .11s .05s .03s .01s .01s .00s .00s .28s .01s .00s
.00s .14s .08s .03s .01s .01s .00s .01s .28s .01s .00s

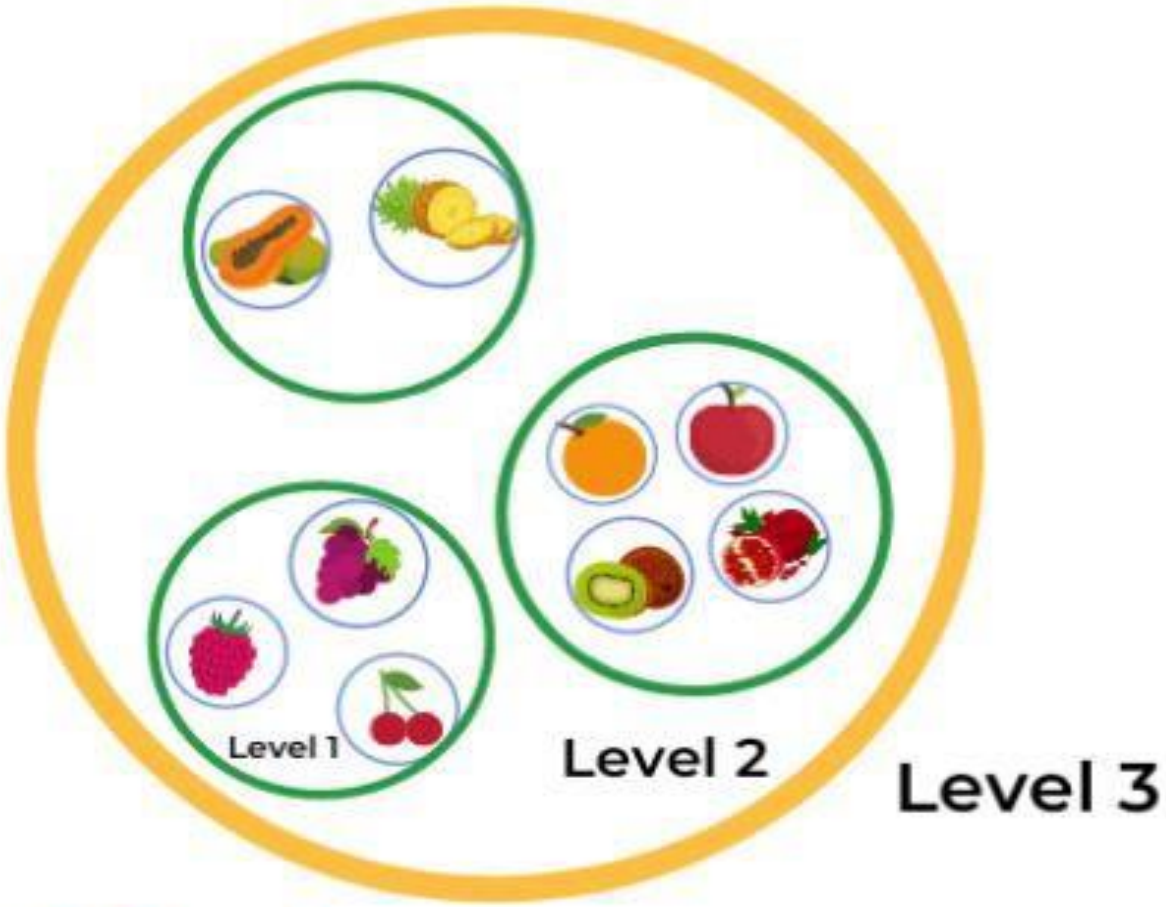

Clusers in the Dataset
Check Shape
Ellipsoidal Shape
Non-Ellipsoidal Shape
Check Density
Different Density
Similar Density
Different Density
Similar Density
Models
GMM
Hierachical
KMeans
GMM
Hierachical
DBSCAN
Hierachical
DBSCAN
Hierachical

Let's Code

Q&A

Questions and answers

Hierarchical Clustering



Hierarchical Clustering

dataaspirant.com

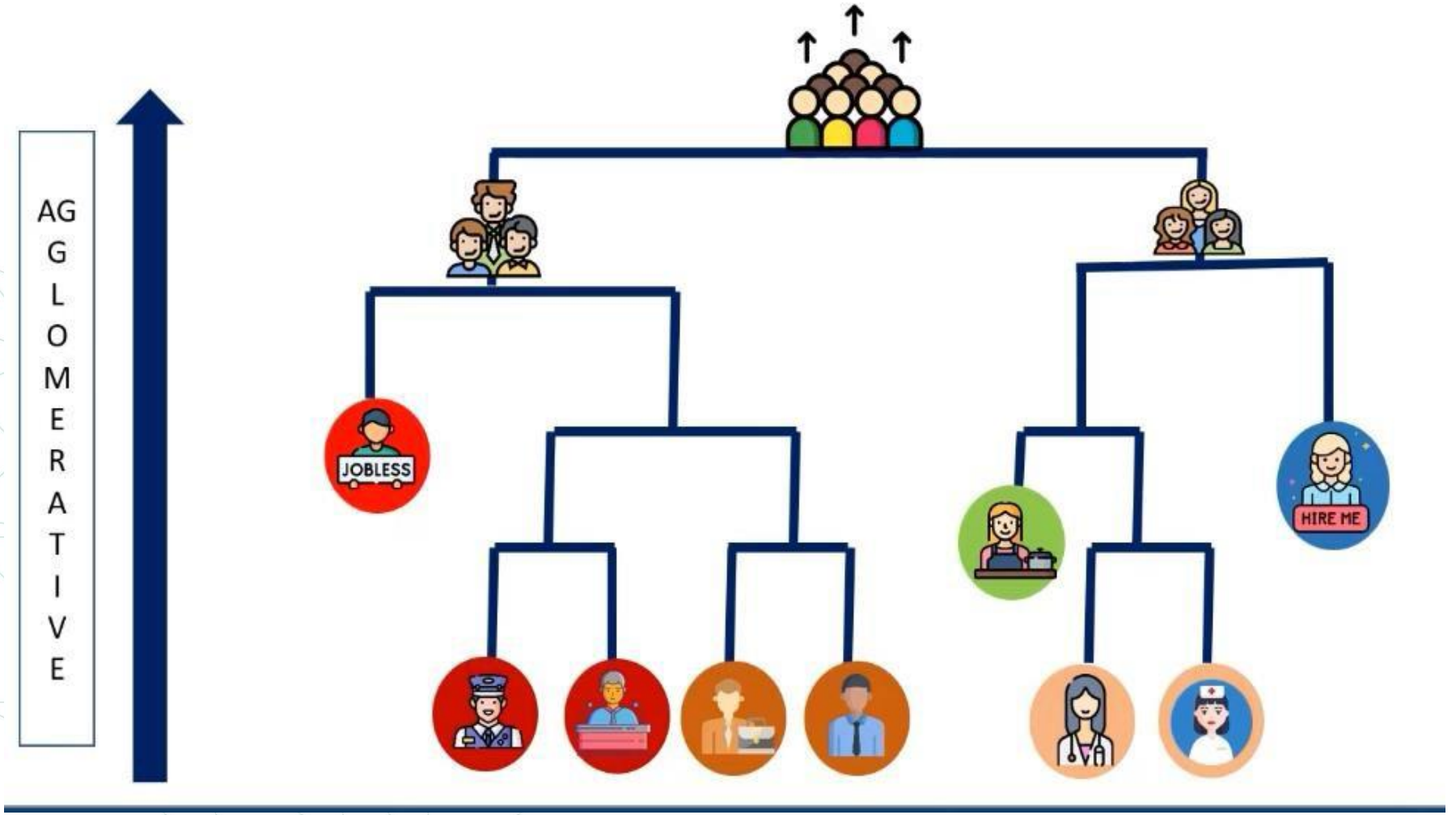
Why Hierarchical Clustering

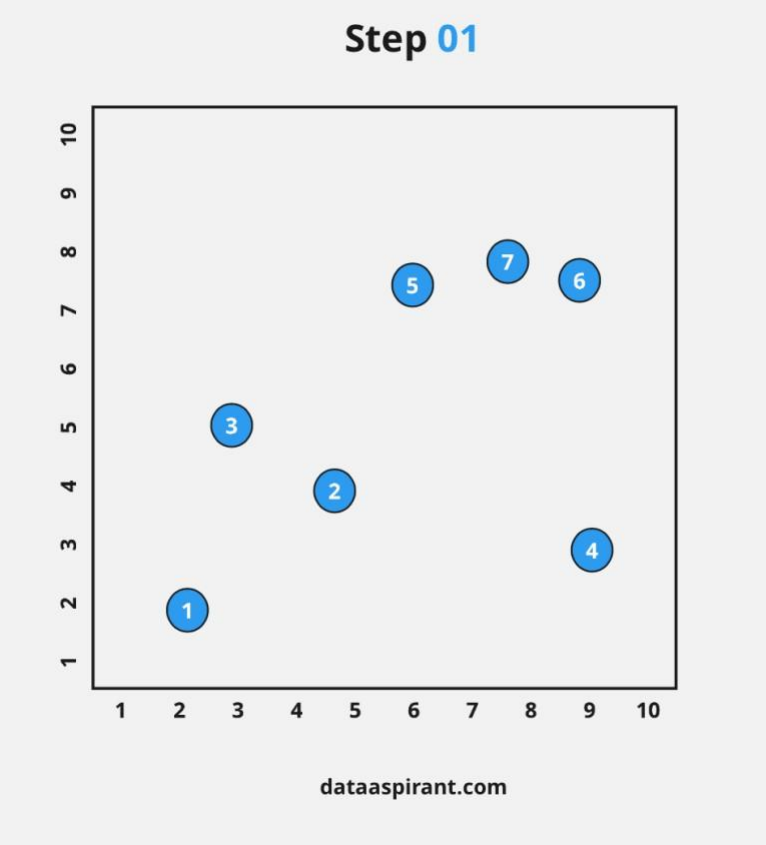
- As we have already seen in the **K-Means Clustering algorithm**, it uses a pre-specified number of clusters. It requires advanced knowledge of **K**., i.e., how to define the number of clusters one wants to divide your data.
- Still, in hierarchical clustering no need to pre-specify the number of clusters as we did in the K-Means Clustering; one can stop at any number of clusters.
- Furthermore, Hierarchical Clustering has an advantage over K-Means Clustering. i.e., it results in an attractive tree-based representation of the observations, called a **Dendrogram**.

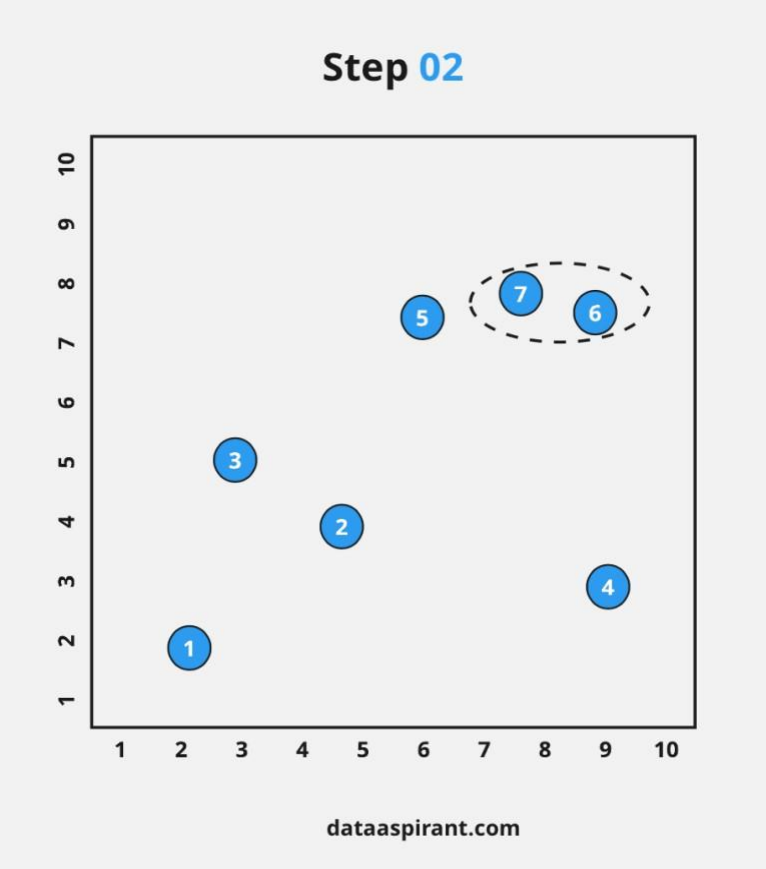
Hierarchical Clustering Types

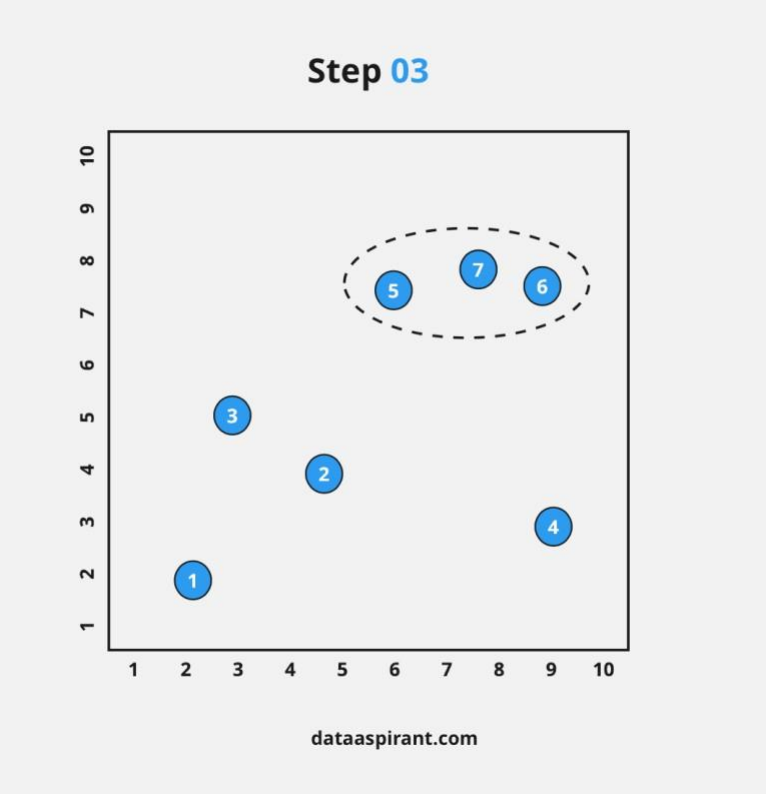
- **Agglomerative Hierarchical Clustering**
  - Start with points as individual clusters.
  - At each step, it merges the closest pair of clusters until only one cluster (or K clusters left).
- **Divisive Hierarchical Clustering**
  - Start with one, all-inclusive cluster.
  - At each step, it splits a cluster until each cluster contains a point (or there are clusters).

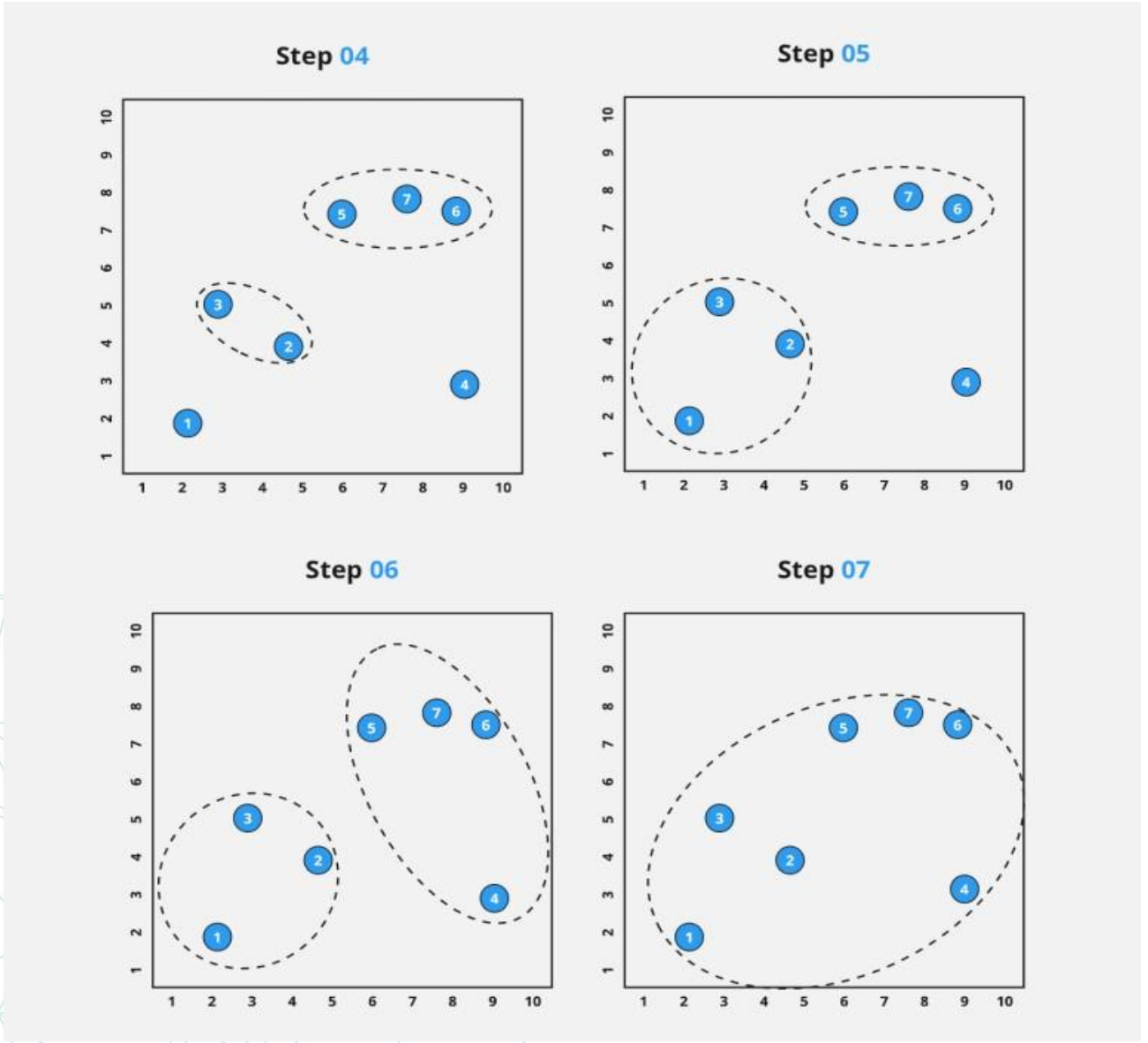
What is Agglomerative Clustering?

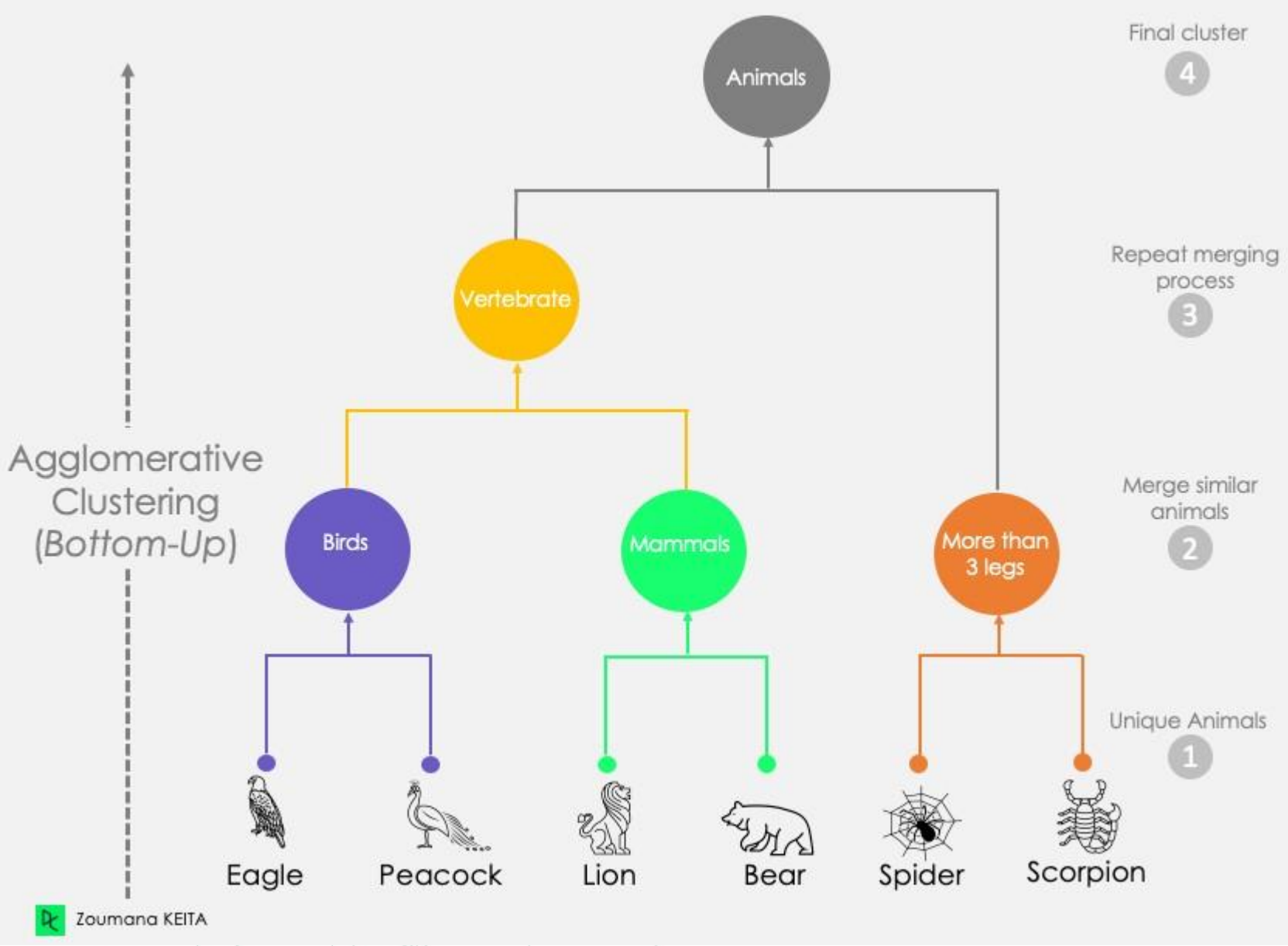



Step 01
dataaspirant.com

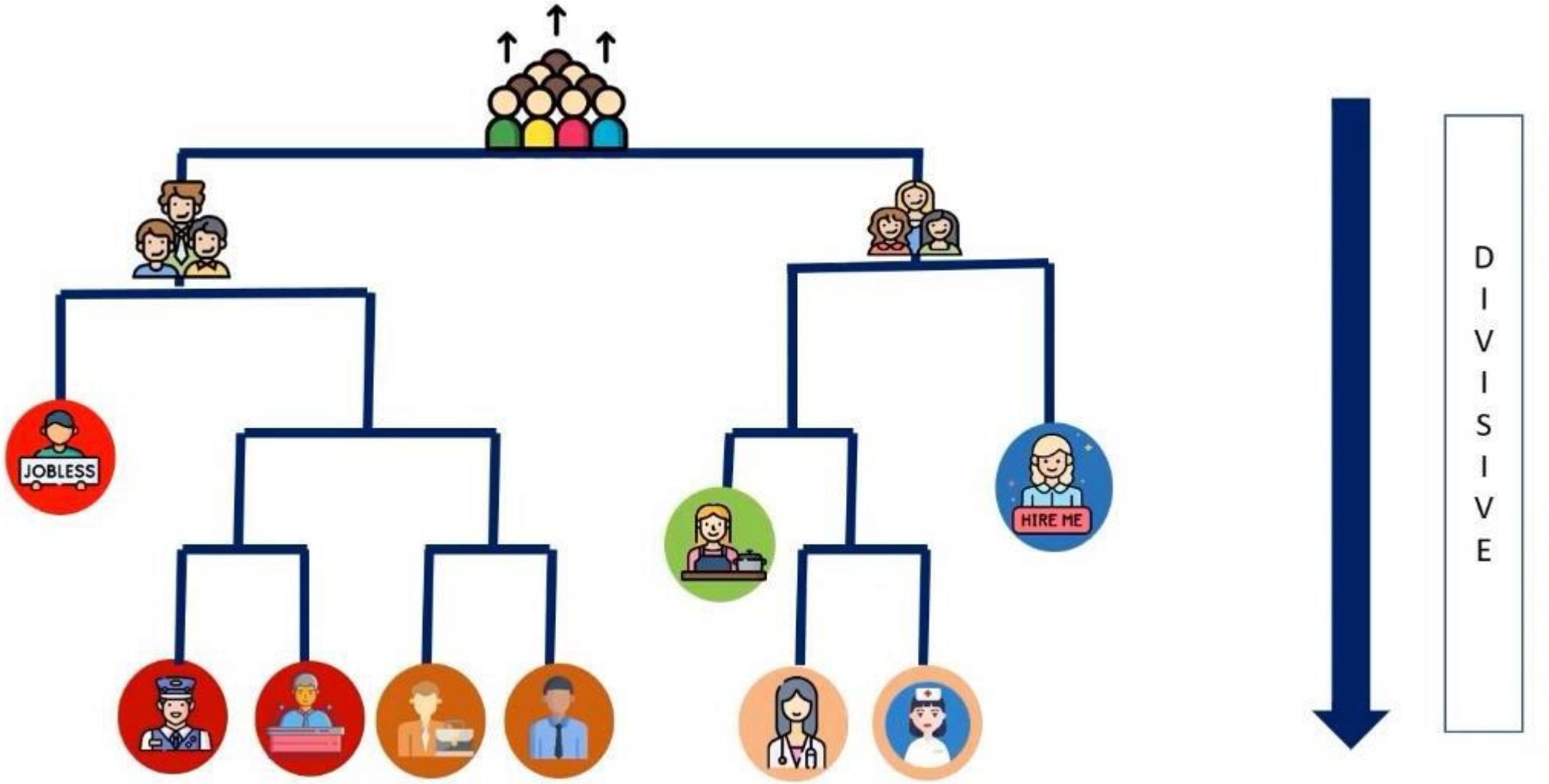

Step 02
dataaspirant.com


Step 03
dataaspirant.com


Step 04
Step 05
Step 06
Step 07


Final cluster
4
Animals
Repeat merging process
3
Vertebrate
Agglomerative Clustering (Bottom-Up)
Merge similar animals
2
Birds
Mammals
More than 3 legs
Unique Animals
1
Eagle
Peacock
Lion
Bear
Spider
Scorpion
Zoumana KEITA

What is Divisive Clustering?



Difference ways to measure
the distance between two clusters

- **Single Linkage**

  $D(c_1,c_2) = \min D(x_1,x_2)$

  Minimum distance or distance between closest elements in clusters

- **Complete Linkage**

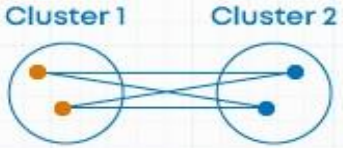
  $D(c_1,c_2) = \max D(x_1,x_2)$

  Maximum distance between elements in clusters

- **Average Linkage**

  $D(c_1,c_2) = \frac{1}{|c_1|}\frac{1}{|c_2|}\Sigma\Sigma D(x_1,x_2)$

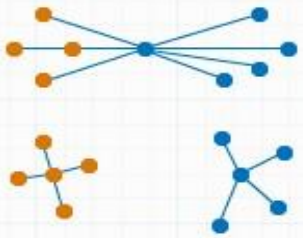
  Average of the distances of all pairs



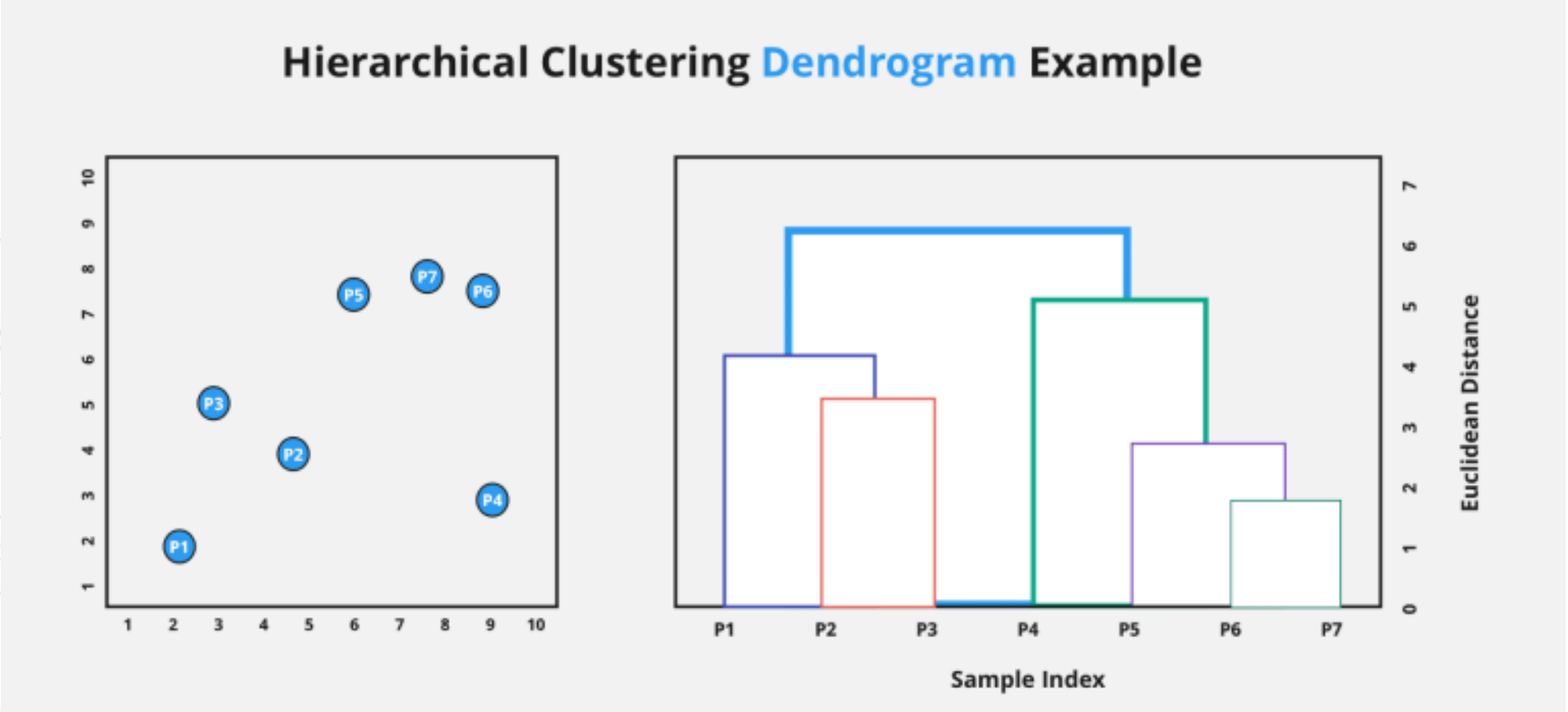
- **Centroid Method**

  Combining clusters with minimum distance between the centroids of the two clusters

- **Ward's Method**
  - Combining clusters where increase in within cluster variance is to the smallest degree.
  - Objective is to minimize the total within cluster vairance



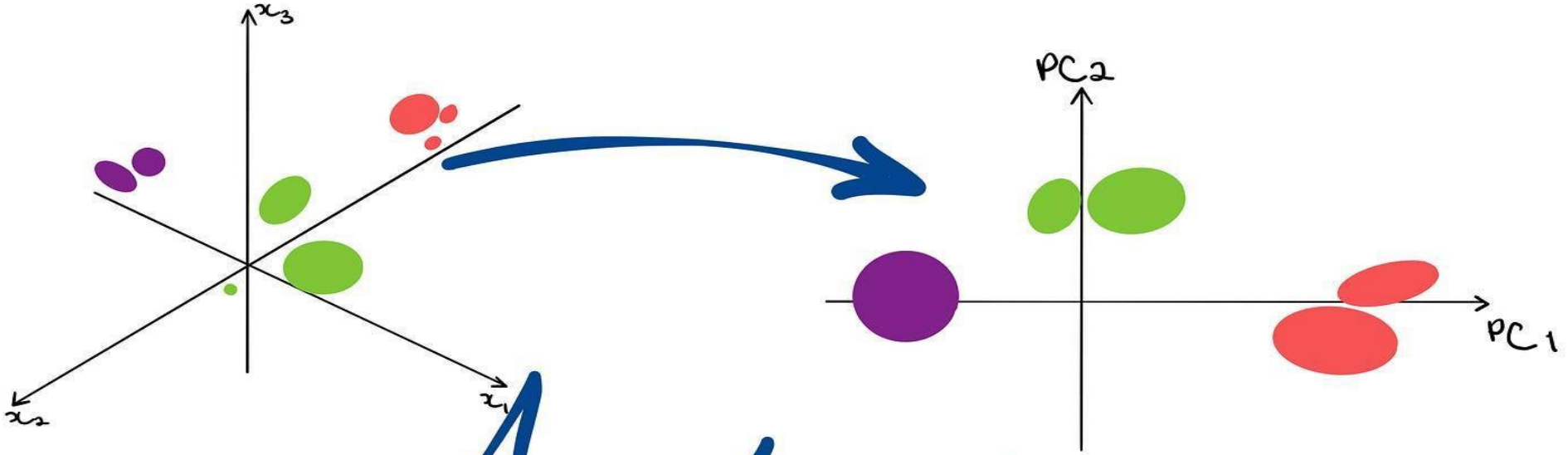
What is Dendrogram



Machine Learning

Principle Component Analysis PCA

Principal Component Analysis



Dimensionality reduction

- Dimensionality reduction offers several other benefits, such as:
  - Removing noise and redundant features.
  - Enhancing the **model's accuracy and performance**.
  - Enabling the use of algorithms that are unsuitable for higher dimensions.
  - Reducing the required storage space (fewer data requires less storage).
  - Compressing data decreases computation time and facilitates faster training.
  - Reducing the number of input variables makes the model less complex and reduces the risk of overfitting.

Dimensionality Reduction Approaches

- **Feature Selection:**
  - In **feature selection**, we select a subset of the original features that are most relevant to the prediction task. This approach involves ranking the features based on their importance and then selecting the top-ranked features.
  - Feature selection can be achieved using **statistical methods**, such as correlation analysis, or using machine learning methods, such as **decision trees**.
- **Feature Extraction:**
  - In feature extraction, we **transform the original features** into a new set of features that are a **linear combination** of the original features. The new set of features, also known as latent variables, captures the essential information from the original features. **Principal Component Analysis** (PCA) and **Linear Discriminant Analysis** (LDA) are two popular feature extraction techniques.

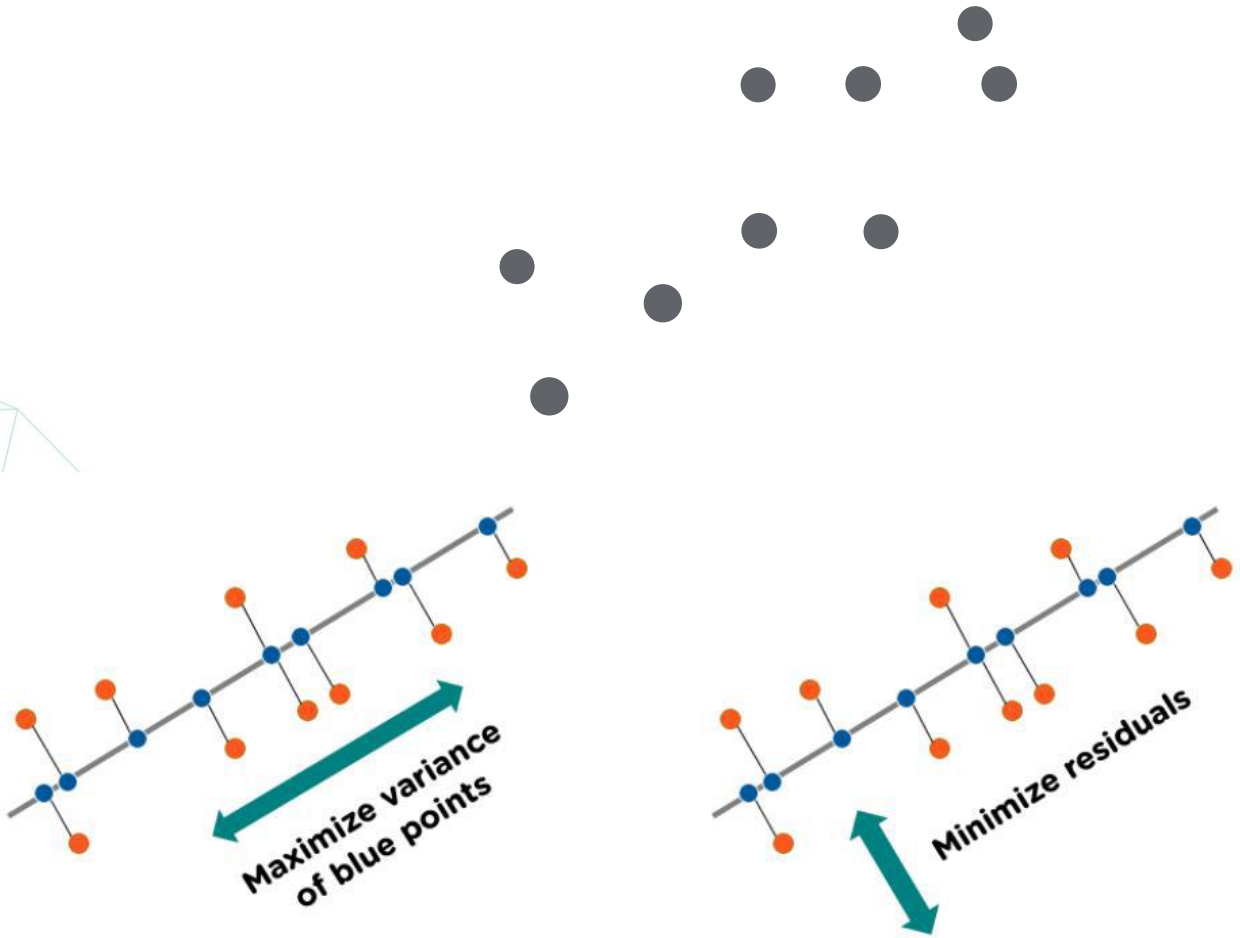
رواد مصر الرقمية

وزارة الاتصـــــالات
وتكنولوجيا المعلومات

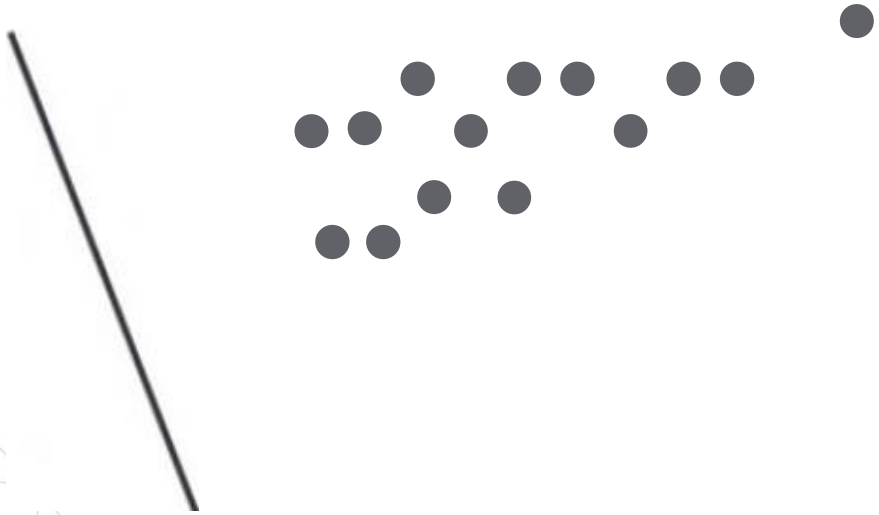
Taking a picture

https://www.youtube.com/watch?v=g-Hb26agBFg

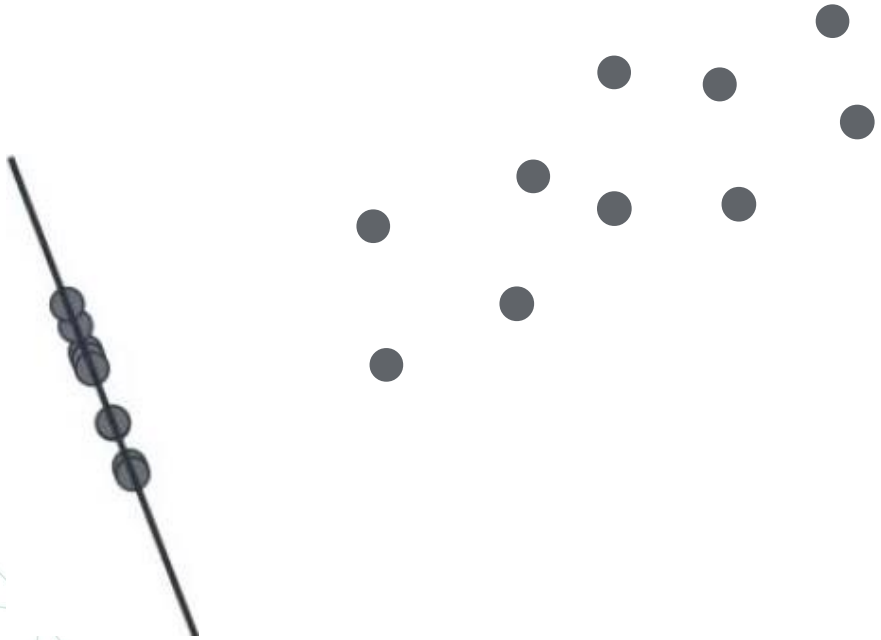
Dimensiona ity Reduction



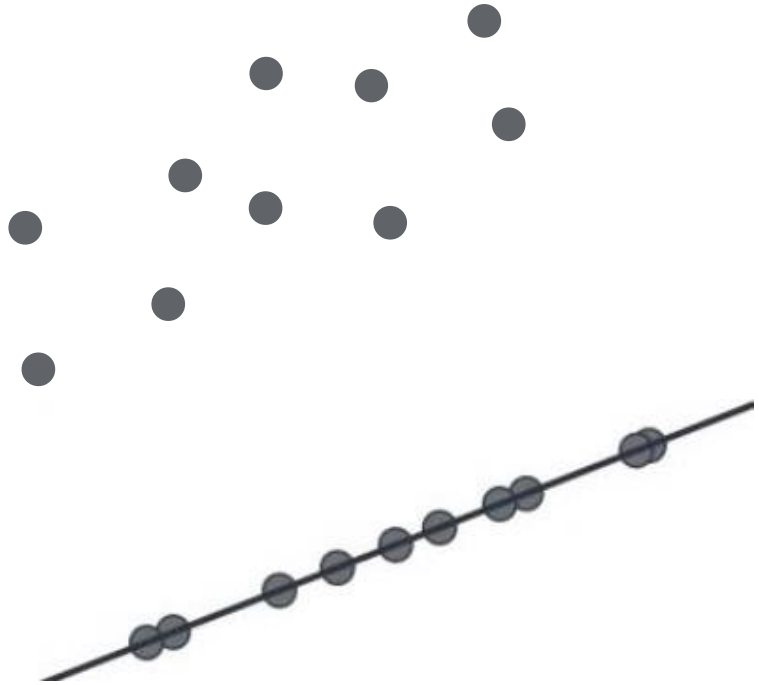
Dimensionality Reduction



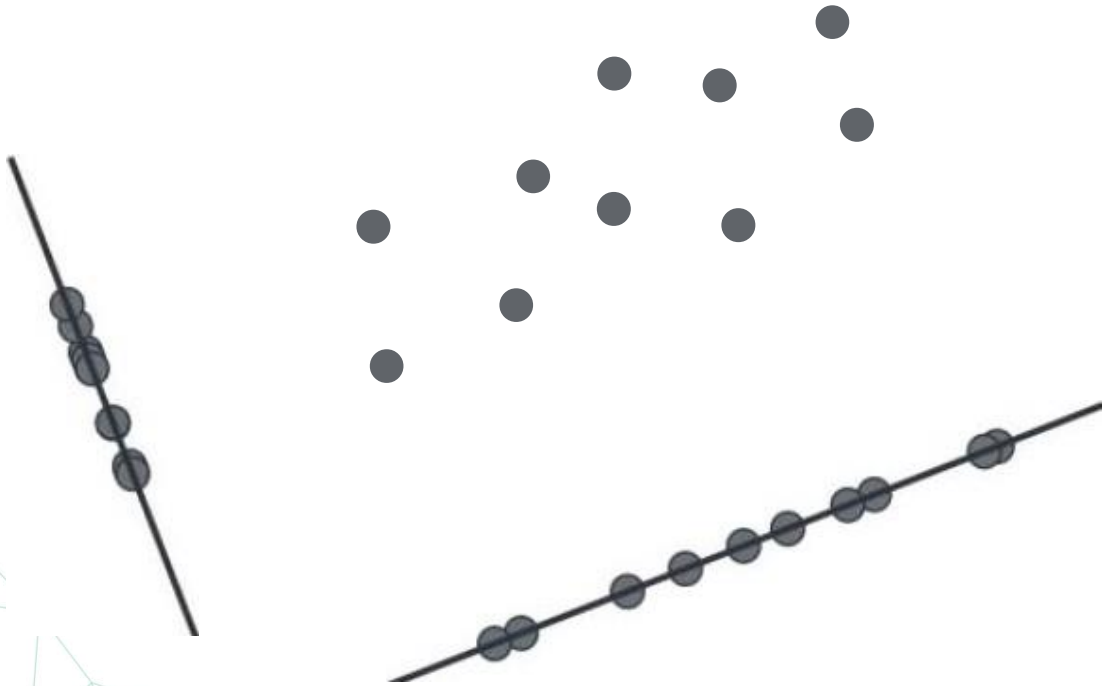
Dimensionality Reduction



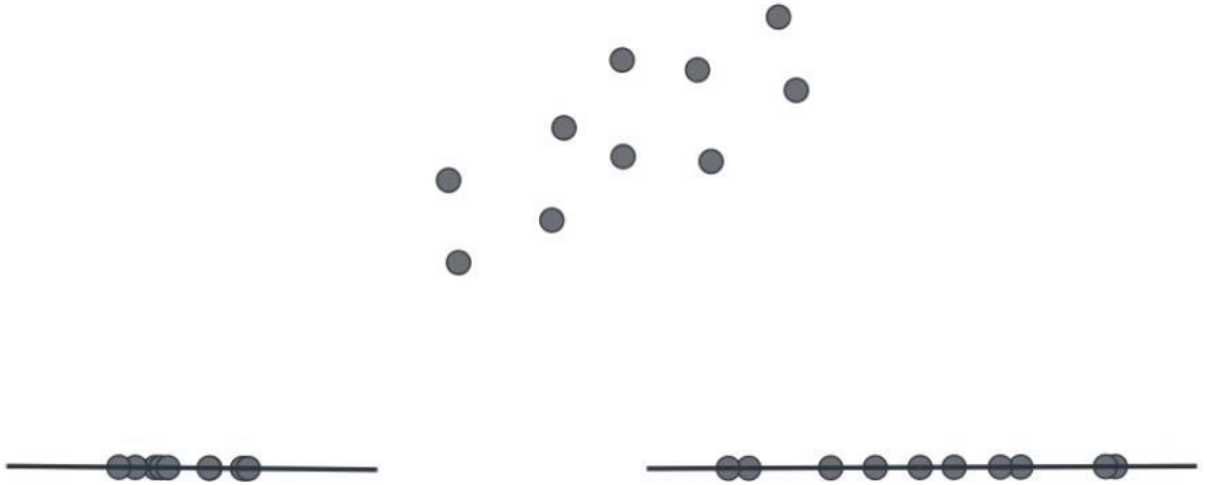
Dimensionality Reduction



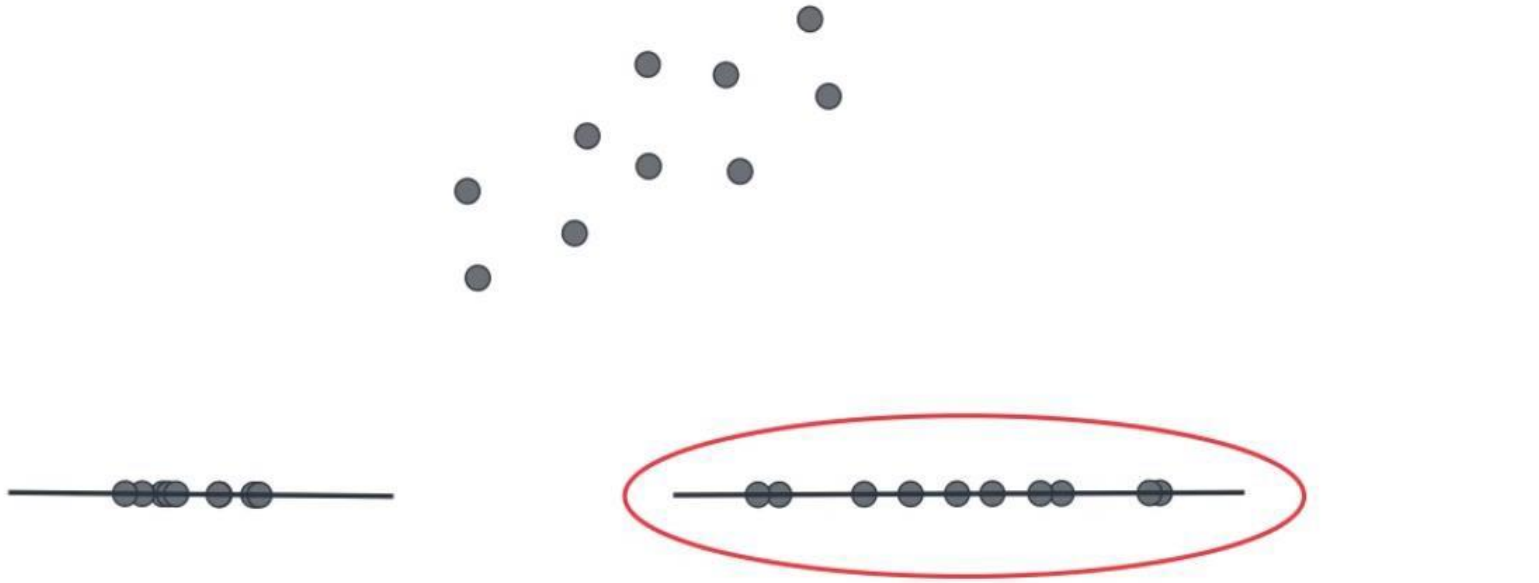
Dimensionality Reduction



Dimensionality Reduction

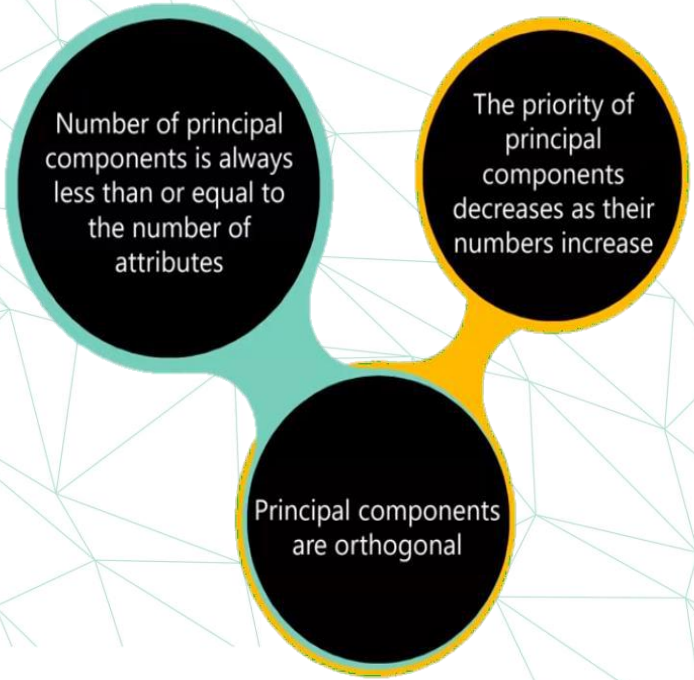


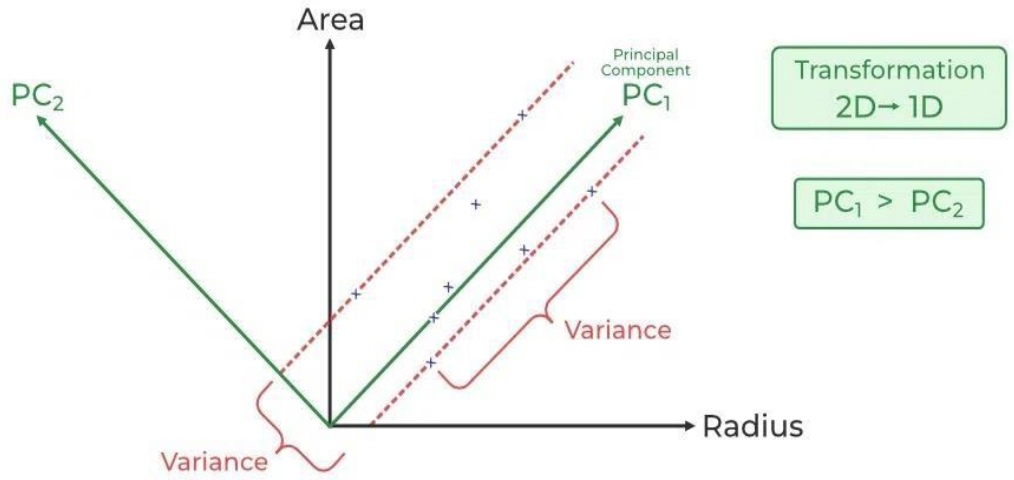
Dimensionality Reduction

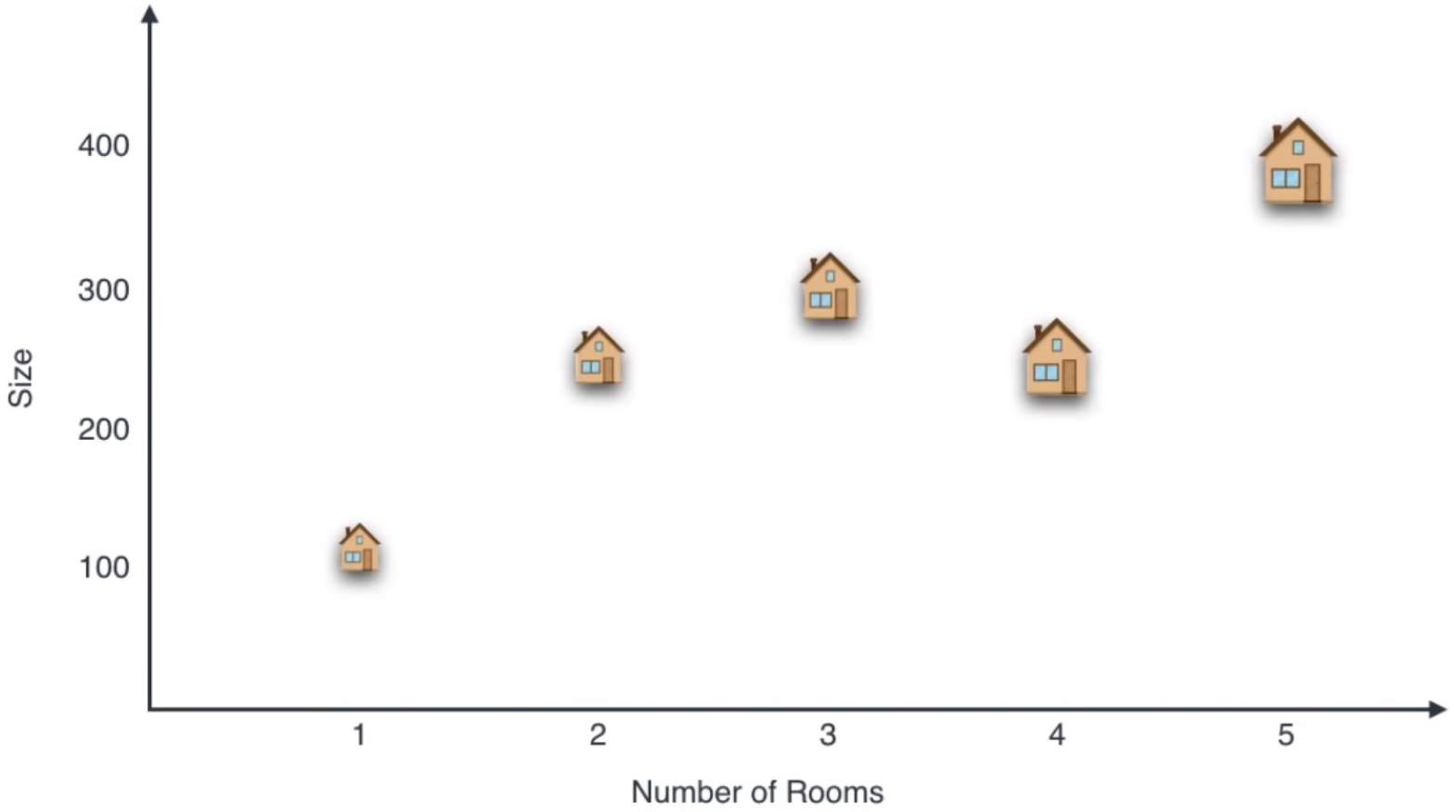


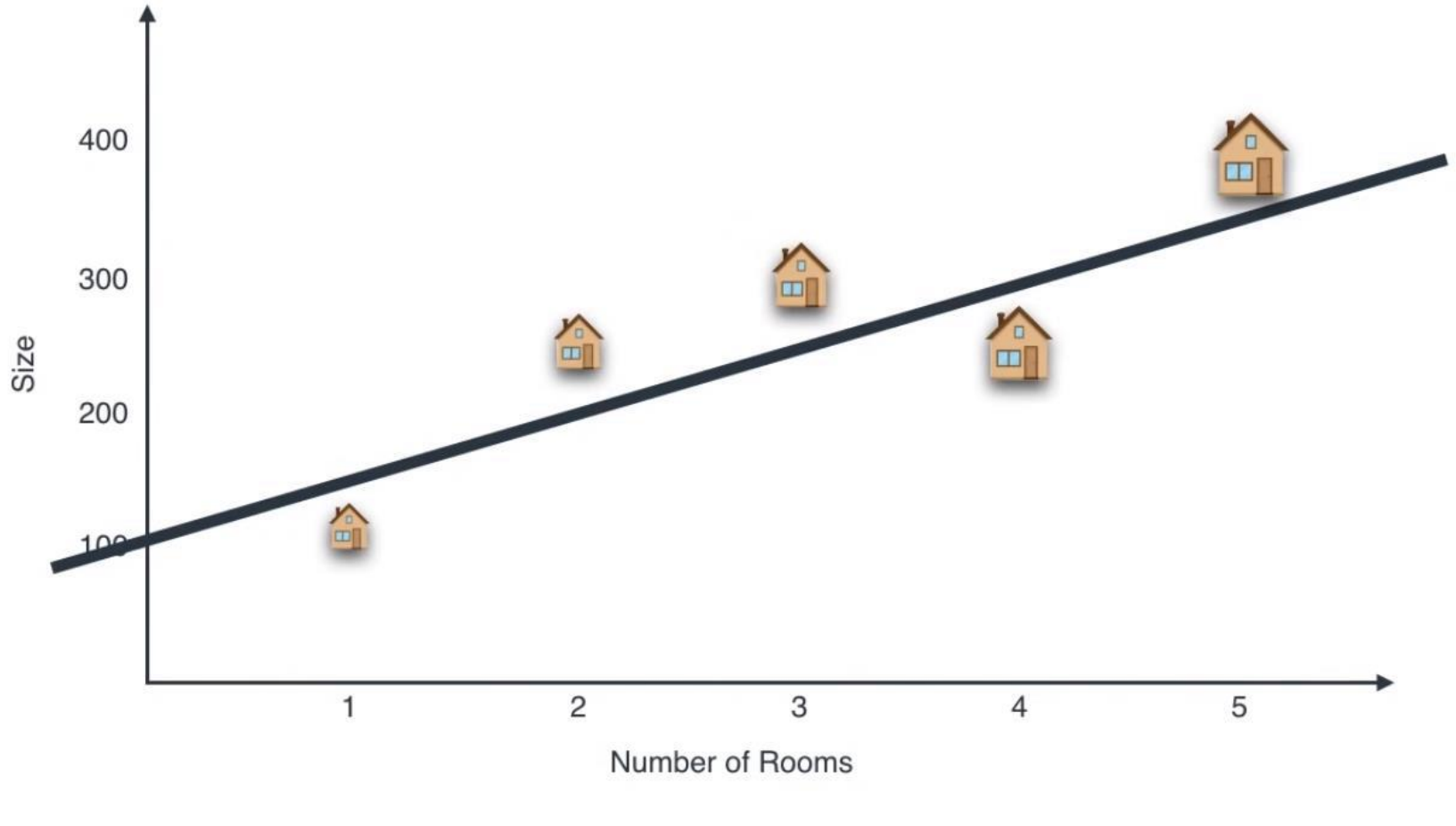
- The “best” vector is the one that gives you the biggest variance after the projection.

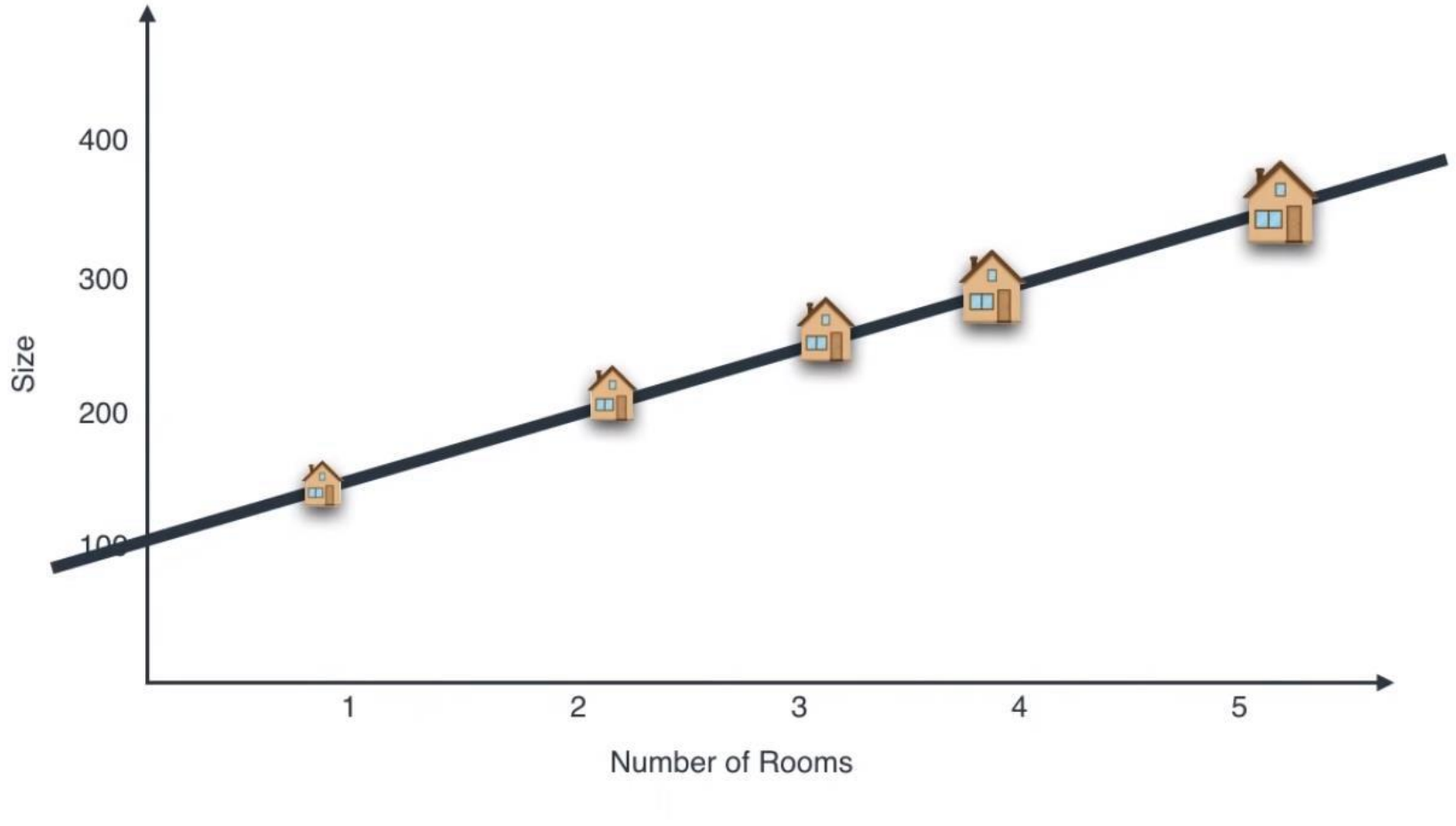
- Each principal component is built to maximize the variance explained by it while adhering to the requirement that it be orthogonal to all other principal components.
- The principal components are computed as linear combinations of the original variables.
- Thus, each principal component is guaranteed to capture a unique and non-redundant part of the variation in the data.



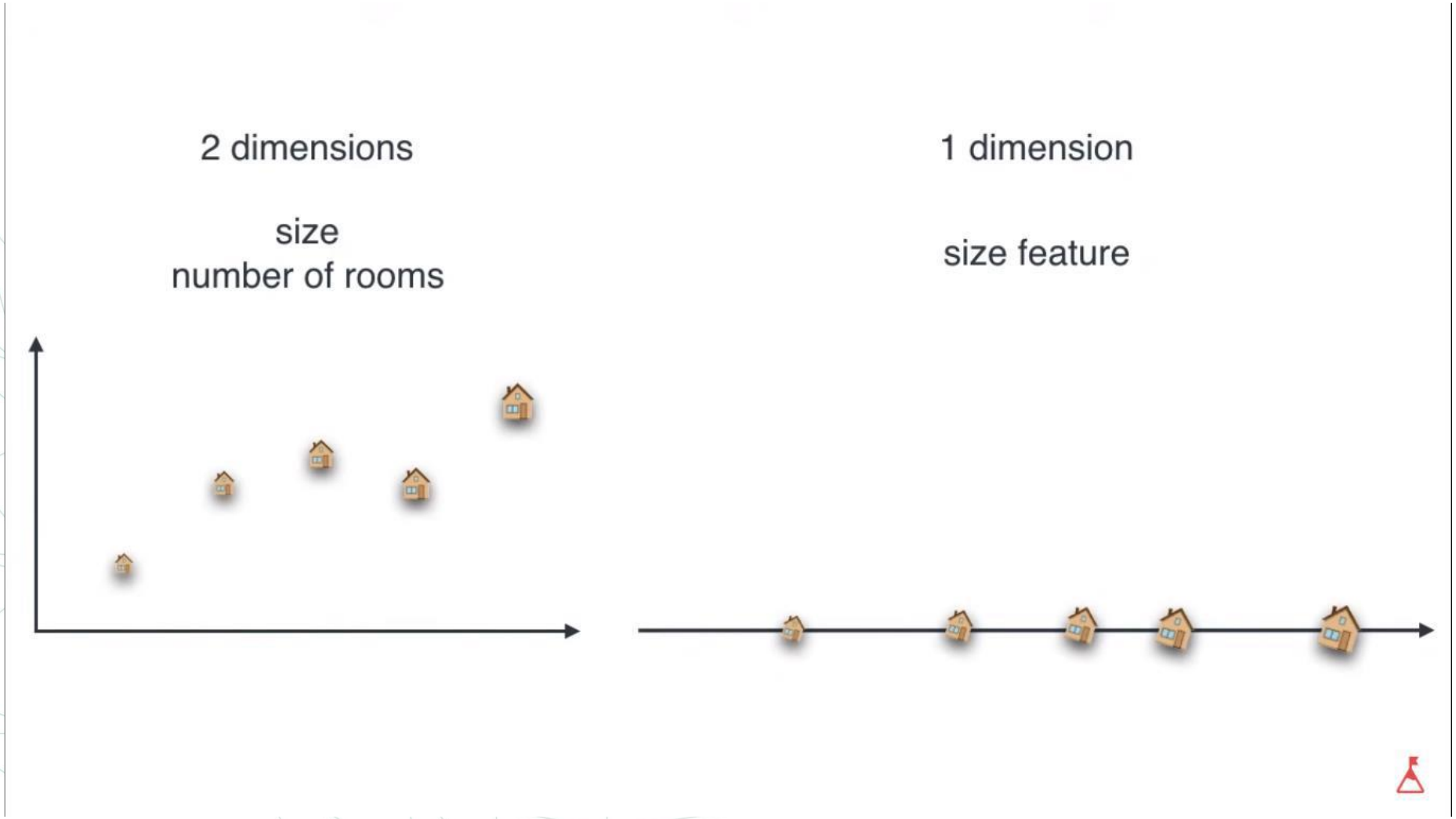



Size
400
300
200
100
1
2
3
4
5
Number of Rooms


400
300
200
Size
1
2
3
4
5
Number of Rooms


400
300
200
Size
1
2
3
4
5
Number of Rooms


Size feature


2 dimensions
size
number of rooms
1 dimension
size feature

Housing Data

5 dimensions

Size
Number of rooms
Number of bathrooms
Schools around
Crime rate

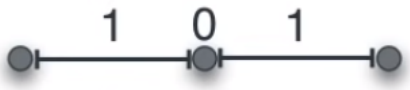
2 dimensions

Size feature

Location feature

Variance

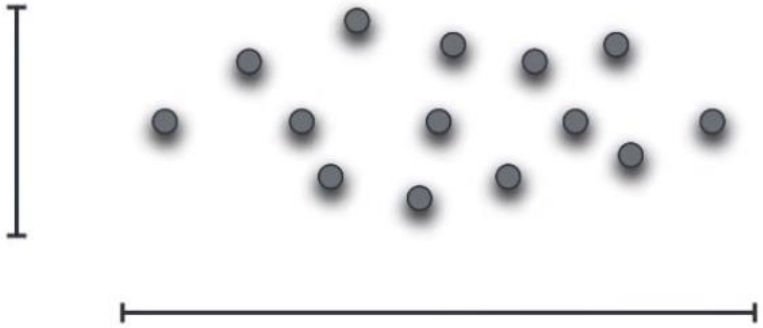
Variance



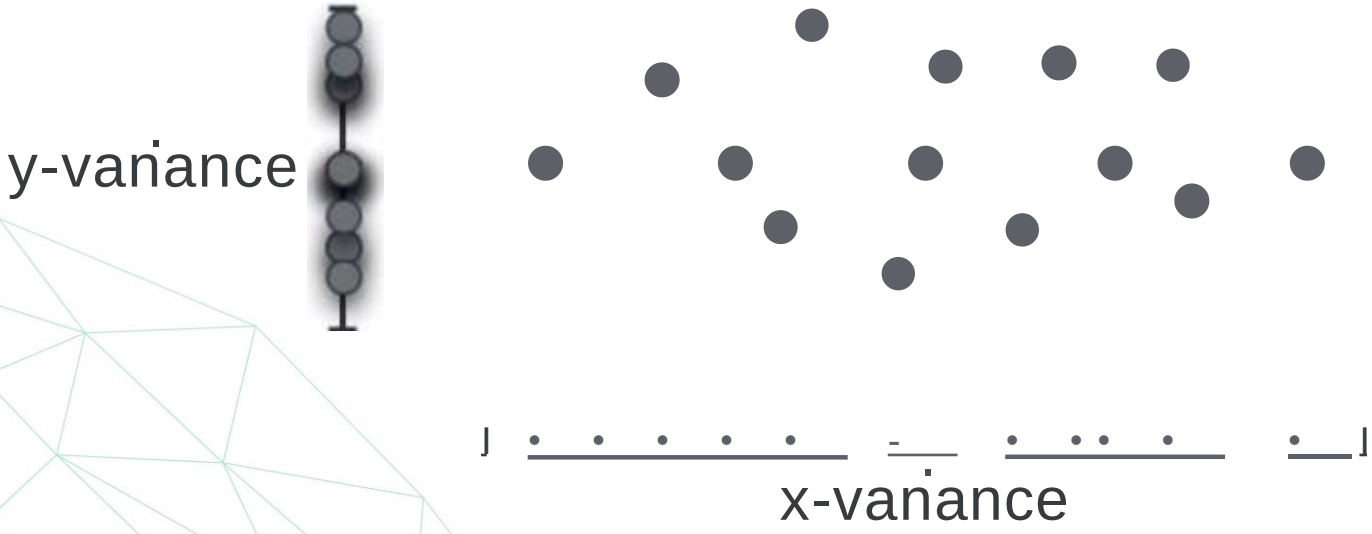
$$\text{Variance} = \frac{1^2+0^2+1^2}{3} = 2/3$$

$$\text{Variance} = \frac{5^2+0^2+5^2}{3} = 50/3$$

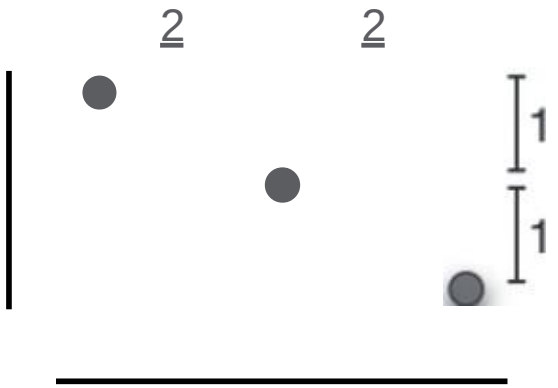
Variance?

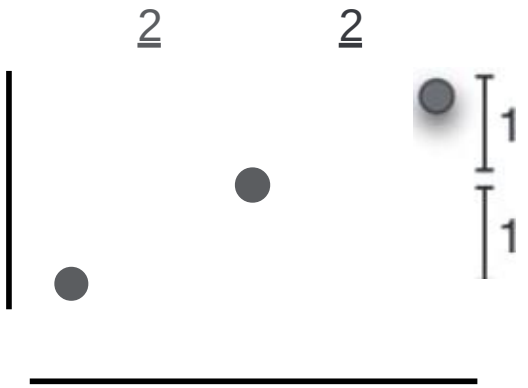


Variance?



Variance?

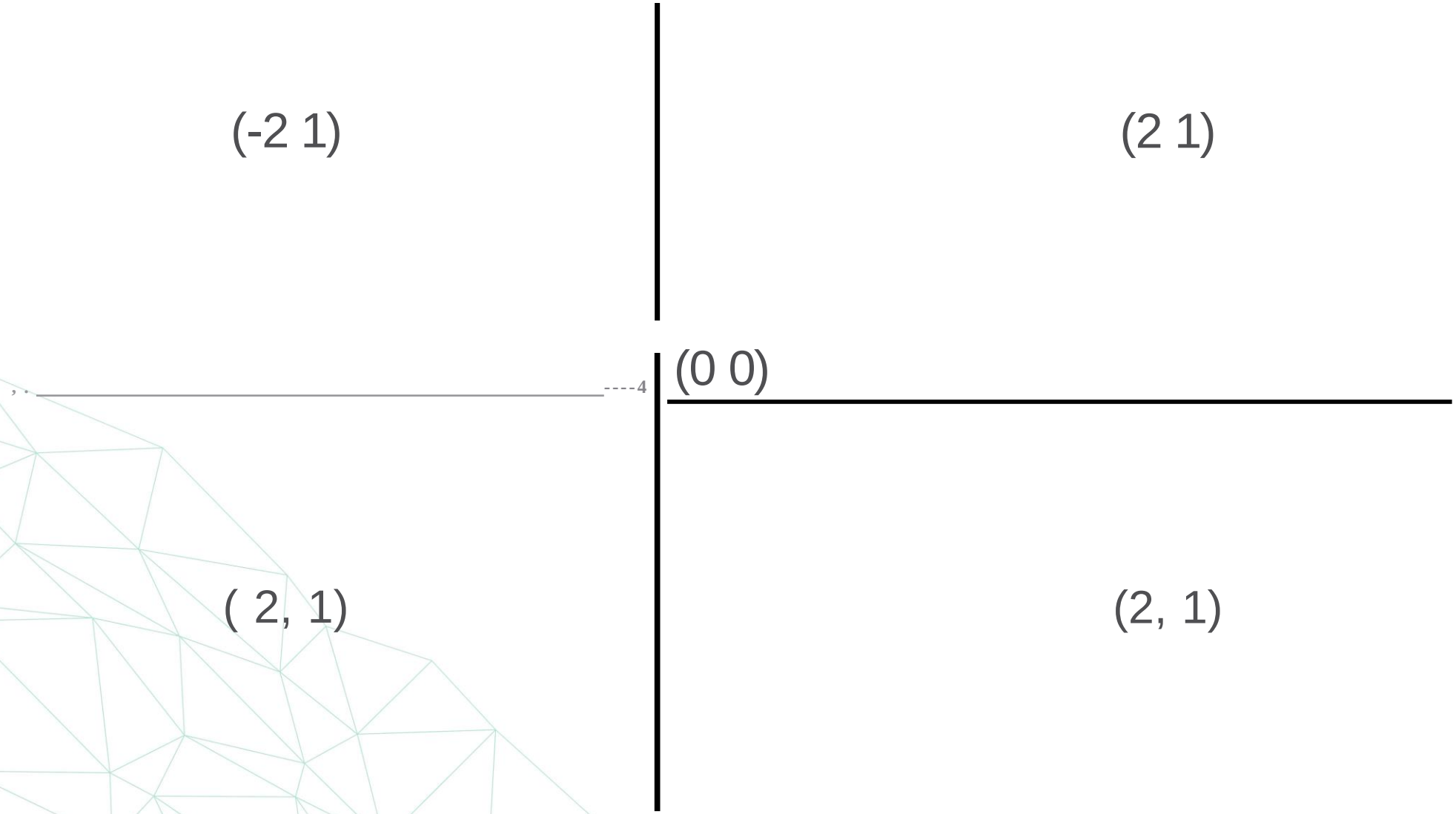




$$\text{x-variance} = \frac{22+02+22}{3} = \mathbf{8/3}$$

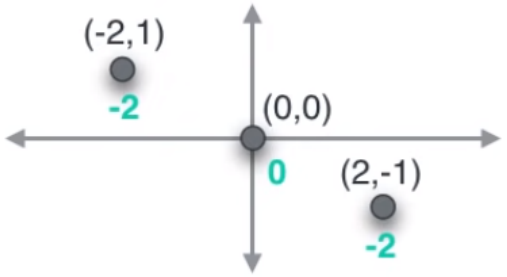
$$\text{y-variance} = \frac{12+02+12}{3} = \mathbf{2/3}$$

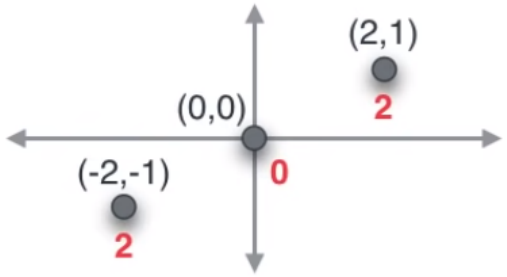
Covariance



How similar the variances of features are?

Covariance



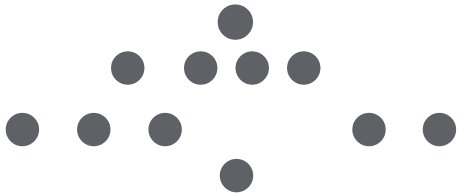


$$\text{covariance} = \frac{(-2) + 0 + (-2)}{3} = -4/3$$

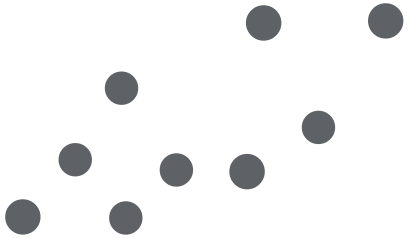
$$\text{covariance} = \frac{2 + 0 + 2}{3} = 4/3$$

Covariance

negative
covariance

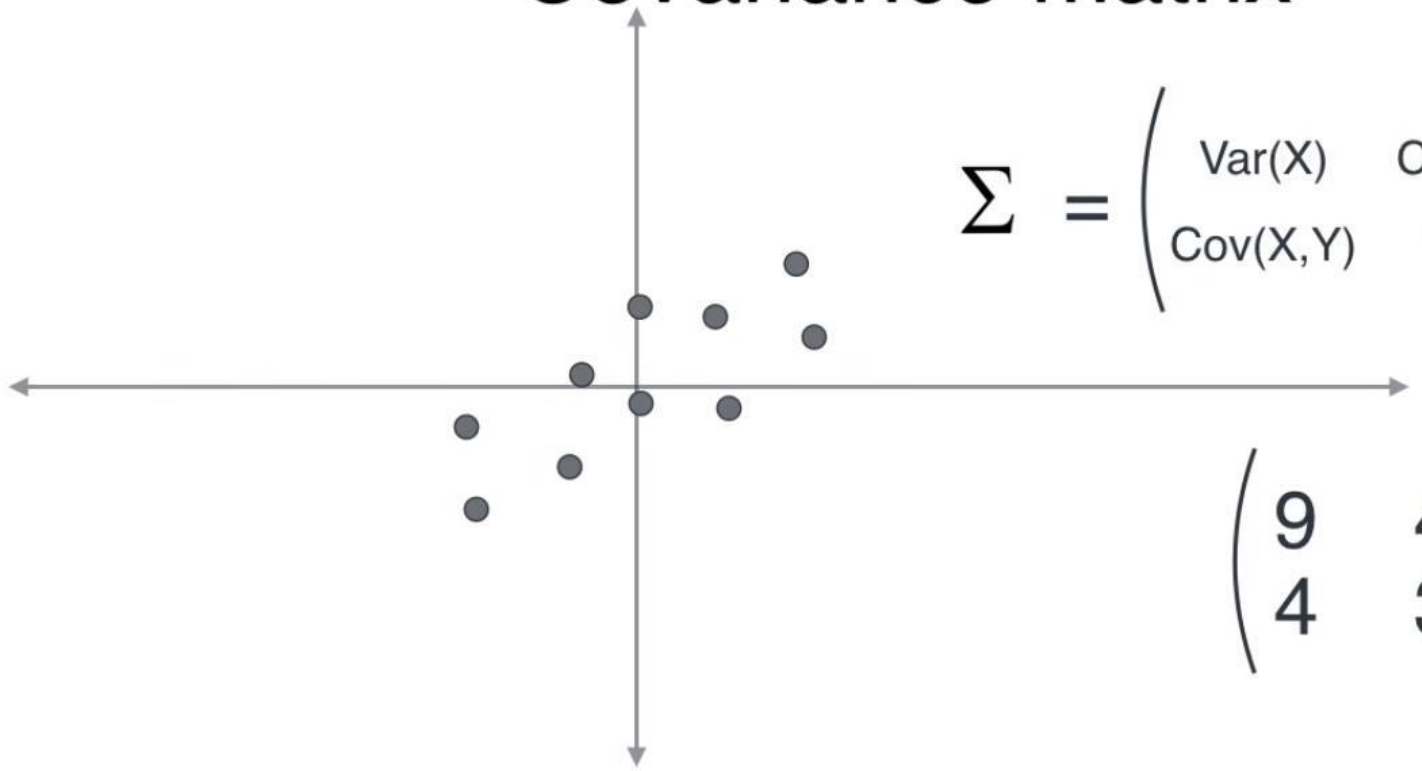


covariance zero
(or very small)



positive
covariance

Covariance matrix



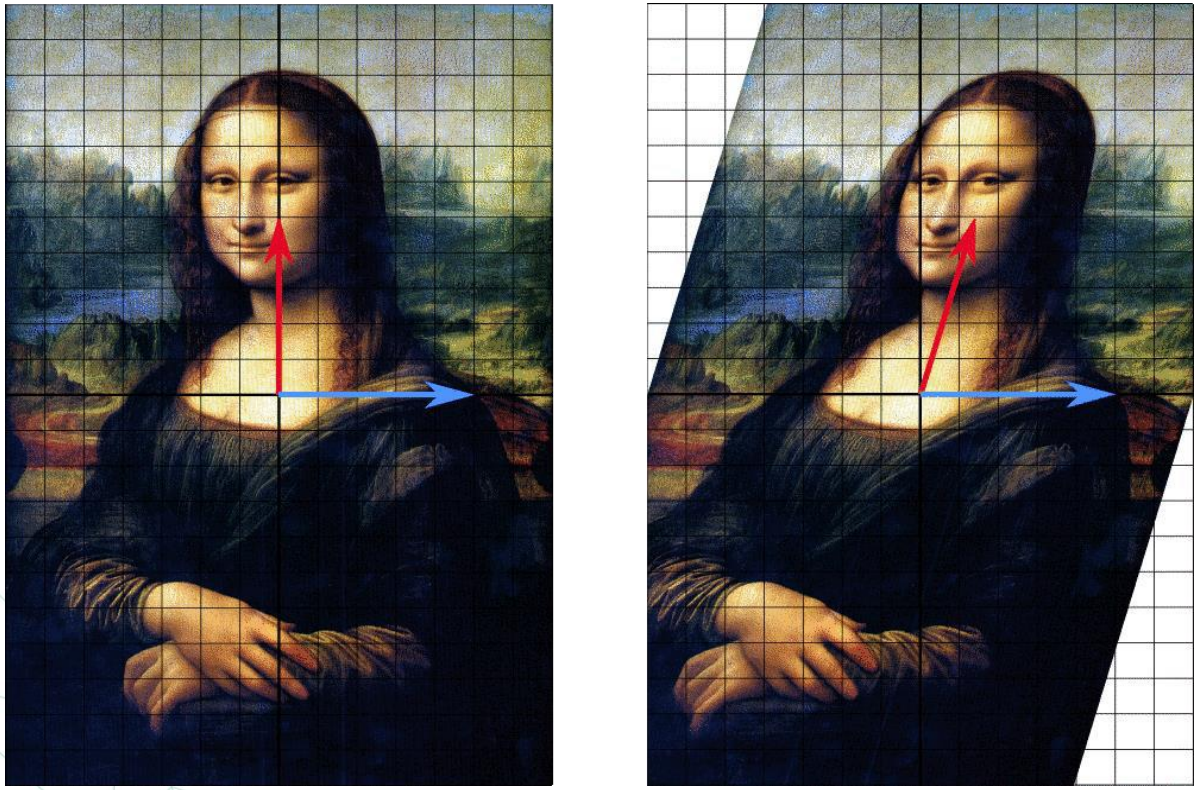
$$\Sigma = \begin{pmatrix} \text{Var}(X) & \text{Cov}(X,Y) \\ \text{Cov}(X,Y) & \text{Var}(Y) \end{pmatrix}$$

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

Eigenvalues and eigenvectors

- The **eigenvectors** for a **linear transformation** matrix are the set of vectors that are only stretched, with no rotation or shear.
- The eigenvalue is the **factor** by which an eigenvector is **stretched/scaled**.
- An **eigenvector** is a **nonzero** vector that changes at most by a scalar factor when that linear transformation is applied to it.

Eigenvalues and Eigenvectors



- In this shear mapping the red arrow changes direction, but the blue arrow does not.
- The blue arrow is an eigenvector of this shear mapping because it does not change direction, and since its length is unchanged, its eigenvalue is 1.

Eigenvalues and Eigenvectors

- Let A be a square matrix (in our case the covariance matrix), v a vector and λ a scalar that satisfies $Av = \lambda v$, then λ is called eigenvalue associated with eigenvector v of A.

$$Av - \lambda v = 0$$

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

- Since we have already know v is a non- zero vector, only way this equation can be equal to zero, if

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

$T: \mathbb{R}^n \to \mathbb{R}^n$

$T(\vec{x}) = A\vec{x}$

$T(\vec{v}) = A\vec{v} = \lambda\vec{v}$

λ: eigen values

$\vec{v}$: eigen vector

$A\vec{v} = \lambda\vec{v}$ $\quad \vec{v} = \vec{0}$

$\vec{v} \neq \vec{0}$

$\vec{v} = I_n\vec{v}$

$\vec{0} = \lambda\vec{v} - A\vec{v}$

$\lambda I_n\vec{v} - A\vec{v} = \vec{0}$

$(\lambda I_n - A)\vec{v} = \vec{0}$

$(\lambda I_n - A)$: some matrix

$A\vec{v} = \lambda\vec{v}$ for nonzero $\vec{v}$'s if and only if

$\det(\lambda I_n - A) = 0$

λ is an eigenvalue of A iff $\det(\lambda I_n - A) = 0$

$A = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$

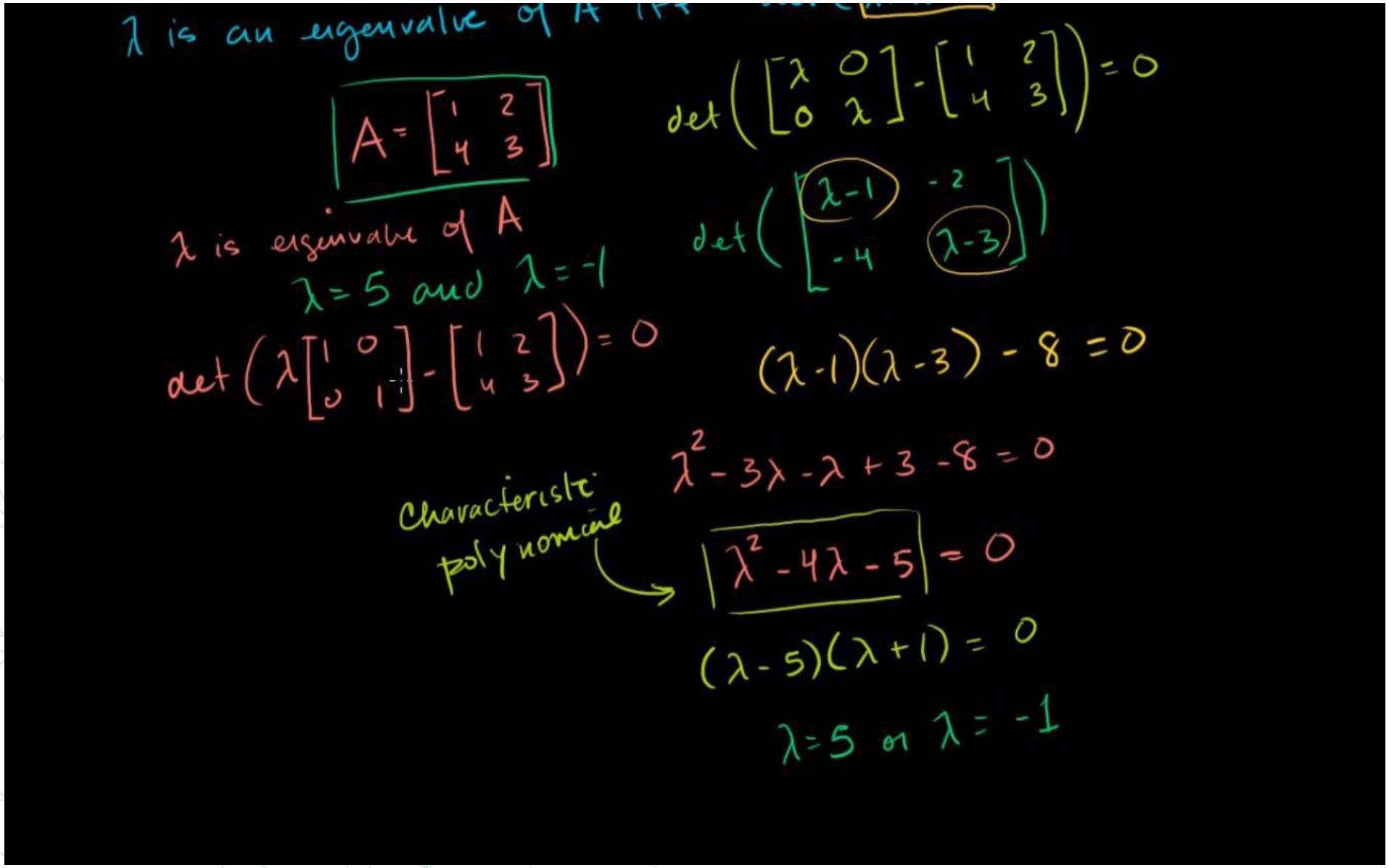
$\det\left(\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}\right) = 0$

λ is eigenvalue of A

$\det\left(\begin{bmatrix} \lambda-1 & -2 \\ -4 & \lambda-3 \end{bmatrix}\right)$

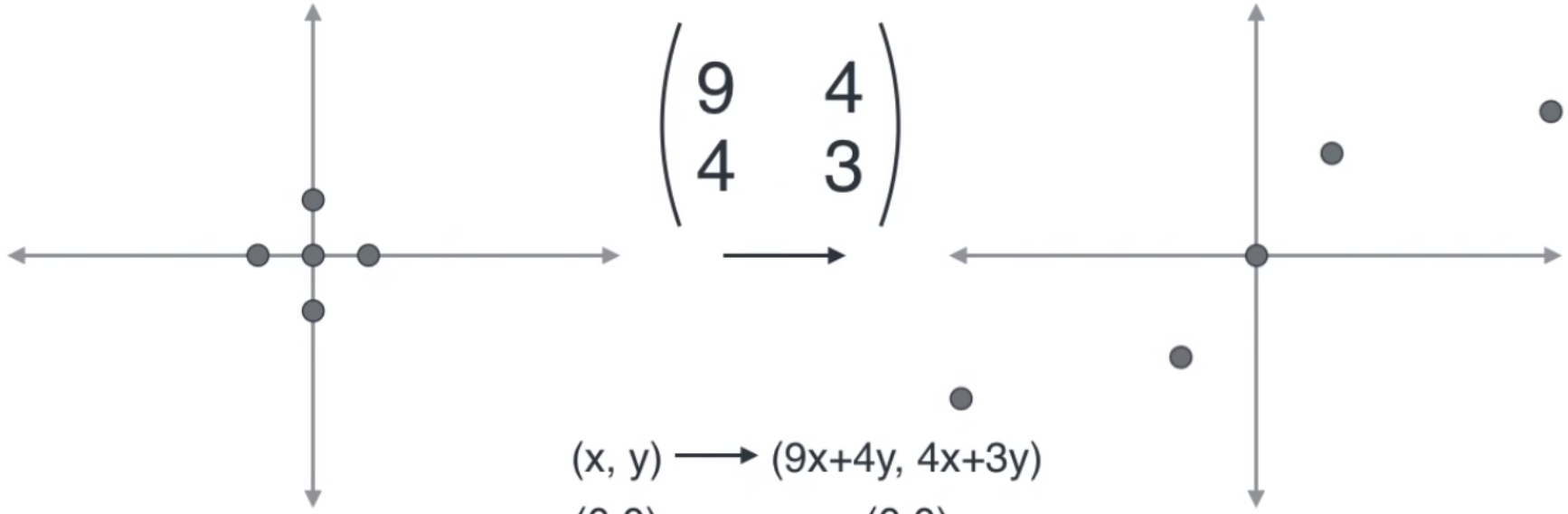
$\det\left(\lambda\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}\right) = 0$

$(\lambda-1)(\lambda-3) - 8 = 0$


λ is an eigenvalue of A
A = [1 2; 4 3]
λ is eigenvalue of A
λ = 5 and λ = -1
det(λ[1 0; 0 1] - [1 2; 4 3]) = 0
det([λ 0; 0 λ] - [1 2; 4 3]) = 0
det([λ-1 -2; -4 λ-3])
(λ-1)(λ-3) - 8 = 0
λ² - 3λ - λ + 3 - 8 = 0
Characteristic polynomial
λ² - 4λ - 5 = 0
(λ - 5)(λ + 1) = 0
λ = 5 or λ = -1

Linear Transformations

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

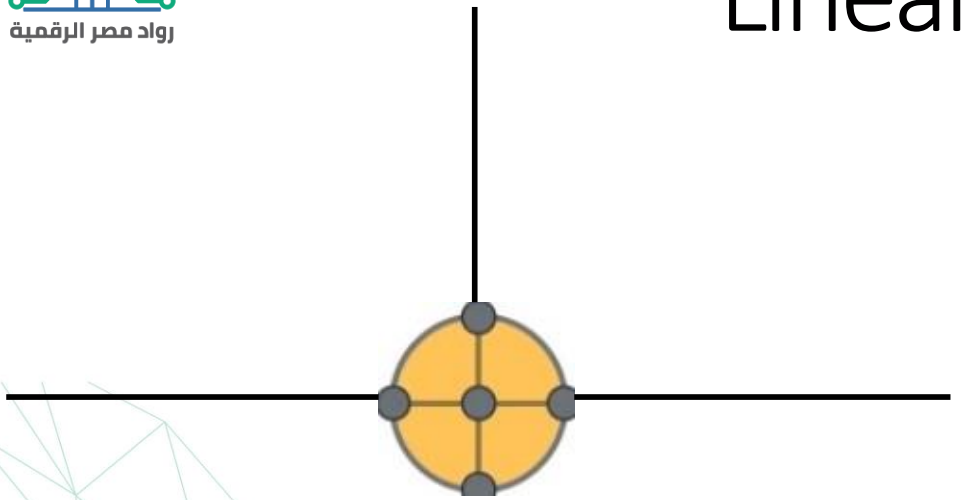


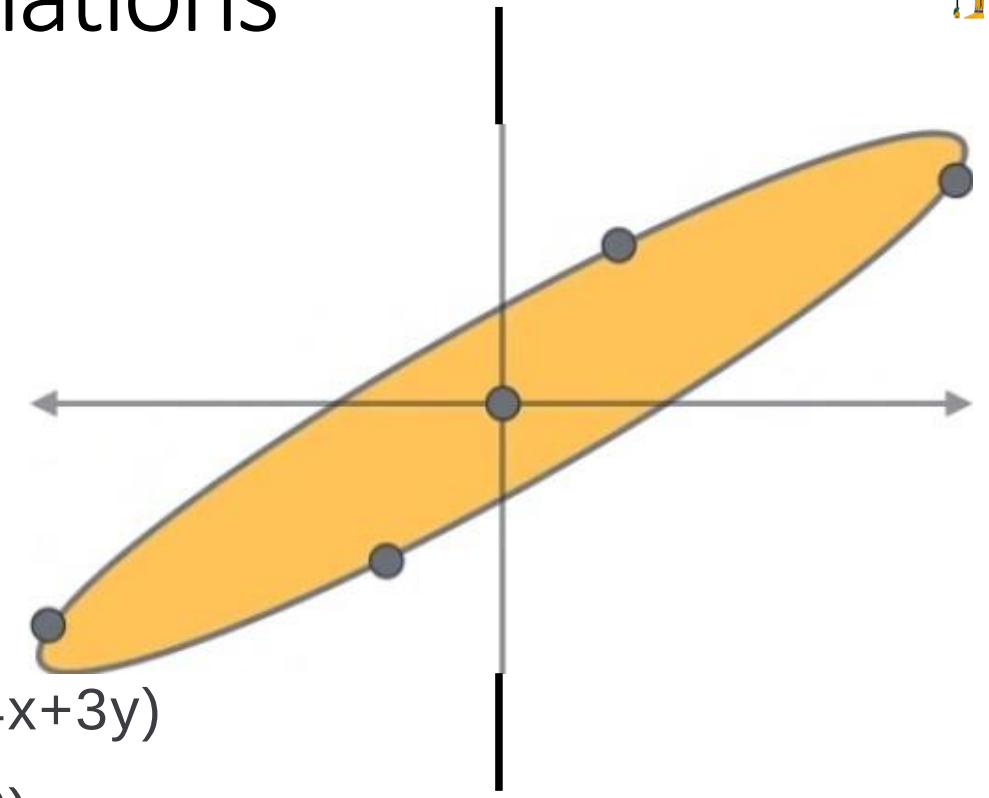
$(x, y) \longrightarrow (9x+4y, 4x+3y)$

(0,0)	(0,0)
(1,0)	(9,4)
(0,1)	(4,3)
(-1,0)	(-9,-4)
(0,-1)	(-4,-3)

Linear Transformations

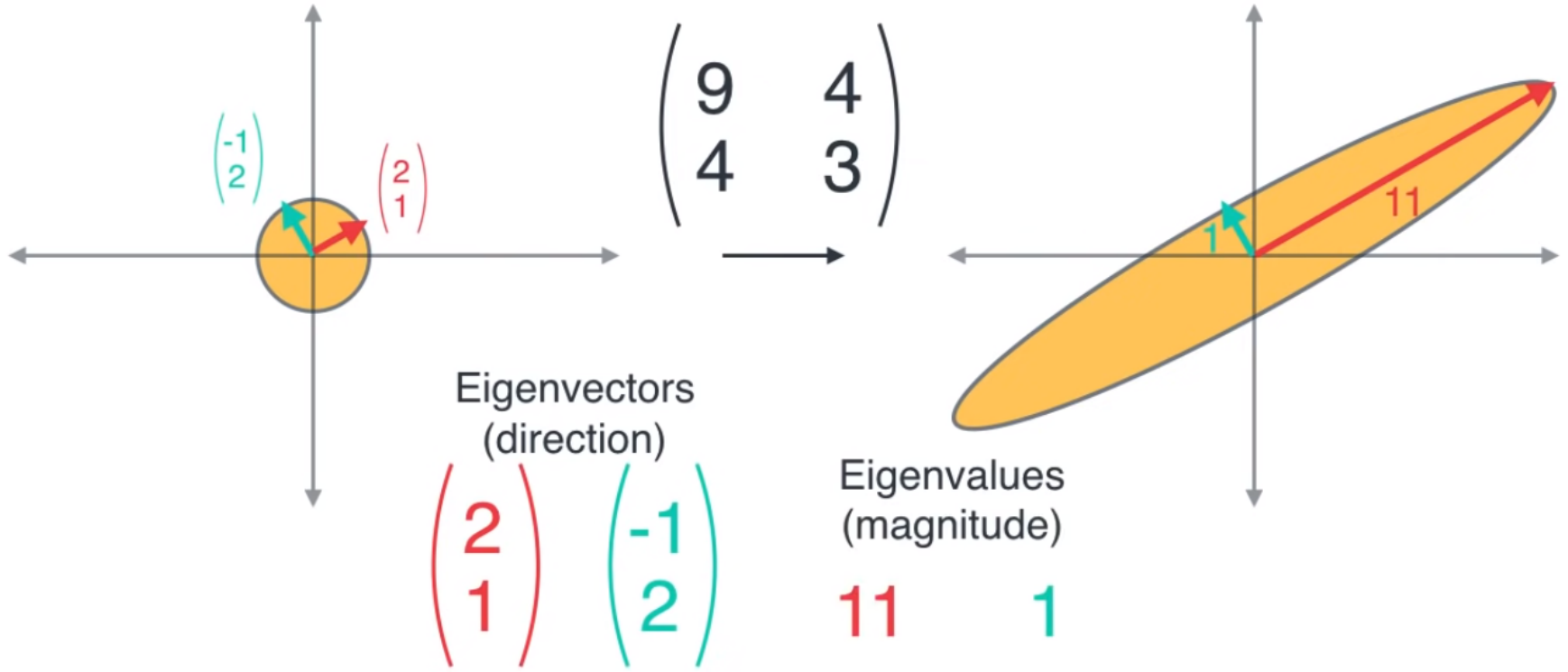
$$\begin{vmatrix} 9 & 4 \\ 4 & 3 \end{vmatrix}$$



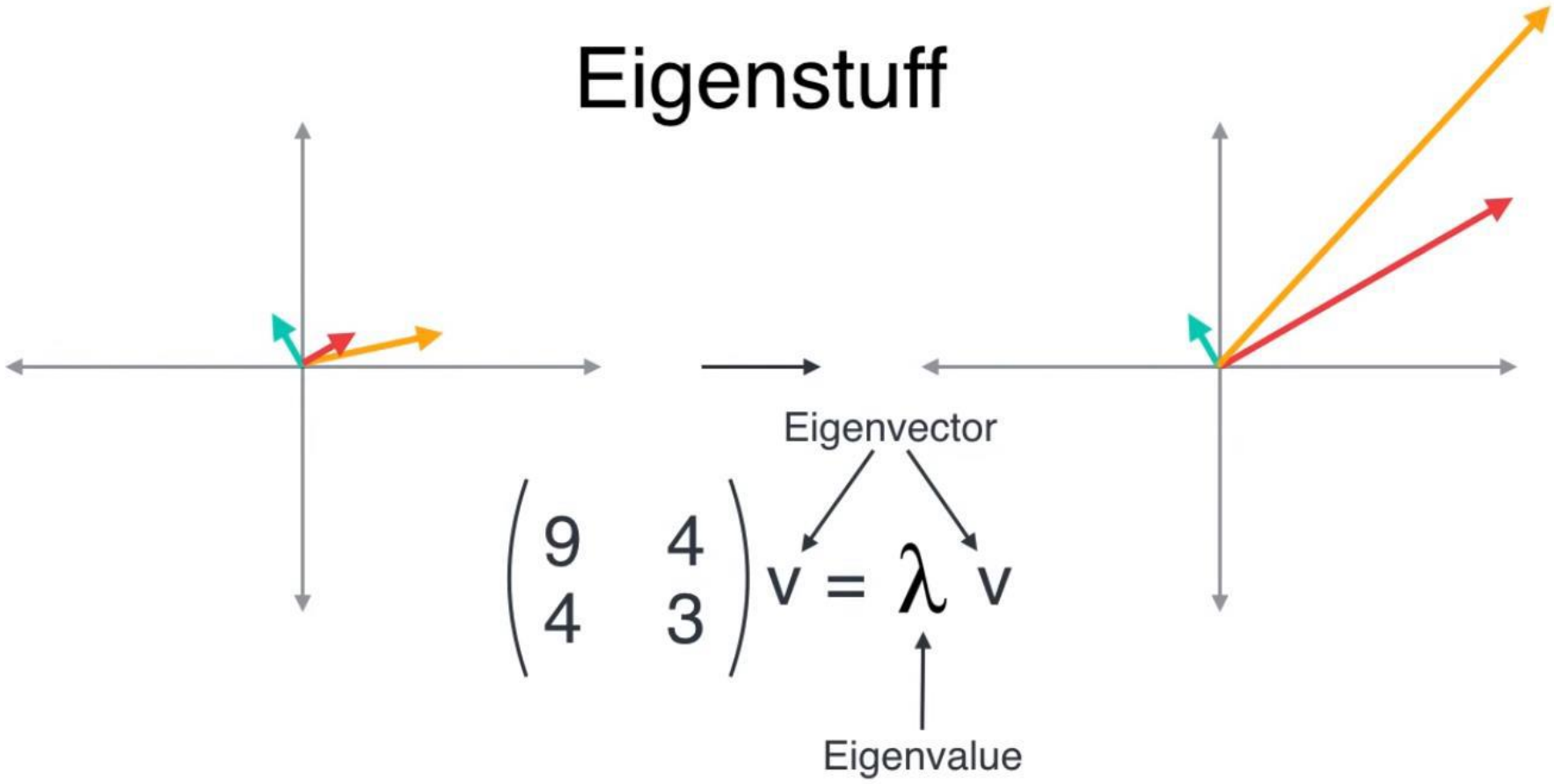


(x, y)	∴ (9x+4y, 4x+3y)
(0,0)	(0,0)
(1,0)	(9,4)
(0,1)	(4,3)
(-1,0)	(-9,-4)
(0.-1)	(-4.-3)

Linear Transformations



Eigenstuff


Eigenvector
$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix} v = \lambda v$
Eigenvalue

Eigenvalues

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

$$\begin{vmatrix} x-9 & -4 \\ -4 & x-3 \end{vmatrix} = (x-9)(x-3) - (-4)(-4) = x^2 - 12x + 11$$

$$= (x-11)(x-1)$$

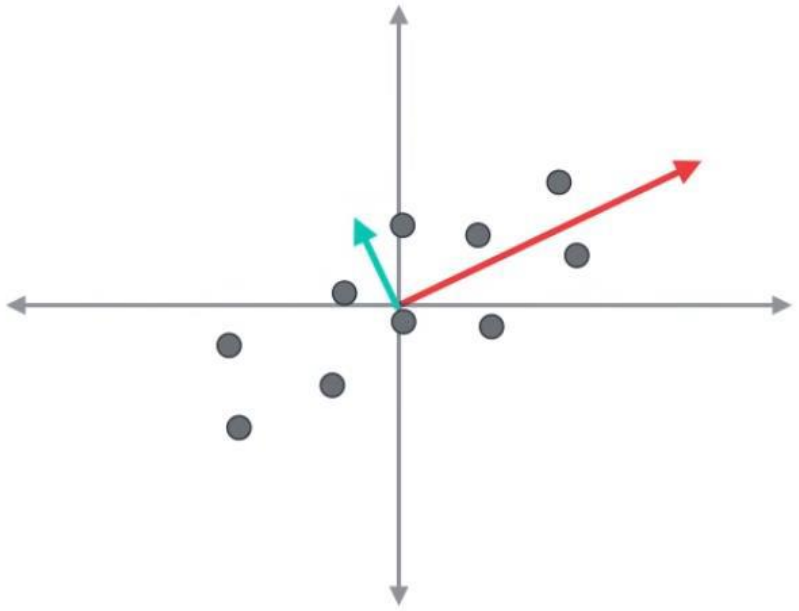
Eigenvalues 11 and 1

Eigenvectors

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = 11\begin{pmatrix} u \\ v \end{pmatrix} \qquad \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = 1\begin{pmatrix} u \\ v \end{pmatrix} \qquad \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

Principal Component Analysis (PCA)



$$\Sigma = \begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

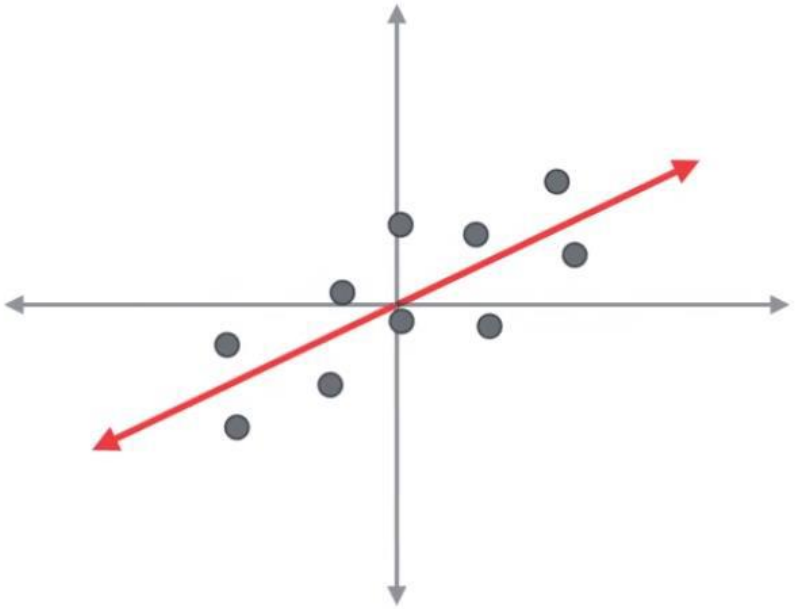
$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

Eigenvectors (direction)

11 1

Eigenvalues (magnitude)

Principal Component Analysis (PCA)



$$\Sigma = \begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

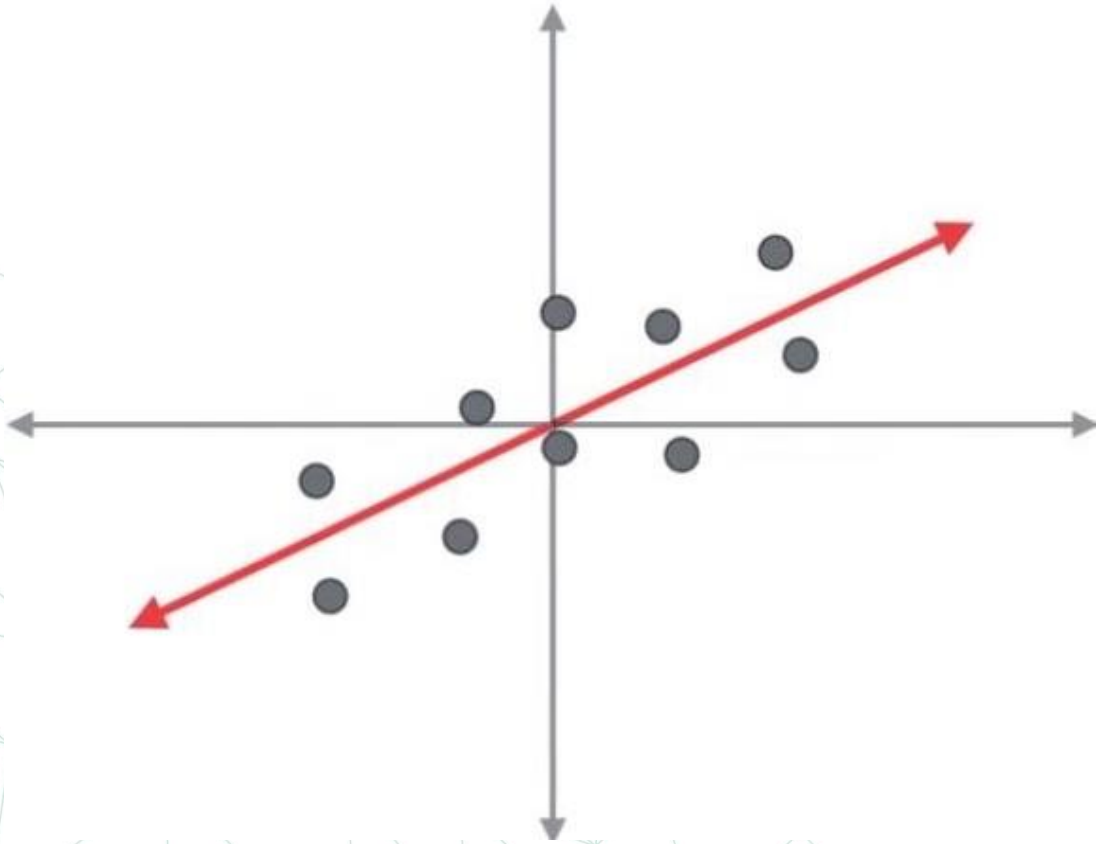
$$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

Eigenvectors (direction)

11

Eigenvalues (magnitude)

Principal Component Analysis (PCA)



$$L = \begin{vmatrix} 9 & 4 \\ 4 & 3 \end{vmatrix}$$

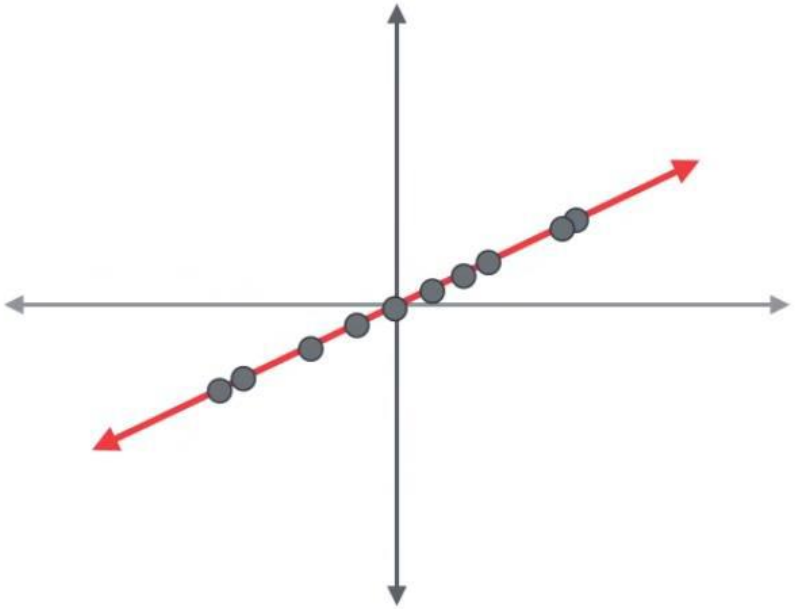
$$\begin{vmatrix} 2 \\ 1 \end{vmatrix}$$

Eigenvectors (direction)

11

Eigenvalues (magnitude)

Principal Component Analysis (PCA)



$$\Sigma = \begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

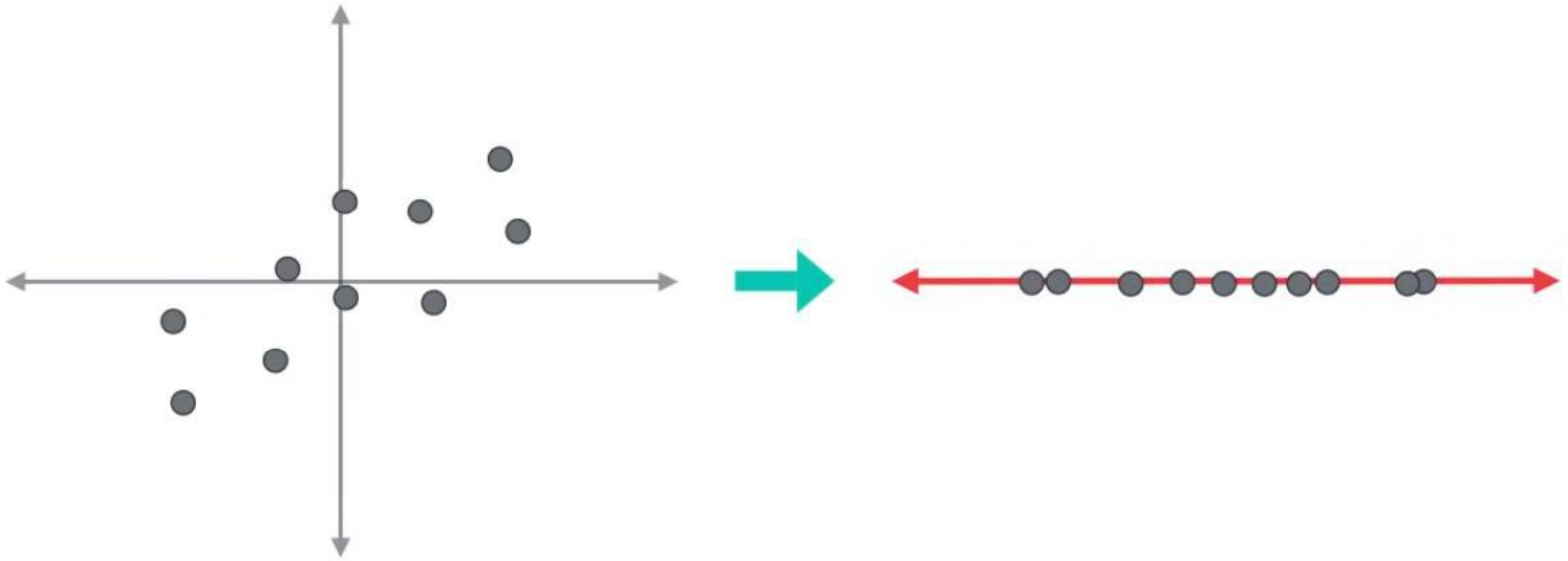
$$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

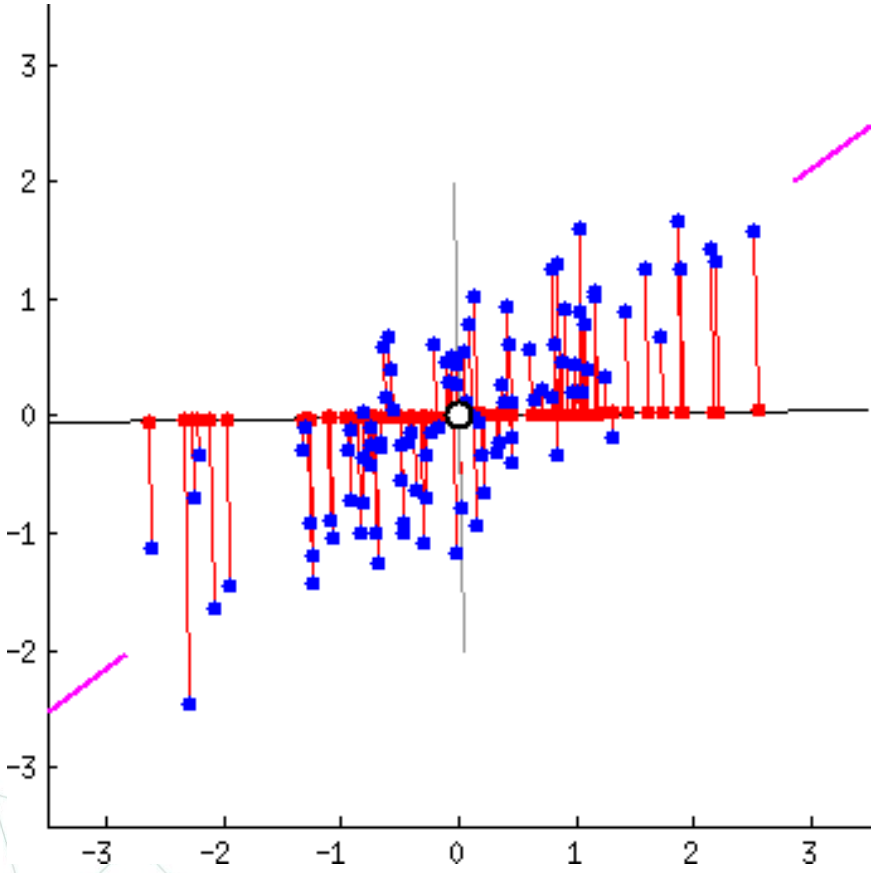
Eigenvectors (direction)

11

Eigenvalues (magnitude)

Principal Component Analysis (PCA)

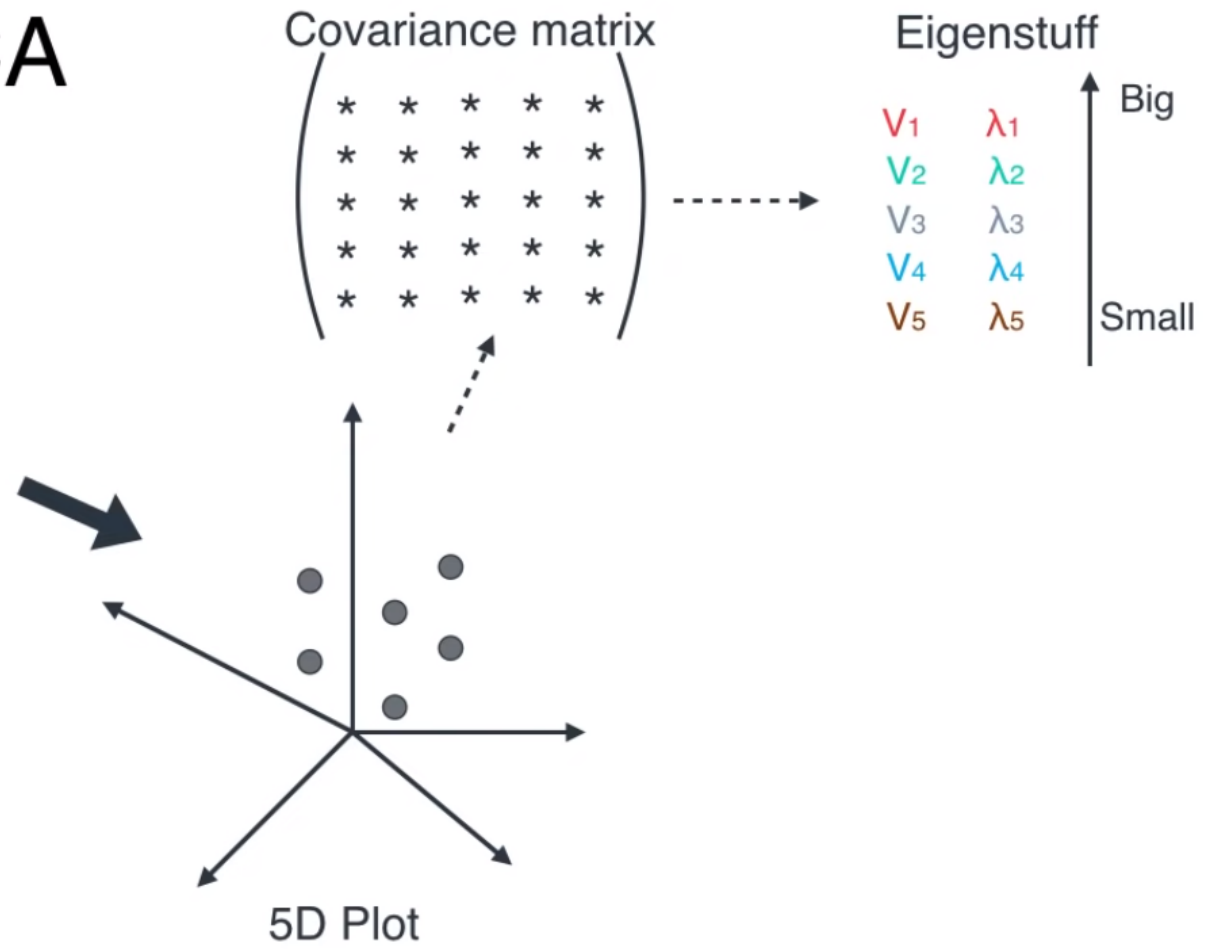


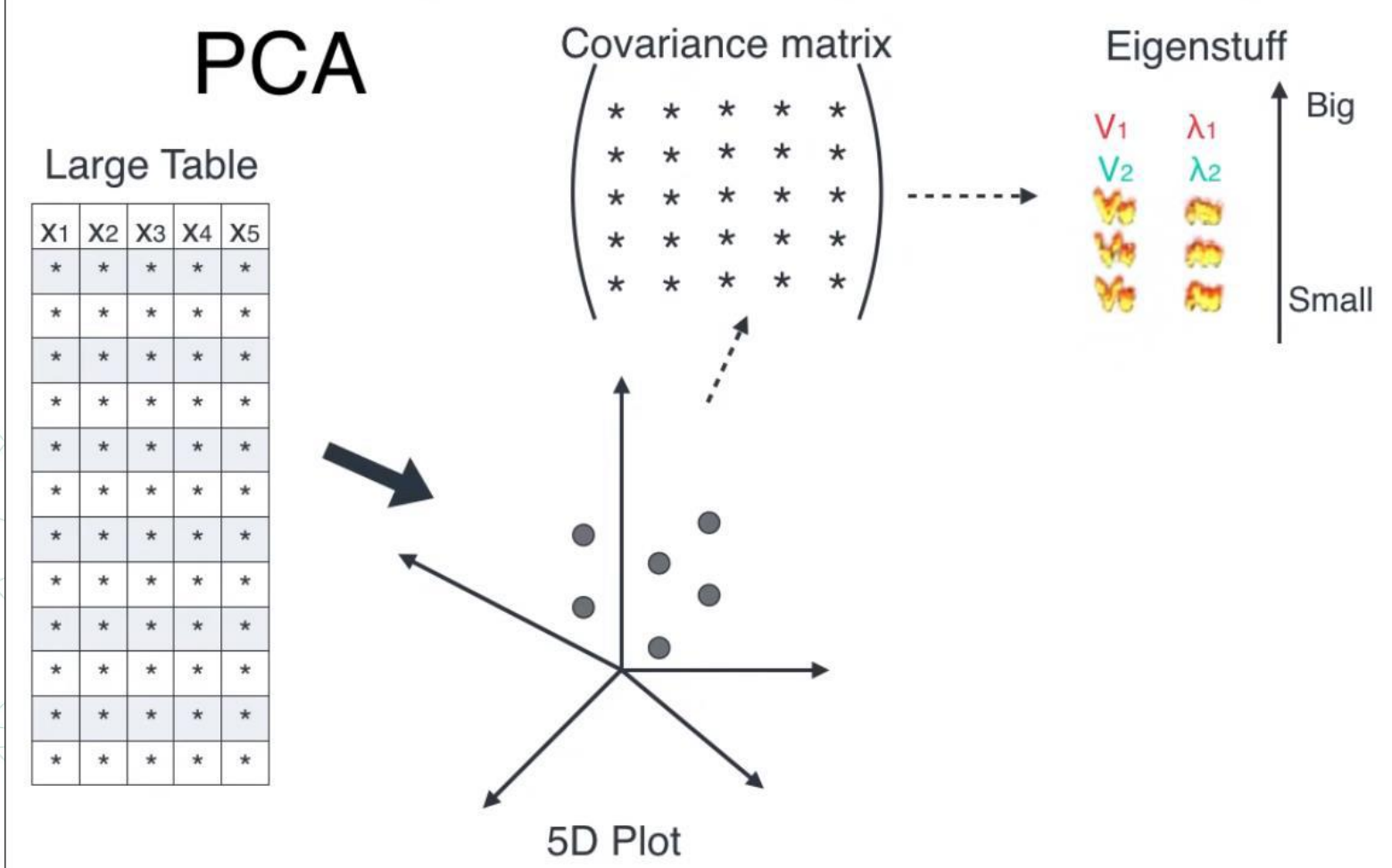

3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

PCA

Large Table

X1	X2	X3	X4	X5
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*

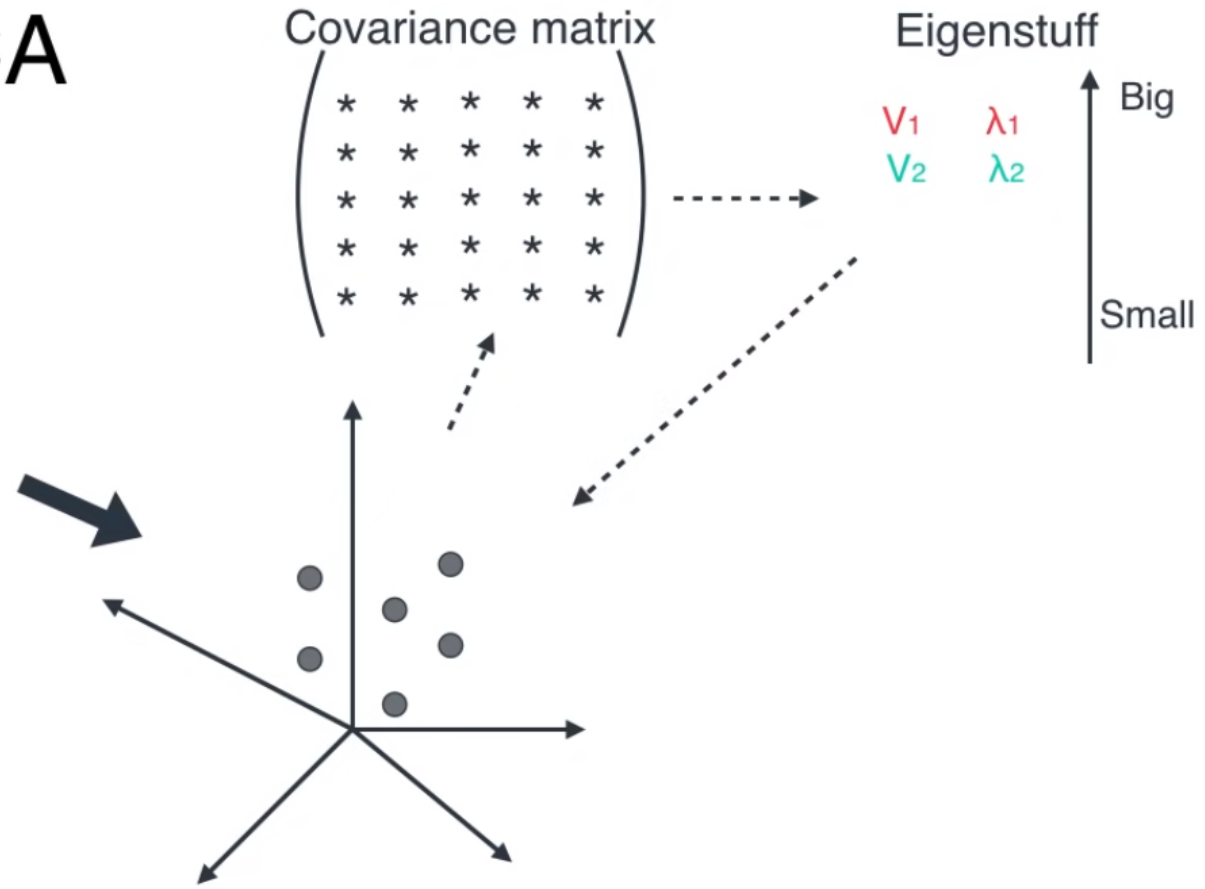


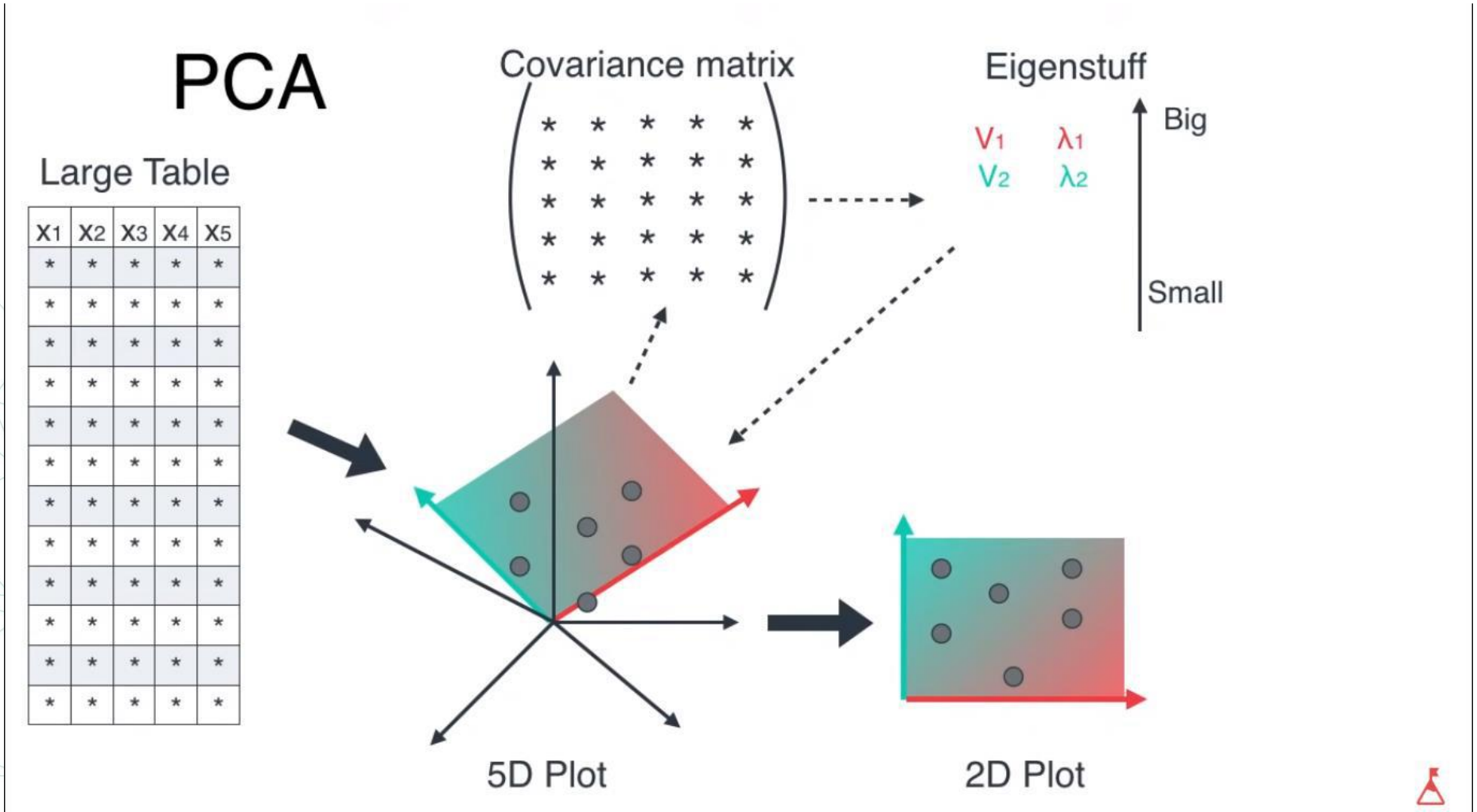

PCA
Large Table
X1 X2 X3 X4 X5
Covariance matrix
Eigenstuff
V1 λ1
V2 λ2
Big
Small
5D Plot

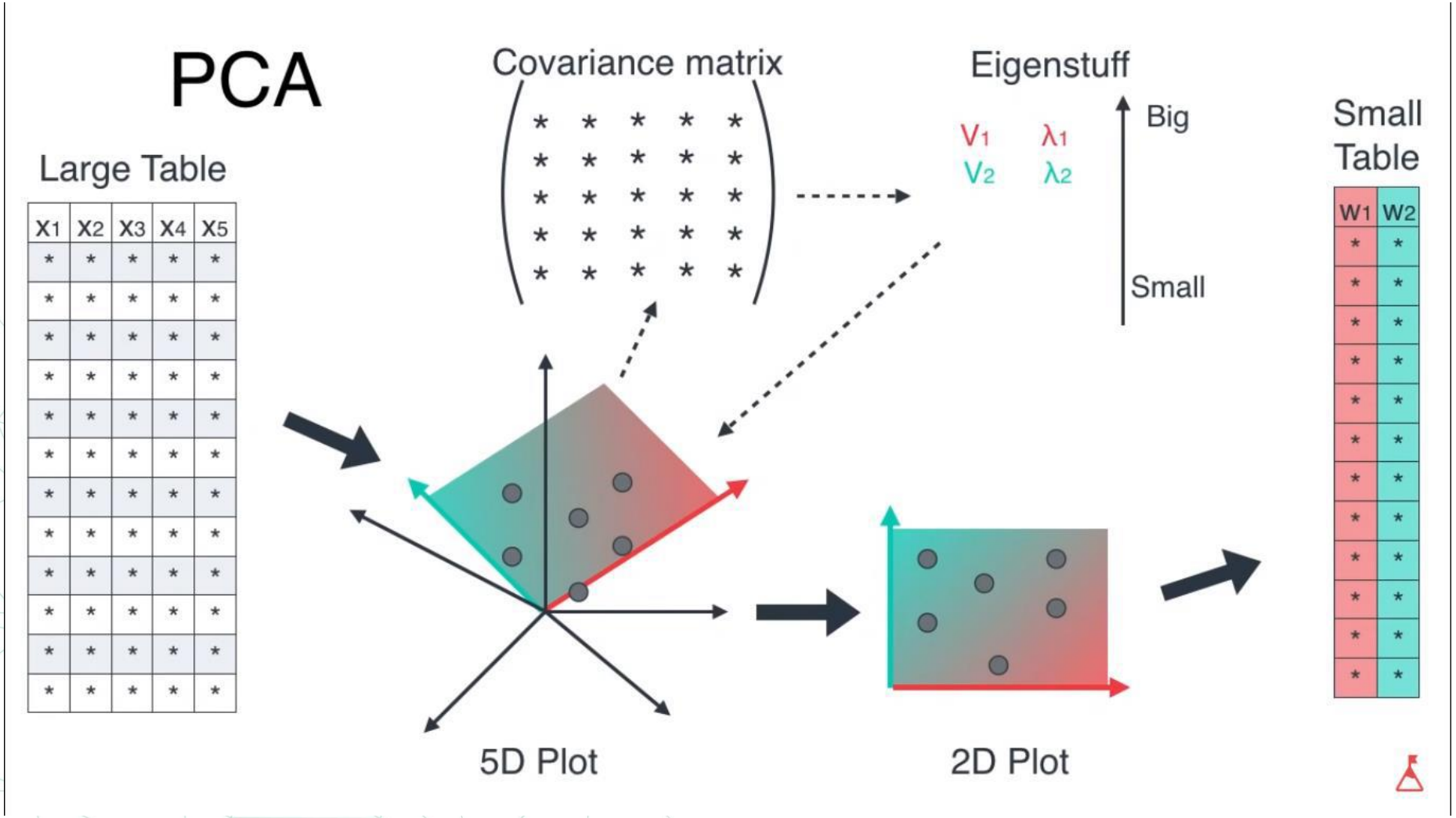
PCA

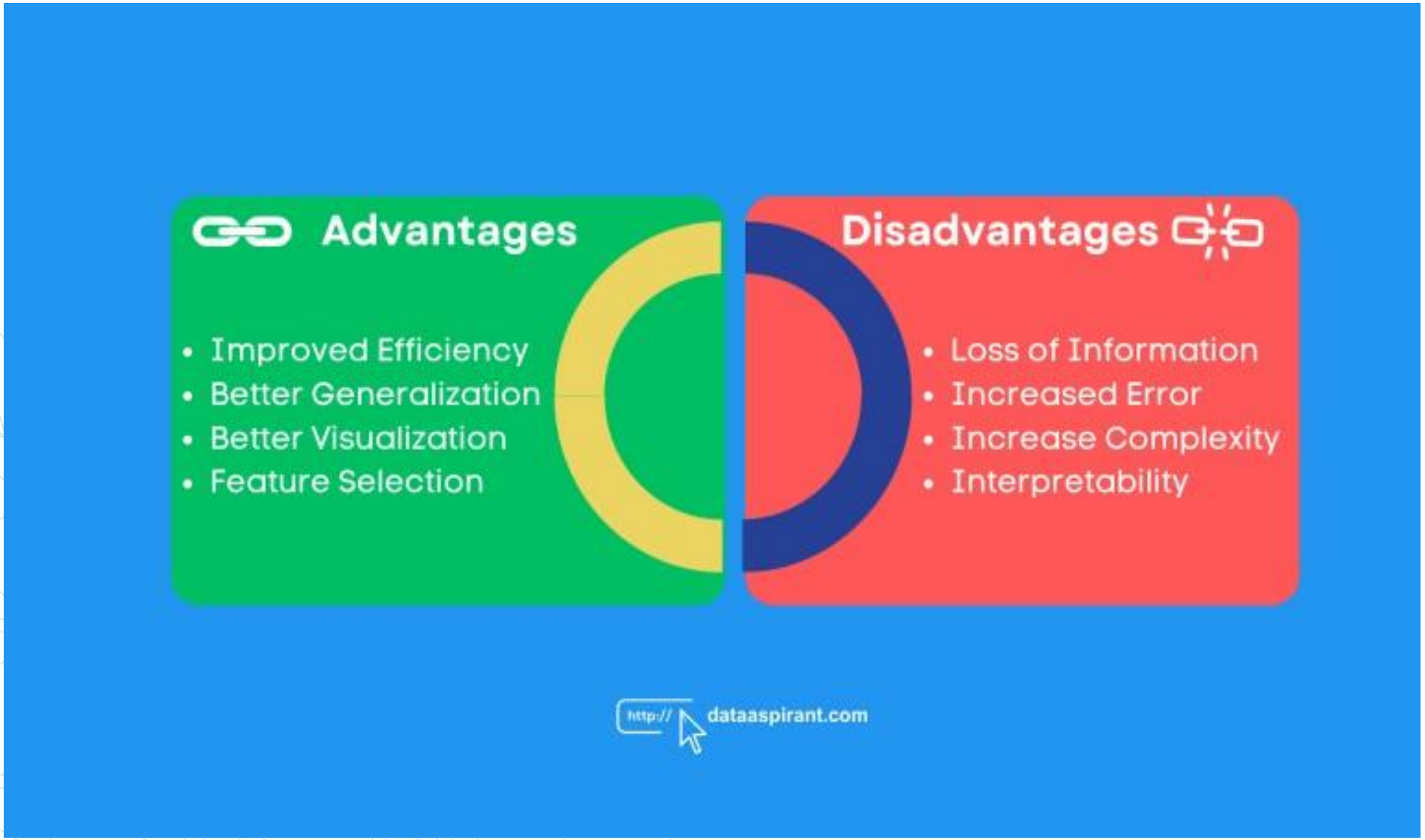
Large Table

X1	X2	X3	X4	X5
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*




PCA
Large Table
X1 X2 X3 X4 X5
Covariance matrix
Eigenstuff
V1 λ1
V2 λ2
Big
Small
5D Plot
2D Plot


PCA
Large Table
X1 X2 X3 X4 X5
Covariance matrix
Eigenstuff
V1 λ1
V2 λ2
Big
Small
Small Table
W1 W2
5D Plot
2D Plot


Advantages
Improved Efficiency
Better Generalization
Better Visualization
Feature Selection
Disadvantages
Loss of Information
Increased Error
Increase Complexity
Interpretability
http://
dataaspirant.com

let's take a look at the mathematical side of PCA:

- **Step 1:** Standardize the dataset.
- **Step 2:** Calculate the covariance matrix for the features in the dataset.
- **Step 3:** Calculate the eigenvalues and eigenvectors for the covariance matrix.
- **Step 4:** Sort eigenvalues and their corresponding eigenvectors.
- **Step 5:** Pick k eigenvalues and form a matrix of eigenvectors.
- **Step 6:** Transform the original matrix.

f1	f2	f3	f4
1	2	3	4
5	5	6	7
1	4	2	3
5	3	2	1
8	1	2	2

Step 1: Standardize the dataset.

	f1	f2	f3	f4
μ =	4	3	3	3.4
σ =	3	1.58114	1.73205	2.30217

f1	f2	f3	f4
-1	-0.63246	0	0.26062
0.33333	1.26491	1.73205	1.56374
-1	0.63246	-0.57735	-0.17375
0.33333	0	-0.57735	-1.04249
1.33333	-1.26491	-0.57735	-0.60812

Step 2: Calculate the covariance matrix for the features in the dataset.

	f1	f2	f3	f4
f1	var(f1)	cov(f1,f2)	cov(f1,f3)	cov(f1,f4)
f2	cov(f2,f1)	var(f2)	cov(f2,f3)	cov(f2,f4)
f3	cov(f3,f1)	cov(f3,f2)	var(f3)	cov(f3,f4)
f4	cov(f4,f1)	cov(f4,f2)	cov(f4,f3)	var(f4)

$$Cov(x,y) = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{(N-1)}$$

	f1	f2	f3	f4
f1	0.8	-0.25298	0.03849	-0.14479
f2	-0.25298	0.8	0.51121	0.4945
f3	0.03849	0.51121	0.8	0.75236
f4	-0.14479	0.4945	0.75236	0.8

Step 3: Calculate the eigenvalues and eigenvectors for the covariance matrix.

	f1	f2	f3	f4
f1	0.8 - λ	-0.25298	0.03849	-0.14479
f2	-0.25298	0.8- λ	0.51121	0.4945
f3	0.03849	0.51121	0.8 - λ	0.75236
f4	-0.14479	0.4945	0.75236	0.8 - λ

det(A-λI) = 0

$$\begin{pmatrix} 0.800000-\lambda & -(0.252982) & 0.038490 & -(0.144791) \\ -(0.252982) & 0.800000-\lambda & 0.511208 & 0.494498 \\ 0.038490 & 0.511208 & 0.800000-\lambda & 0.752355 \\ -(0.144791) & 0.494498 & 0.752355 & 0.800000-\lambda \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = 0$$

Solving the above equation = 0

λ = 2.51579324 , 1.0652885 , 0.39388704 , 0.02503121

e1	e2	e3	e4
0.161960	-0.917059	-0.307071	0.196162
-0.524048	0.206922	-0.817319	0.120610
-0.585896	-0.320539	0.188250	-0.720099
-0.596547	-0.115935	0.449733	0.654547

Step 4: Sort eigenvalues and their corresponding eigenvectors.
Step 5: Pick k eigenvalues and form a matrix of eigenvectors.

e1	e2
0.161960	-0.917059
-0.524048	0.206922
-0.585896	-0.320539
-0.596547	-0.115935

Top 2 eigenvectors(4*2 matrix)

Step 6: Transform the original matrix.

f1	f2	f3	f4
-1.000000	-0.632456	0.000000	0.260623
0.333333	1.264911	1.732051	1.563740
-1.000000	0.632456	-0.577350	-0.173749
0.333333	0.000000	-0.577350	-1.042493
1.333333	-1.264911	-0.577350	-0.608121

(5,4)

*

e1	e2
0.161960	-0.917059
-0.524048	0.206922
-0.585896	-0.320539
-0.596547	-0.115935

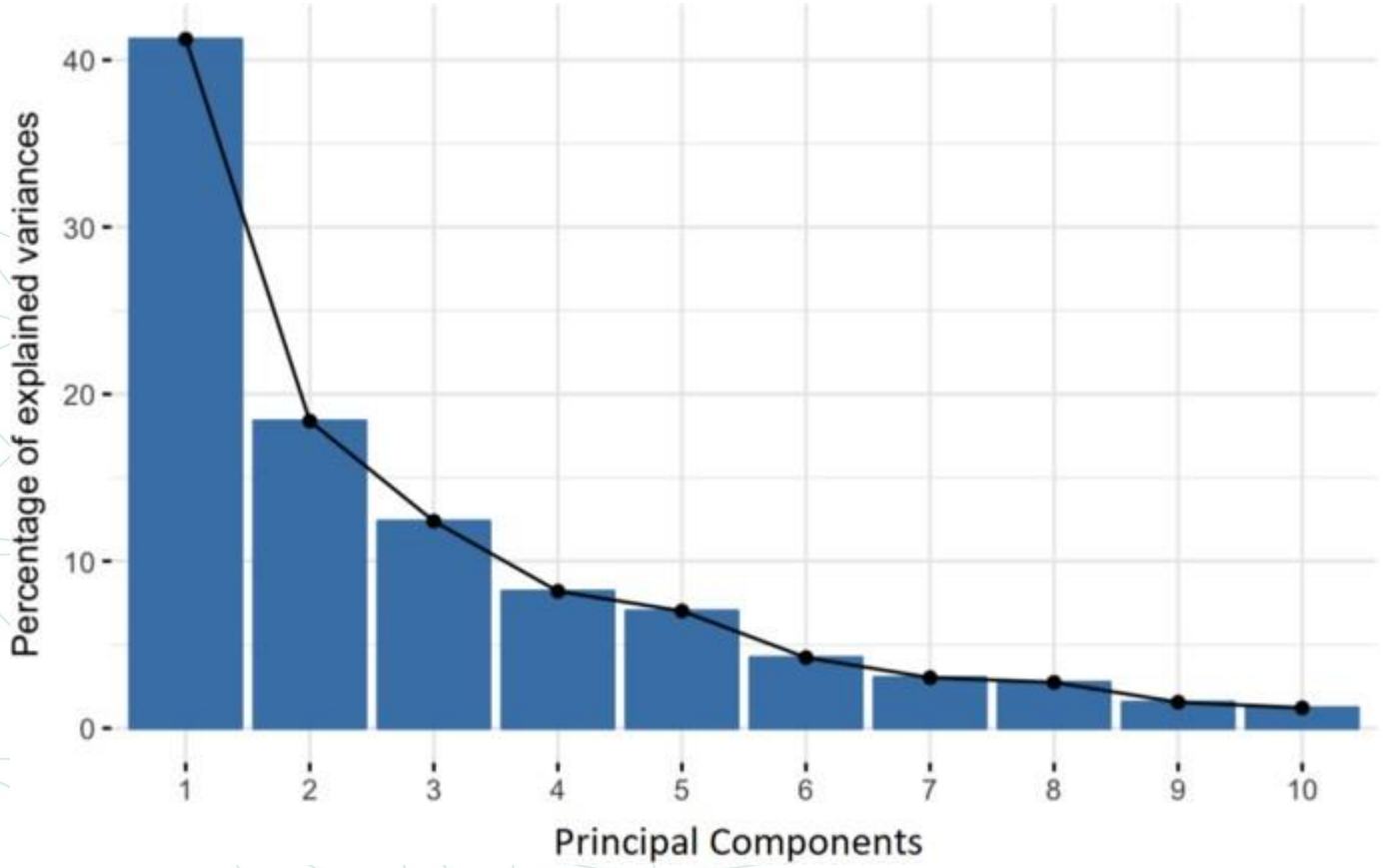
(4,2)

=

nf1	nf2
0.014003	0.755975
-2.556534	-0.780432
-0.051480	1.253135
1.014150	0.000239
1.579861	-1.228917

(5,2)

Percentage of Variance (Information) for each by PC



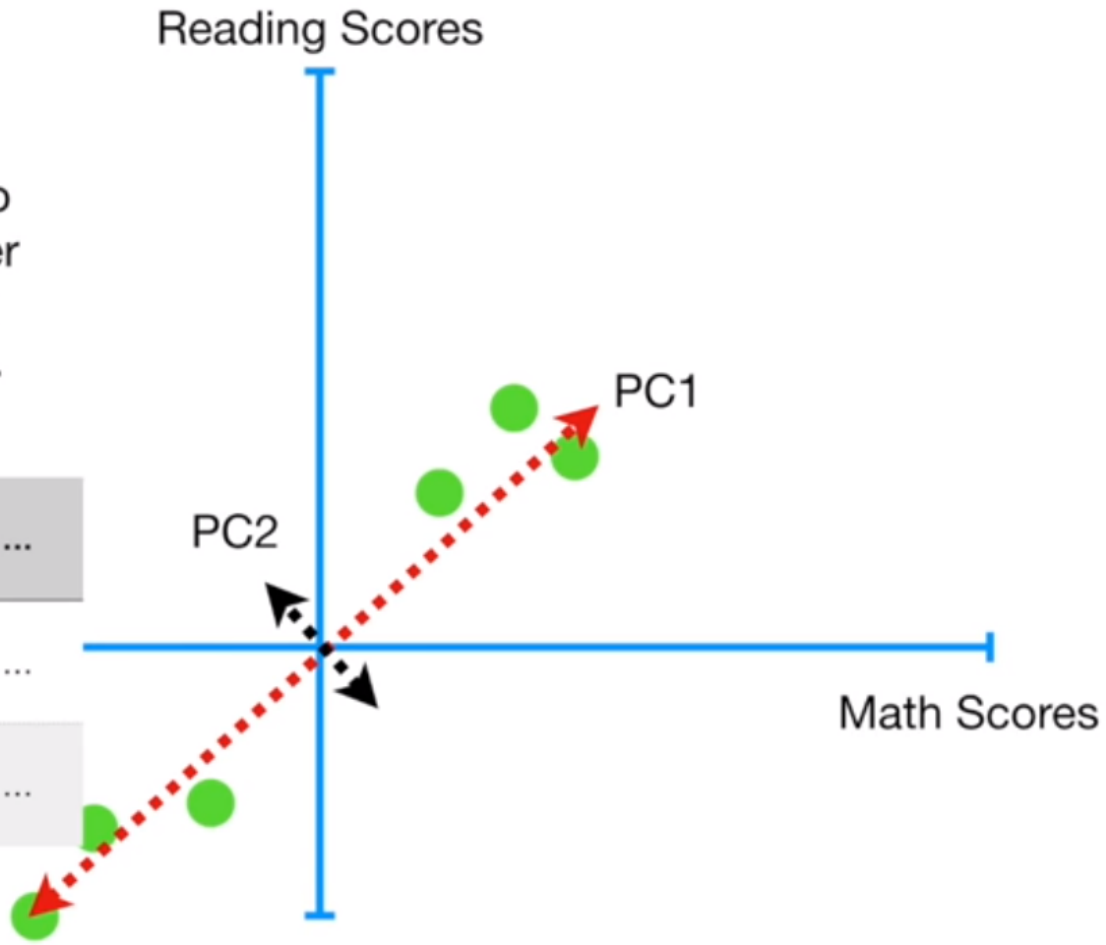
Practical Tip #1: Make sure the variables are on the same scale, and if not, scale them.

Practical Tip #2: Make sure your data is centered.

Practical Tip #3: How many principal components can you expect to find?

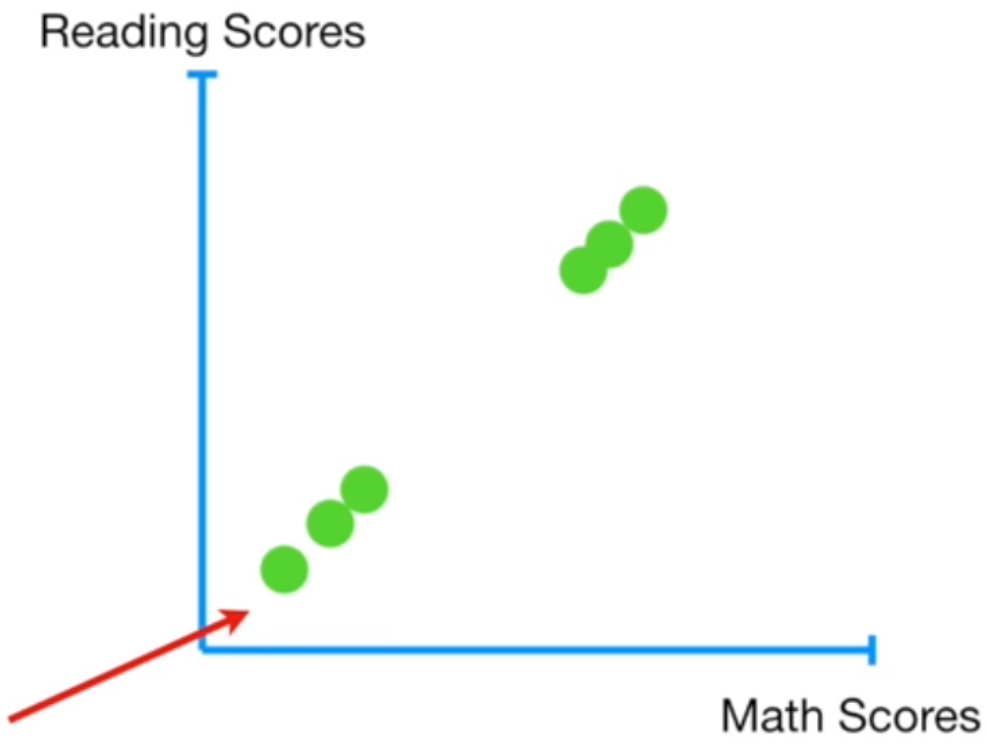
Thus, when we measure two things (math and reading) per sample (i.e. per student), at most we can have two PCs.

	Student 1	Student 2	Student 3	Student 4	...
Math	9.5	8.8	9.3	7.5	...
Reading	9	8	10	7	...

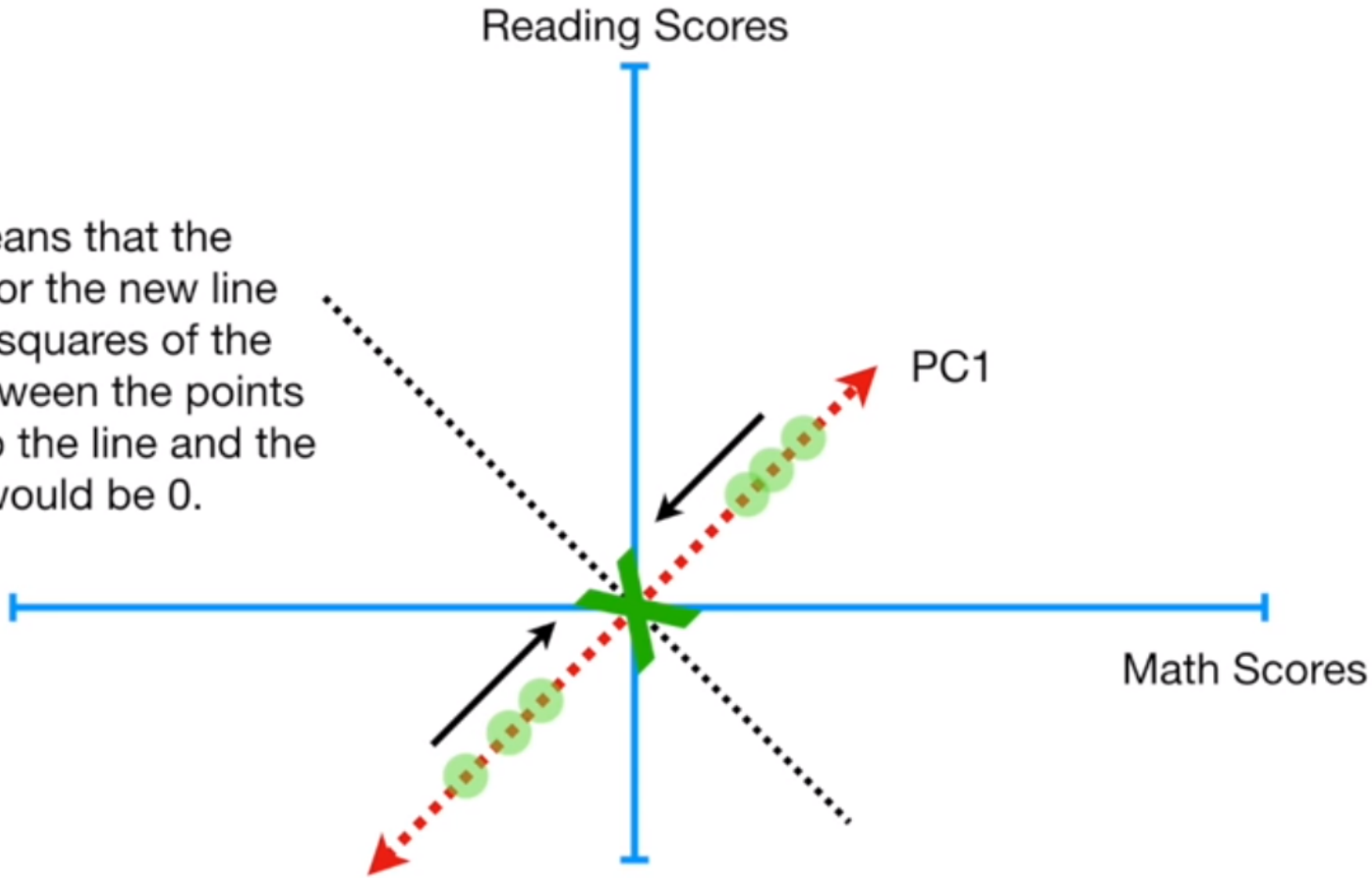


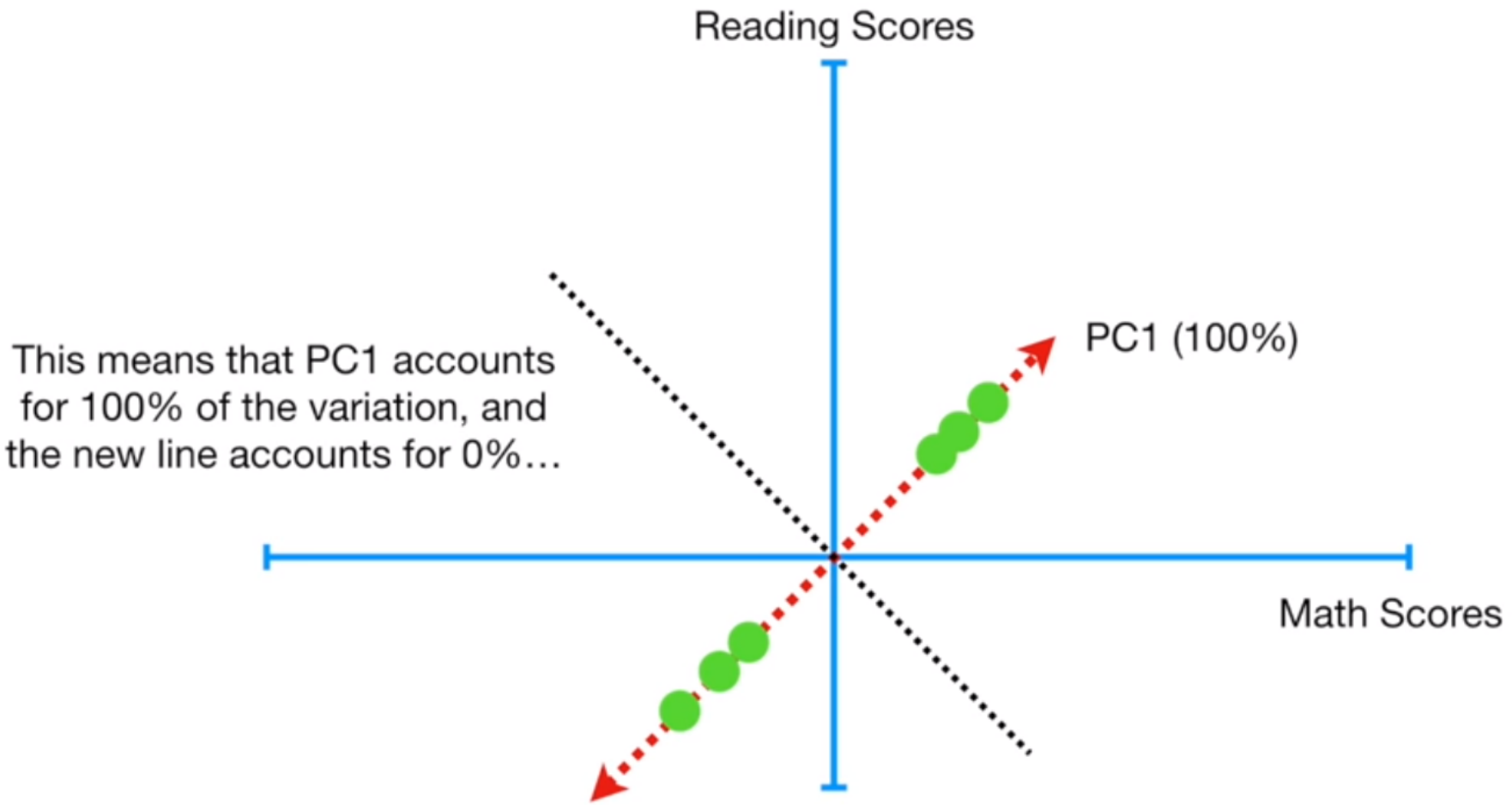
Now imagine that math and reading scores are 100% correlated…

	Student 1	Student 2	Student 3	Student 4	…
Math	9.5	8.8	9.3	7.5	…
Reading	9.5	8.8	9.3	7.5	…



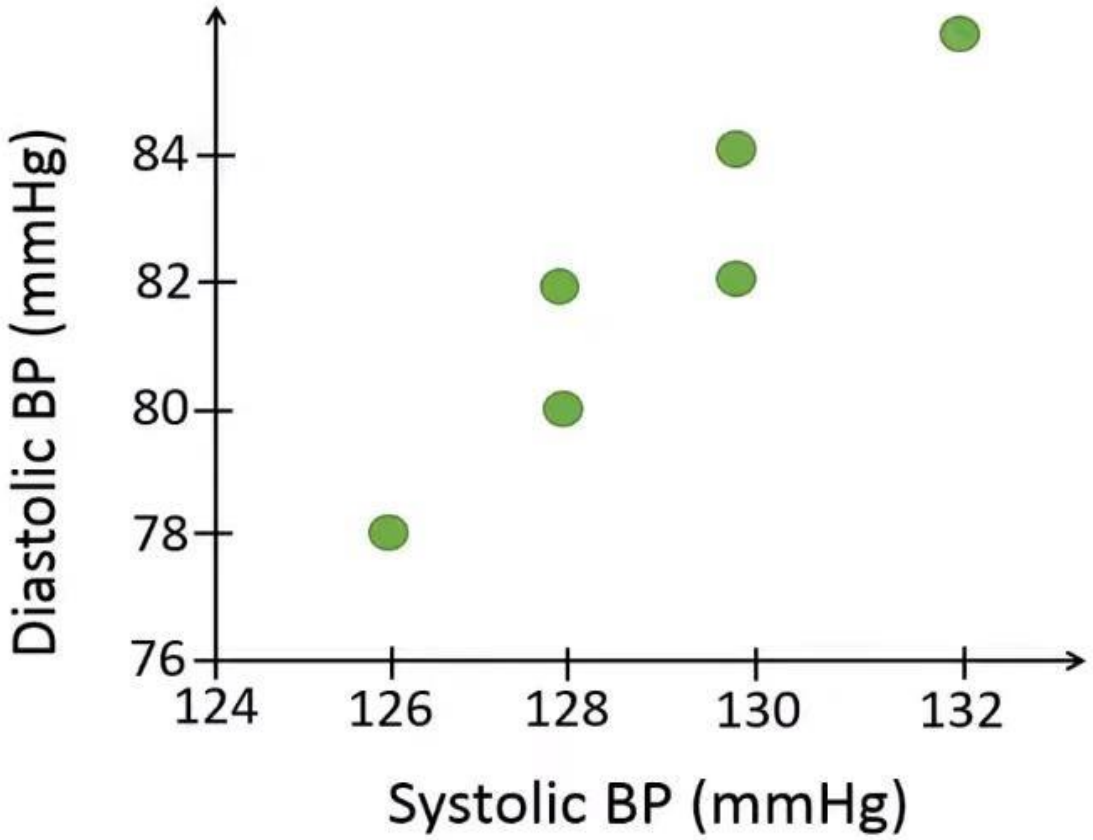
…This means that the eigenvalue for the new line (the sum of squares of the distances between the points projected onto the line and the origin), would be 0.



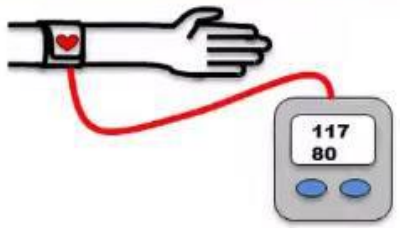

Reading Scores
This means that PC1 accounts for 100% of the variation, and the new line accounts for 0%…
PC1 (100%)
Math Scores

- PCA Lab

https://www.youtube.com/watch?v=S51bTyIwxFs



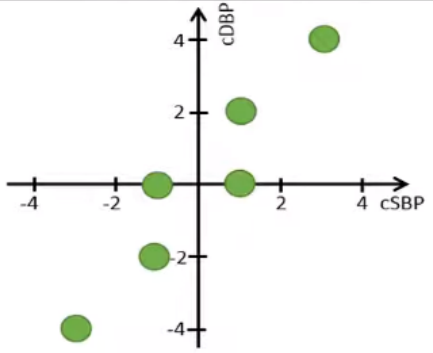
Systolic BP	Diastolic BP
126	78
128	80
128	82
130	82
130	84
132	86



To explain how the PCA works, we will use the following example data. We will use PCA to combine the two blood pressure variables into just one variable based on data from six individuals.

3. Calculate the eigenvalues of the covariance matrix

Centered SBP	Centered DBP
-3	-4
-1	-2
-1	0
1	0
1	2
3	4



$$\det\left|A-\lambda I\right|=0$$

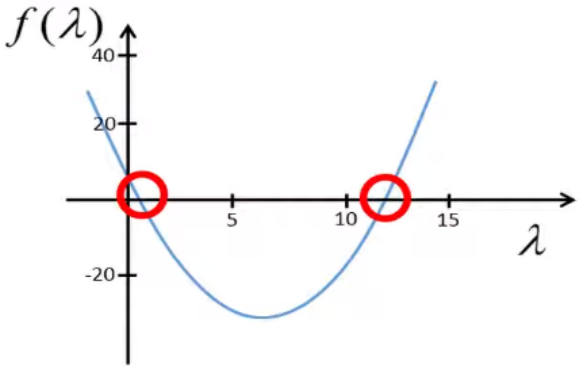
	SBP	DBP
SBP	4.4	5.6
DBP	5.6	8.0

$$\det\left|\begin{bmatrix}4.4 & 5.6\\ 5.6 & 8.0\end{bmatrix}-\begin{bmatrix}\lambda & 0\\ 0 & \lambda\end{bmatrix}\right|=0$$

We substitute A by the covariance matrix,

3. Calculate the eigenvalues of the covariance matrix

Centered SBP	Centered DBP
-3	-4
-1	-2
-1	0
1	0
1	2
3	4



	SBP	DBP
SBP	4.4	5.6
DBP	5.6	8.0

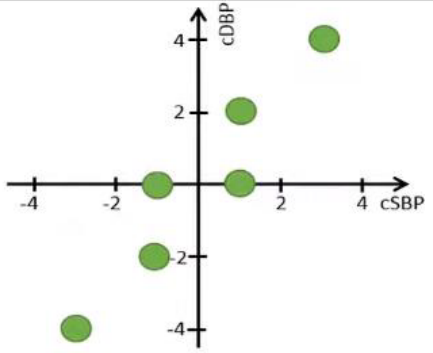
$$3.84 - 12.4\lambda + \lambda^2 = 0$$

$$\lambda_1 = 0.32 \qquad \lambda_2 = 12.08$$

This means that if we set lambda to either 0.32 or 12.08, the left-hand side of this equation will become equal to zero, or close to zero due to rounding effects in this example.

4. Calculate the eigenvectors of the covariance matrix

Centered SBP	Centered DBP
-3	-4
-1	-2
-1	0
1	0
1	2
3	4



$$A \cdot v = \lambda \cdot v$$

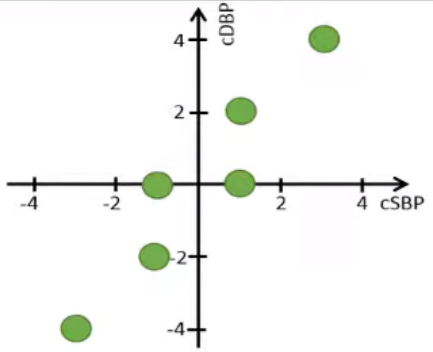
	SBP	DBP
SBP	4.4	5.6
DBP	5.6	8.0

$$\begin{bmatrix} 4.4 & 5.6 \\ 5.6 & 8.0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 12.08 \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

We plug in the covariance matrix,

4. Calculate the eigenvectors of the covariance matrix

Centered SBP	Centered DBP
-3	-4
-1	-2
-1	0
1	0
1	2
3	4



$$A \cdot v = \lambda \cdot v$$

	SBP	DBP
SBP	4.4	5.6
DBP	5.6	8.0

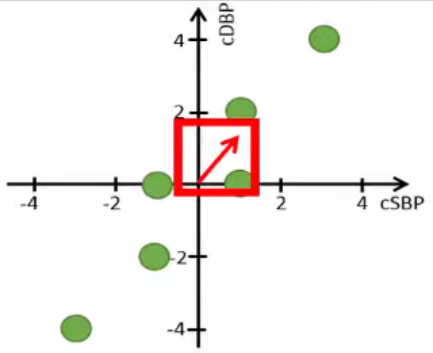
$$\begin{bmatrix} 4.4 & 5.6 \\ 5.6 & 8.0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = 12.08 \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$$4.4x + 5.6y = 12.08x$$
$$5.6x + 8.0y = 12.08y$$
$$5.6y = 7.68x$$
$$5.6x = 4.08y$$
$$y = 1.37x$$
$$1.37x = y$$

Solving for y in the two equations, results in that y is equal to 1.37 x.

4. Calculate the eigenvectors of the covariance matrix

Centered SBP	Centered DBP
-3	-4
-1	-2
-1	0
1	0
1	2
3	4



$$v_2 = \begin{bmatrix} 1 \\ 1.37 \end{bmatrix}$$

	SBP	DBP
SBP	4.4	5.6
DBP	5.6	8.0

We can illustrate this vector in the plot like this, by drawing an arrow from the origin to the coordinates,

Some thoughts about different classifiers: - both logistic regression and SVMs work great for linear problems, logistic regression may be preferable for very noisy data - naive Bayes can work better than logistic regression for small training sizes; also, it is pretty fast, e.g., if you have a large multi- class problem, you'd only have to train one classifier whereas you'd have to use One-vs-Rest or One-vs-One with in SVMs or logistic regression - kernel SVM for nonlinear data - k-nearest neighbour can also work quite - Random forests & Extra Trees work well on linear and nonlinear problems My favourites (from a theoretical standpoint) are Neural Networks, but I at least try a handful of different algorithm (e.g., see the above) when I build a model. Since this question is (impossible) to answer directly, let's maybe tackle the question from another angle and list the algorithms that are not so good: - clearly perceptron's: they are inferior to related classifiers such as adaptive linear neurons and logistic regression and there is no point in using them (okay, maybe speed can be a pro argument in very, very rare cases) - although k-nearest neighbours can be neat in certain cases where you have very functions that you want to approximate in a simple way, I wouldn't count it towards my "favourite" algorithms. Unless you have a very low-dimensional dataset you will likely suffer from the curse of dimensionality. - unless you need it for interpretability I would decision trees (unless you want to visualize the decision rules); Random Forests and Extra Trees are "better" Now, the same rules apply for regression, - Start with Ordinary least squares regression - use LASSO, Ridge, or elastic net regression if high variance (overfitting) or collinearity is a problem - try RANSAC if you have many outliers - try Support Vector Regression (with a nonlinear kernel) or Random Forest regression if you have nonlinear data

In [ ]:

Q&A

Questions and answers

Thanks