Introduction to Computation (CS2201)

Lecture 8

Kripabandhu Ghosh

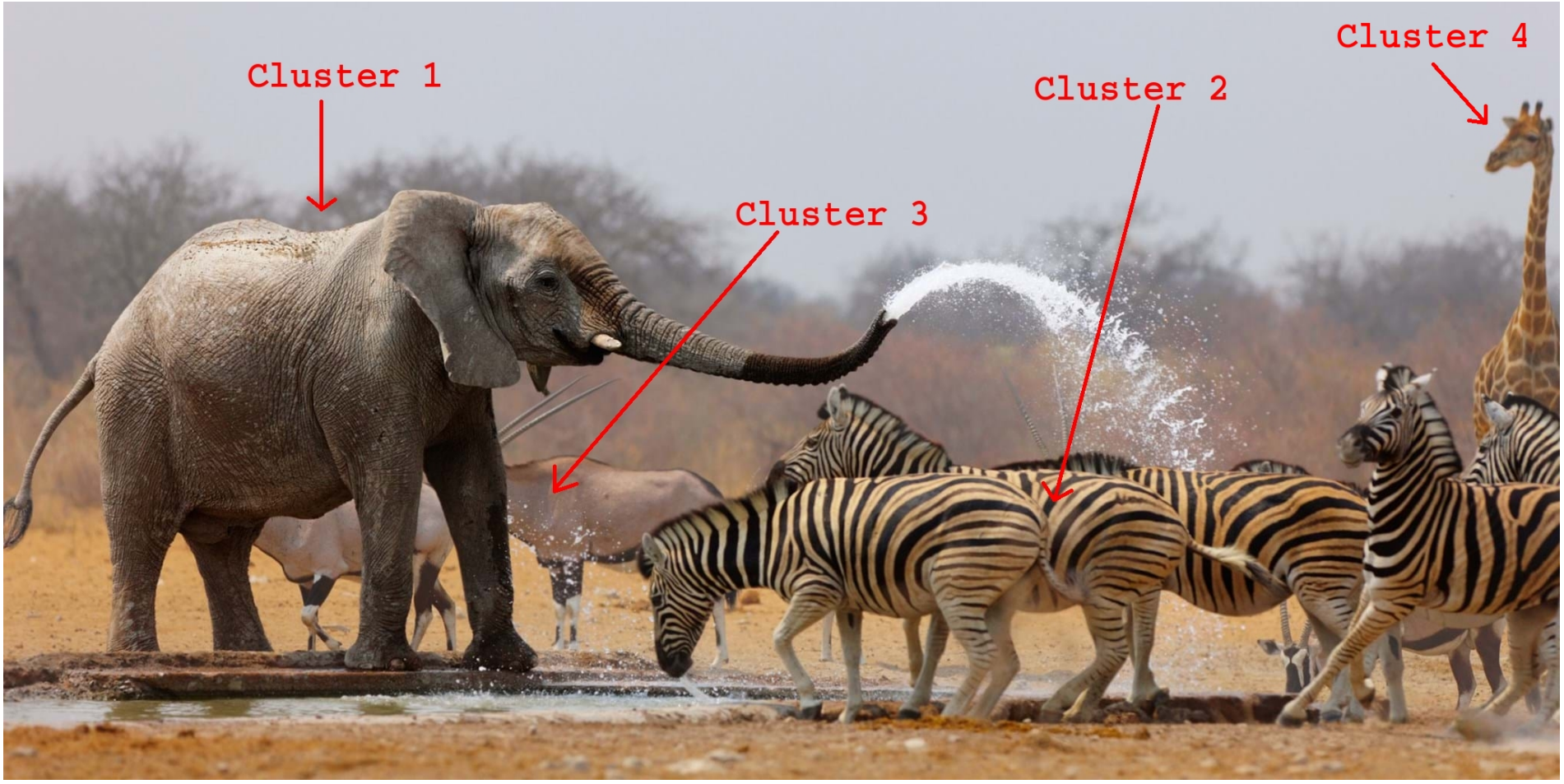
CDS, IISER Kolkata

CLUSTERING

Unknown World

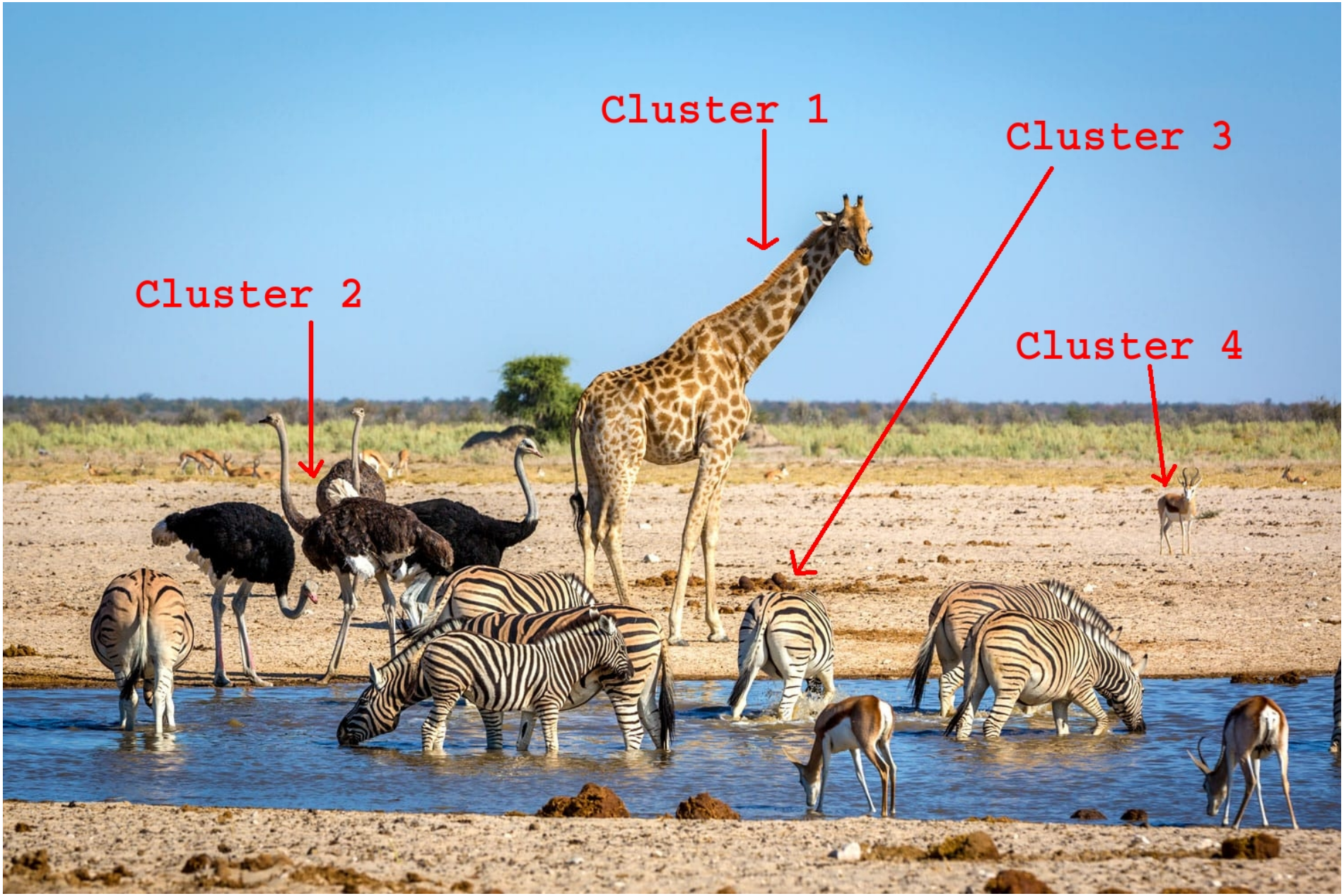
Unknown World

Unknown World



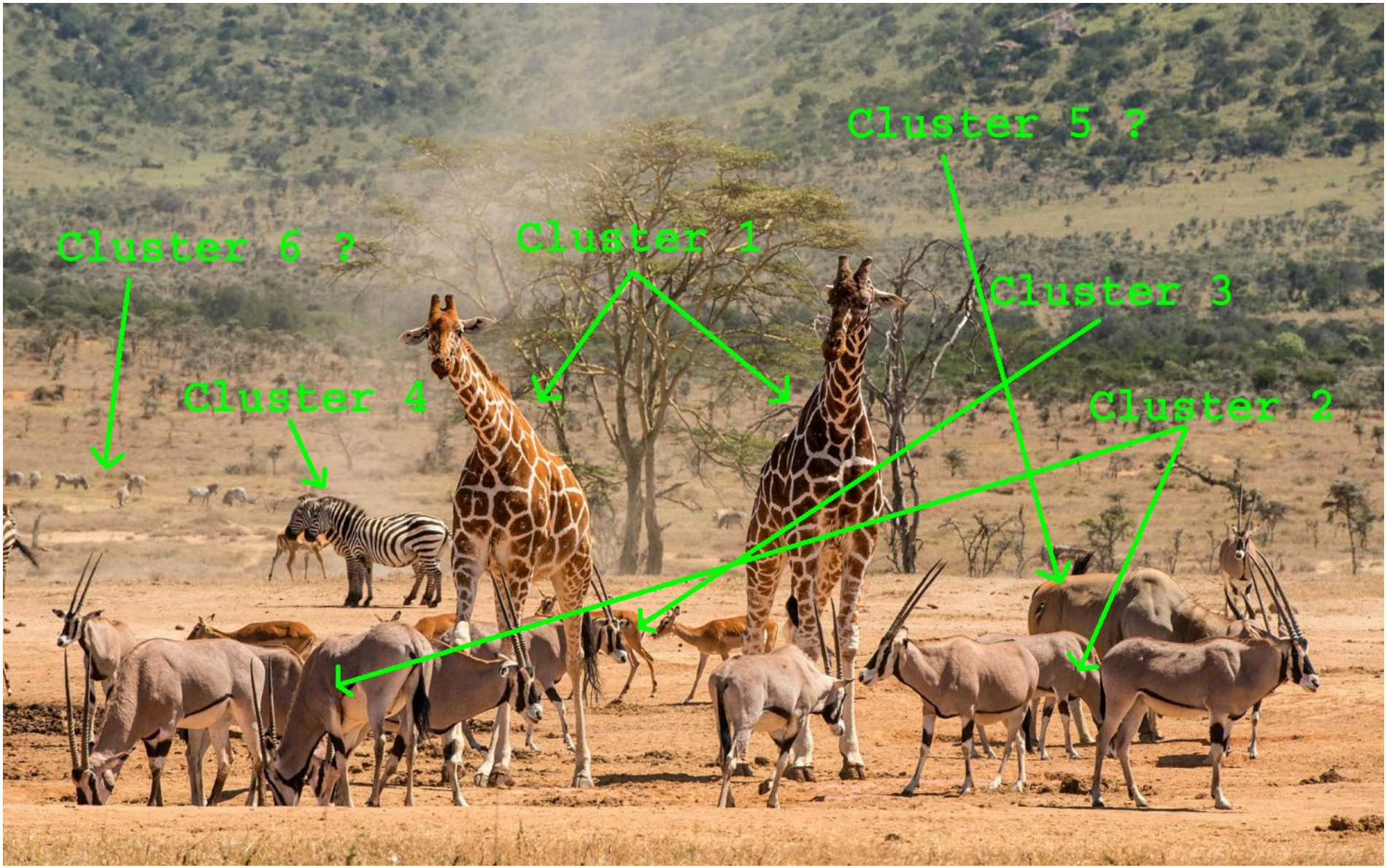
Unknown World

Unknown World



Unknown World

Unknown World

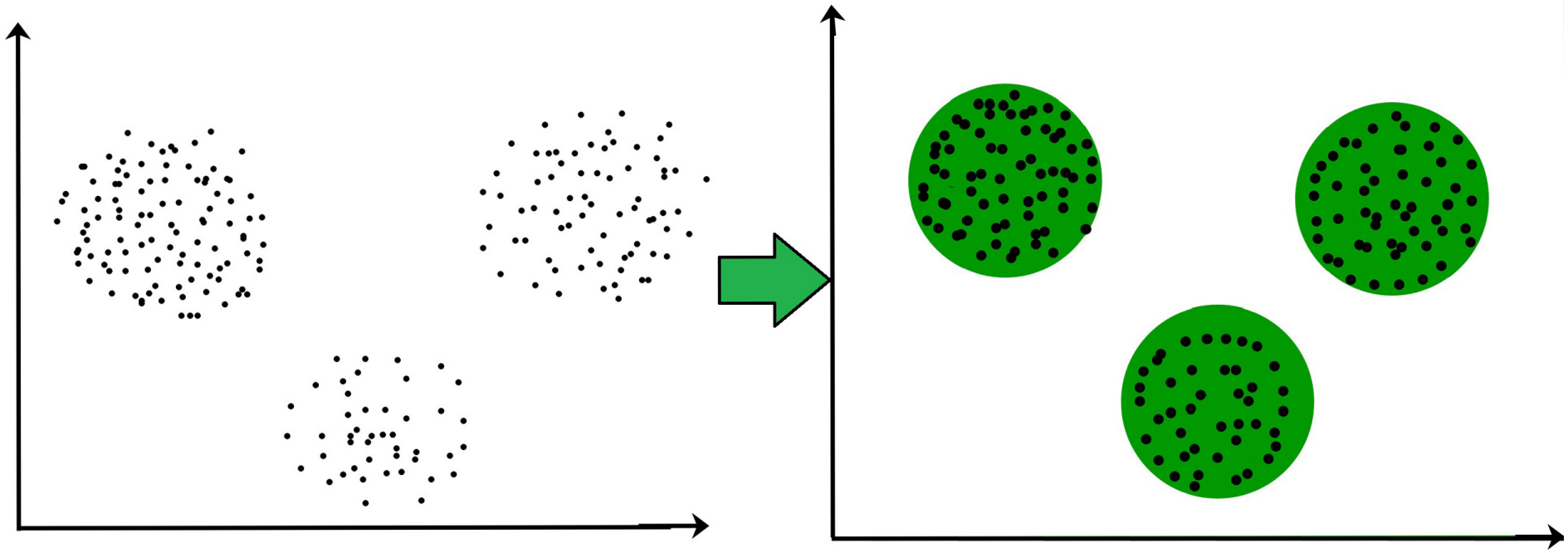


Clustering

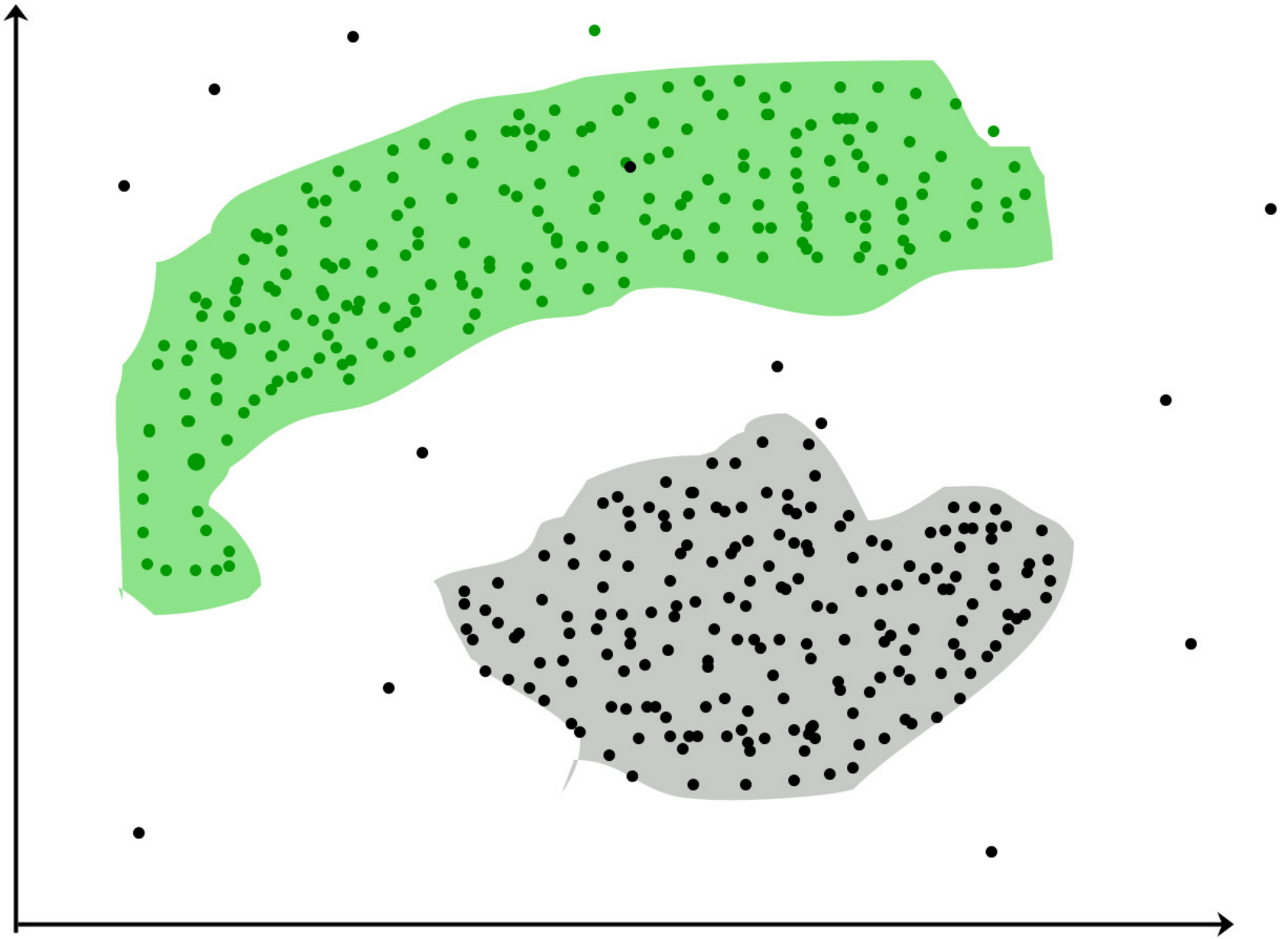
Task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).

- We might think of a cluster as comprising a group of data points whose inter-point distances are small compared with the distances to points outside of the cluster.
- Unsupervised method (no class label of any data point is known)

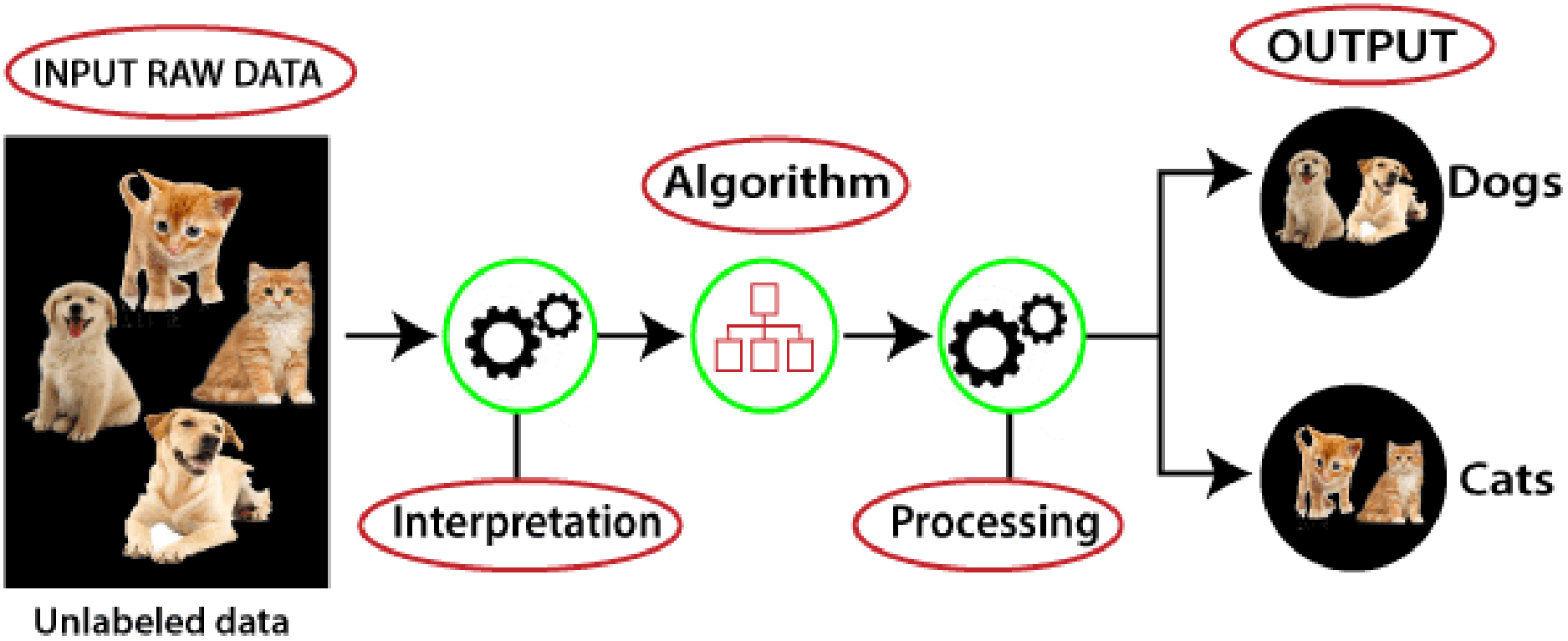
Clustering



Clustering



Clustering


INPUT RAW DATA
Algorithm
OUTPUT
Dogs
Cats
Interpretation
Processing
Unlabeled data

Clustering: application

Features

Definition

- individual measurable property or characteristic of a phenomenon
- The original input variables are typically preprocessed to transform them into some new space of variables
- A feature vector is an n-dimensional vector of numerical features that represent some object

Examples

- In spam detection algorithms, features may include the presence or absence of certain email headers, the email structure, the language, the frequency of specific terms, the grammatical correctness of the text.

Features

Features

Features

Feature vector

Animal	Trunk	Tusk	Stripes	Long neck
Elephant	1	1	0	0
Zebra	0	0	1	0
Giraffe	0	0	0	1

Feature vectors

- All Elephants represented as $\{1, 1, 0, 0\}$
- All Zebras represented as $\{0, 0, 1, 0\}$
- All Giraffes represented as $\{0, 0, 0, 1\}$

Distance between data points

Any distance measure like Euclidean distance

Are these good features?

Feature: Stripes?

Feature: Stripes?

More features

Colour, cat-like/horse-like features

Feature: Stripes?

More features

Colour, cat-like/horse-like features

Feature selection is a research problem

K-MEANS CLUSTERING

K-means Clustering

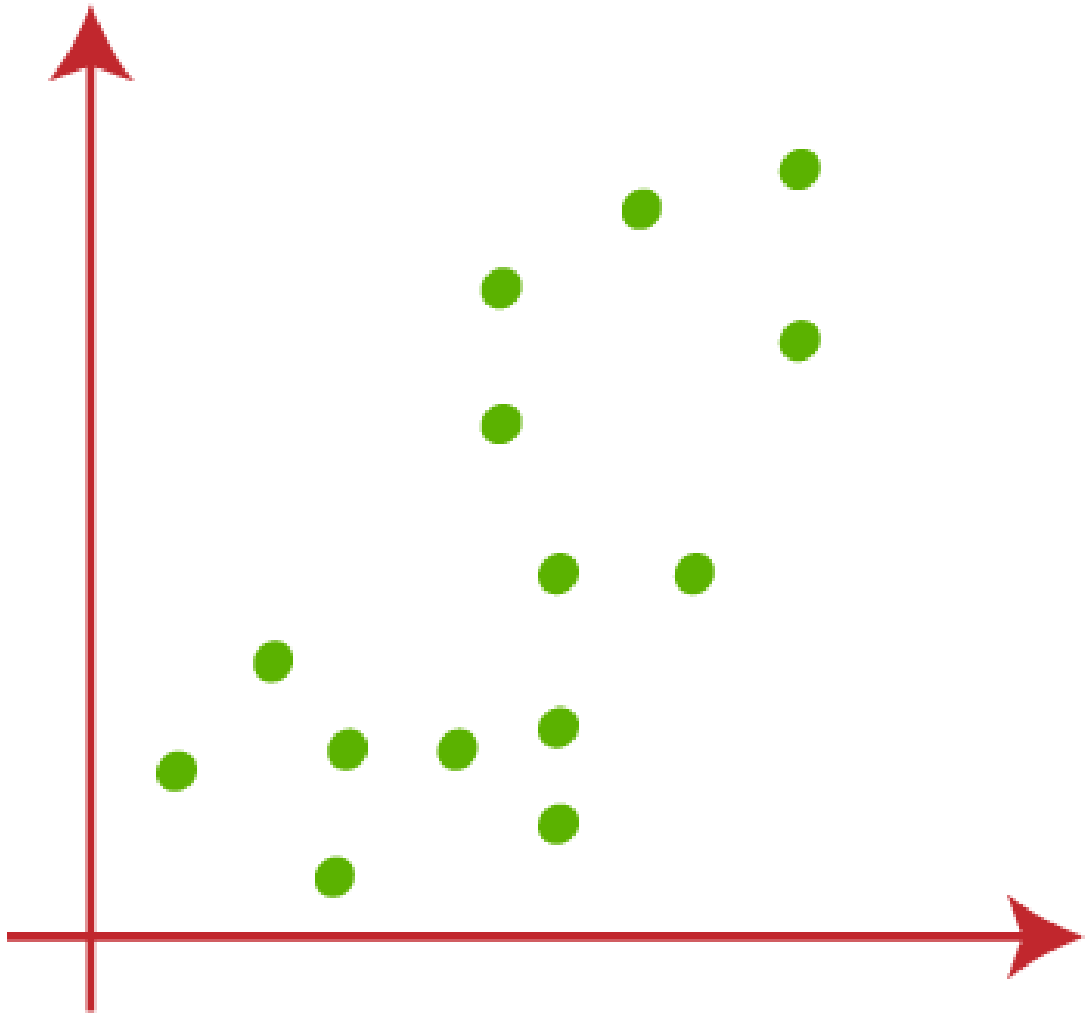
Premise

- Suppose we have a data set $\{x_1, ..., x_N\}$ consisting of N data points
- The goal is to partition the data set into some number K of clusters, where we shall suppose for the moment that the value of K is given

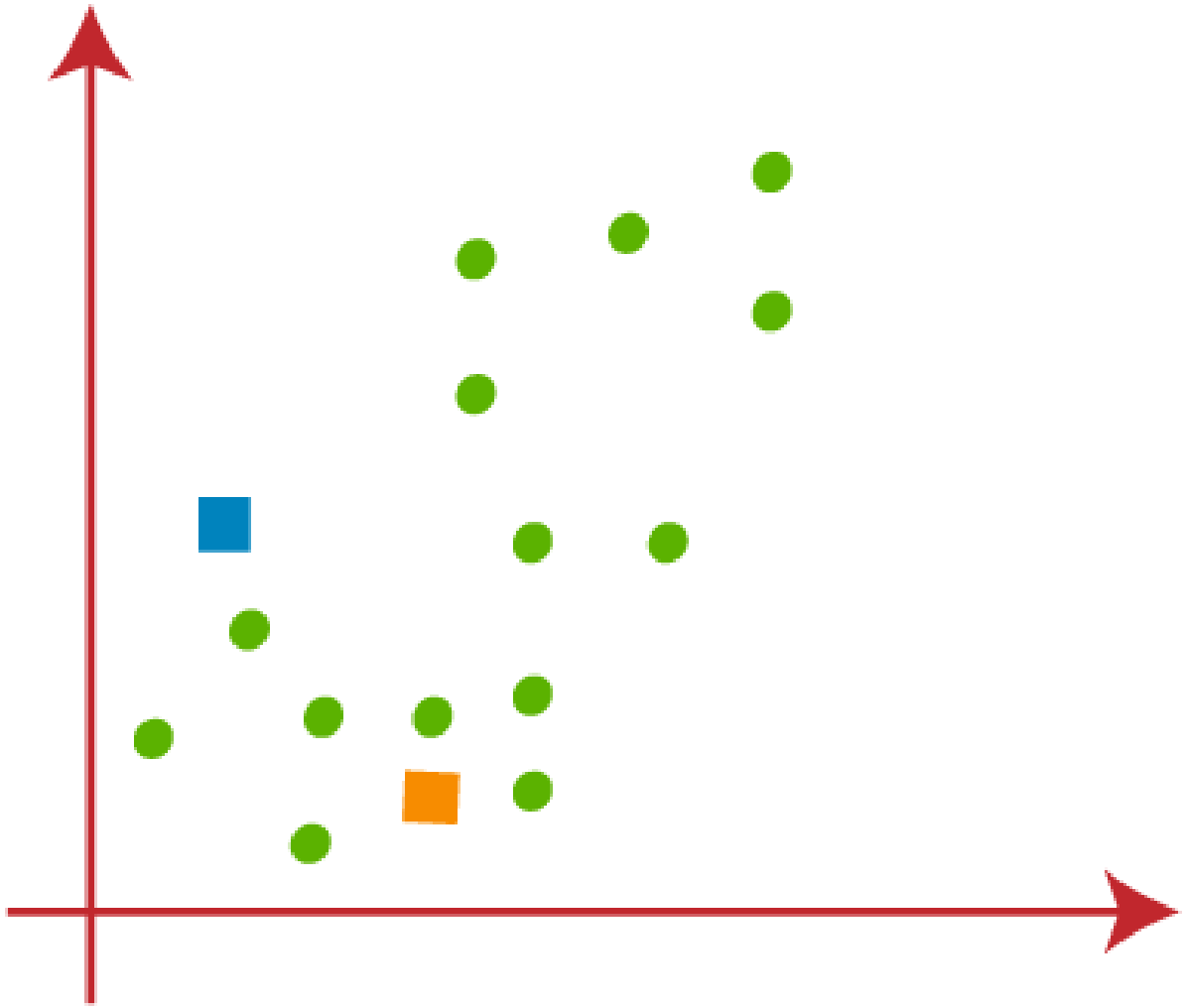
Algorithm

1. Select random K points or centroids.
2. Assign each data point to their closest centroid, which will form the predefined K clusters.
3. Compute a new centroid for each cluster
4. Repeat Step 2
5. If any reassignment occurs, then go to step-3 else END.

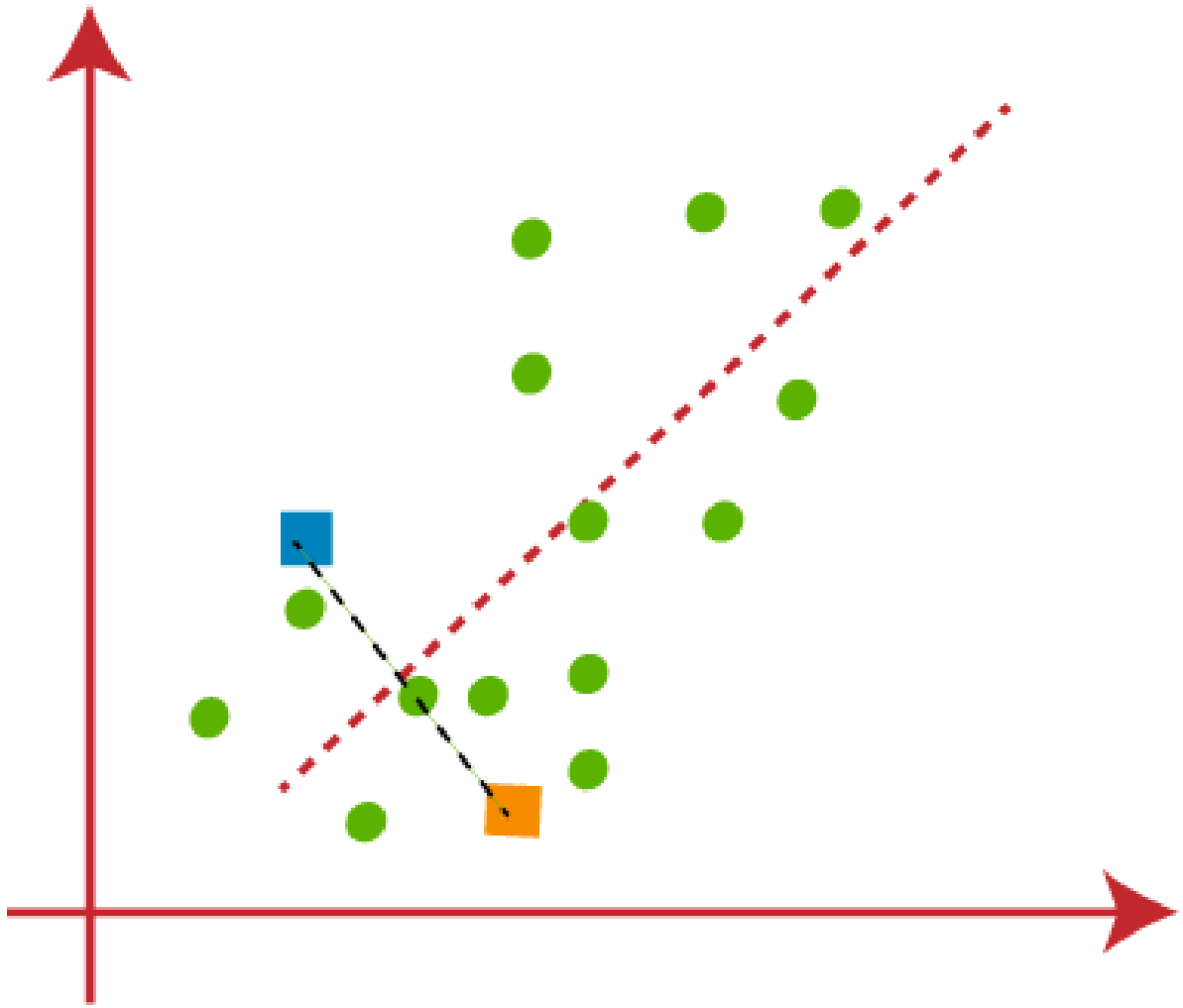
K-means: data points



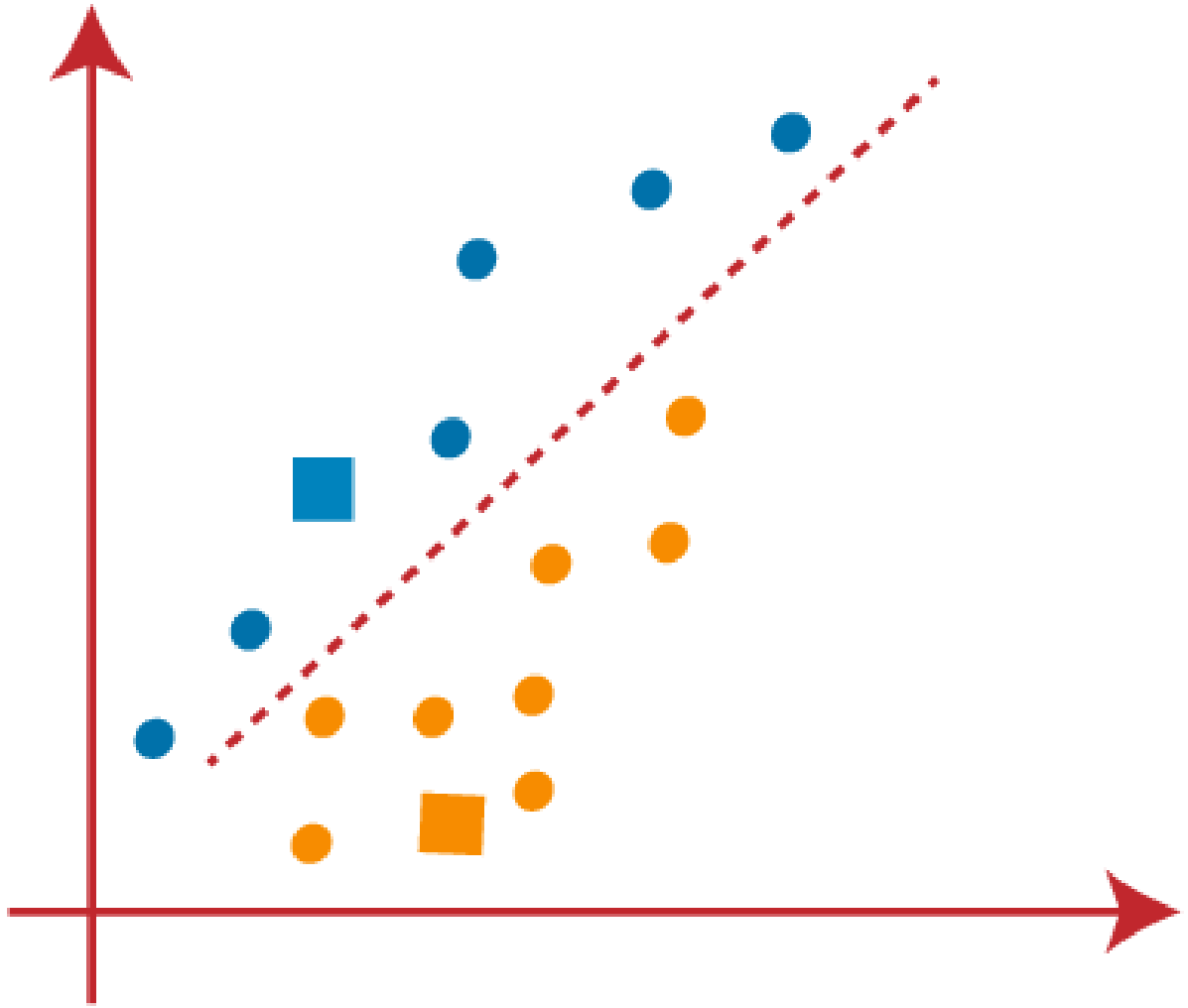
K-means: random centroids (K=2)



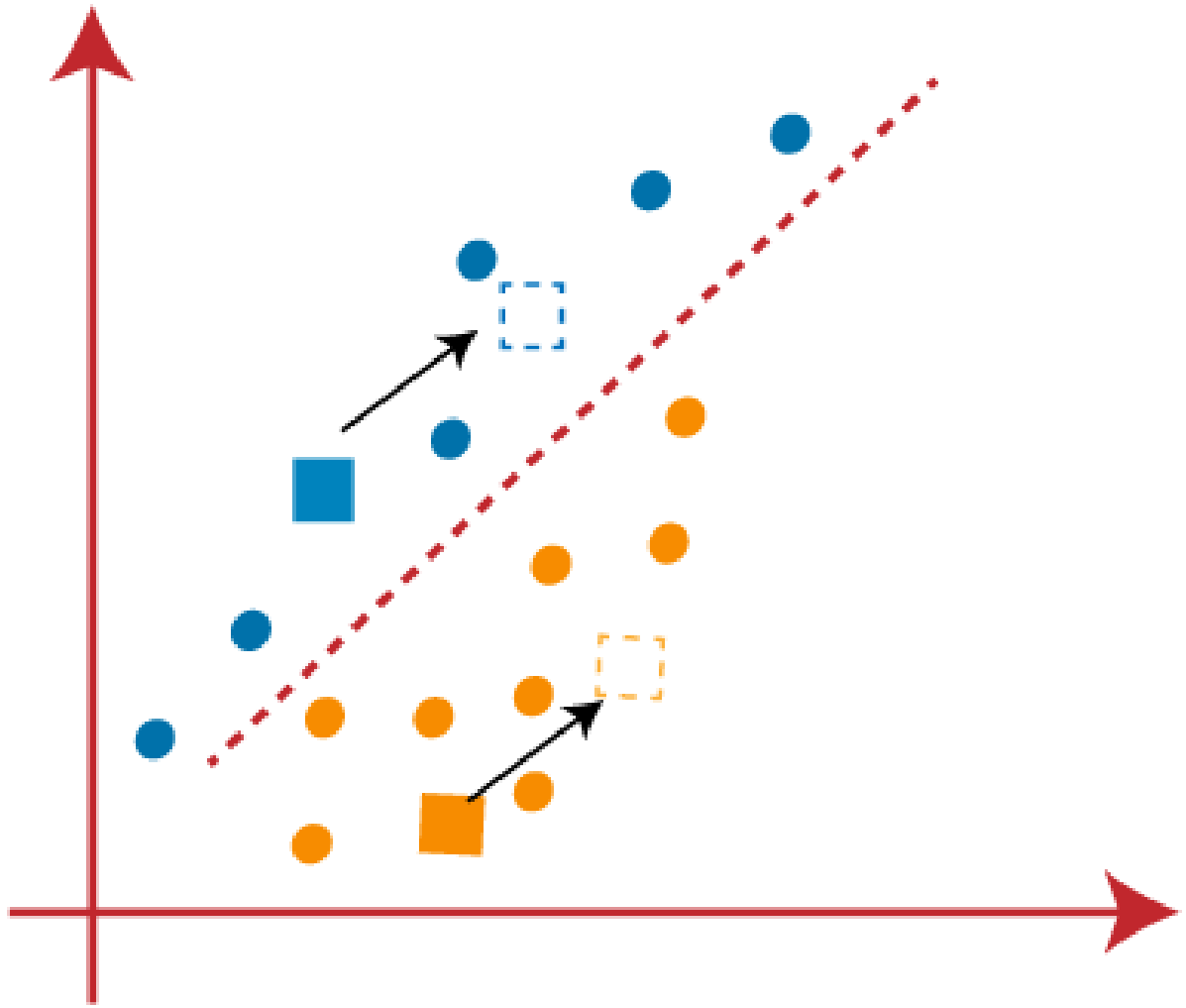
K-means: distance from centroids (K=2)



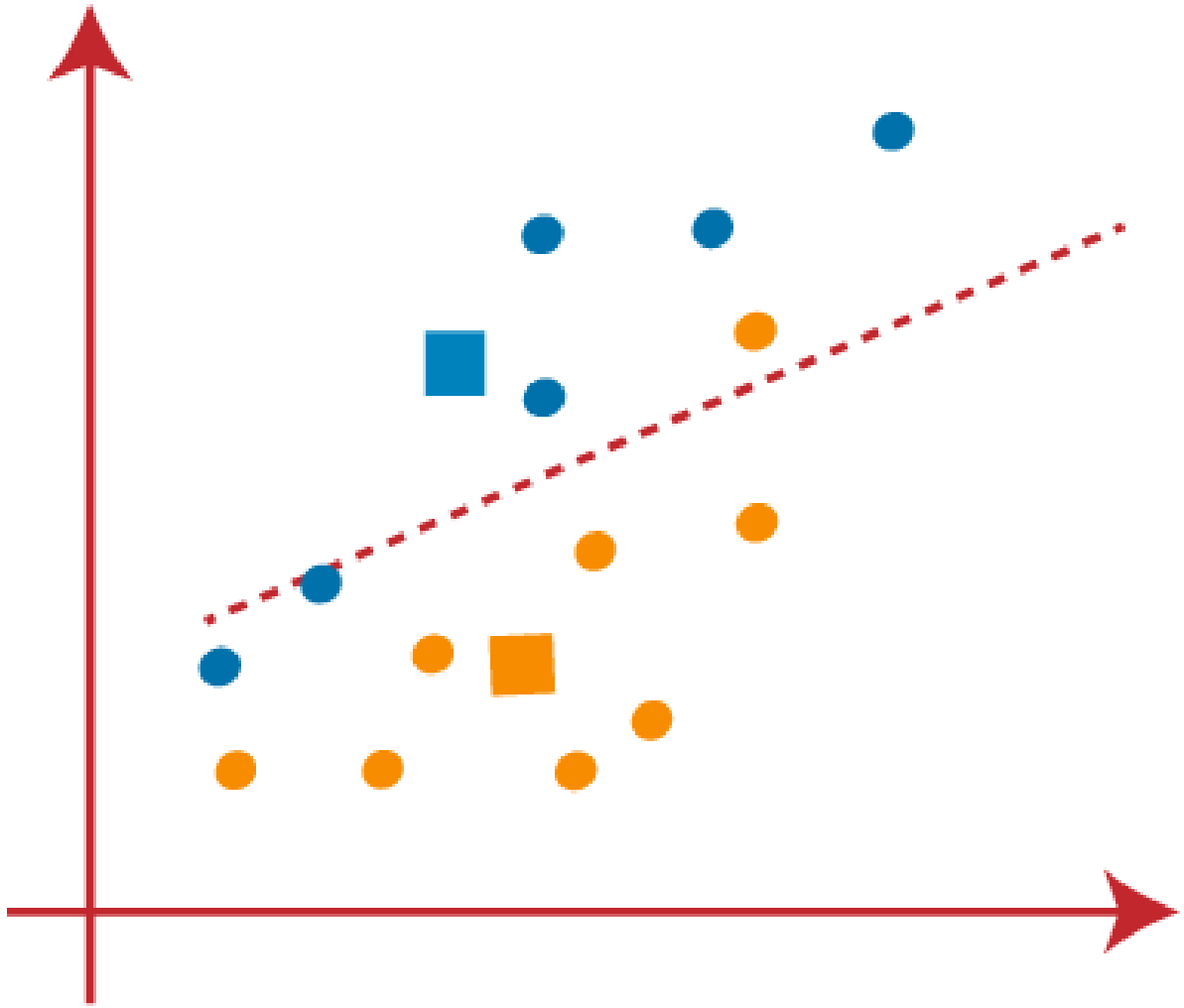
K-means: centroids based separation (K=2)



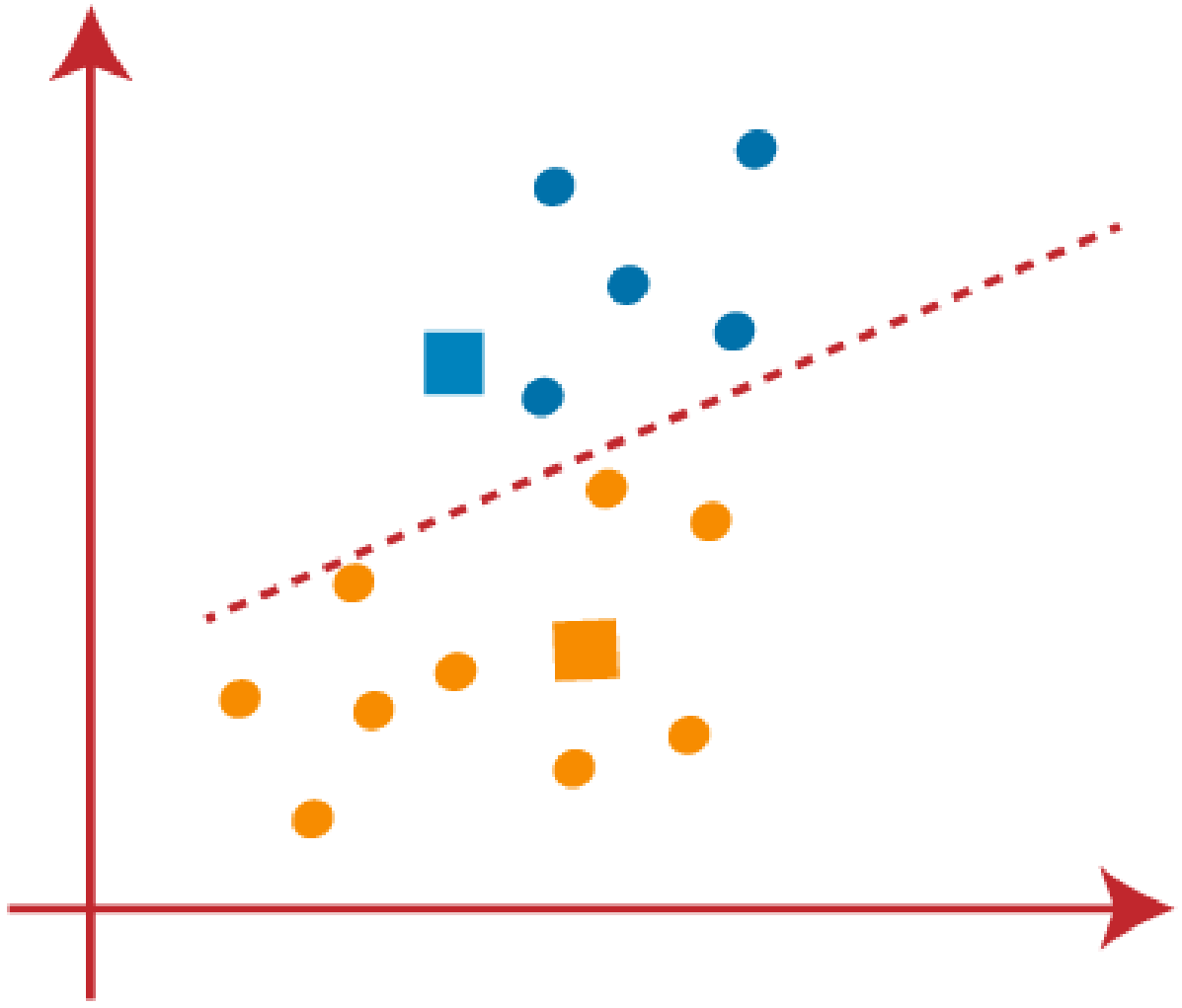
K-means: centroid recomputation (K=2)



K-means: recomputation of distance from centroids (K=2)

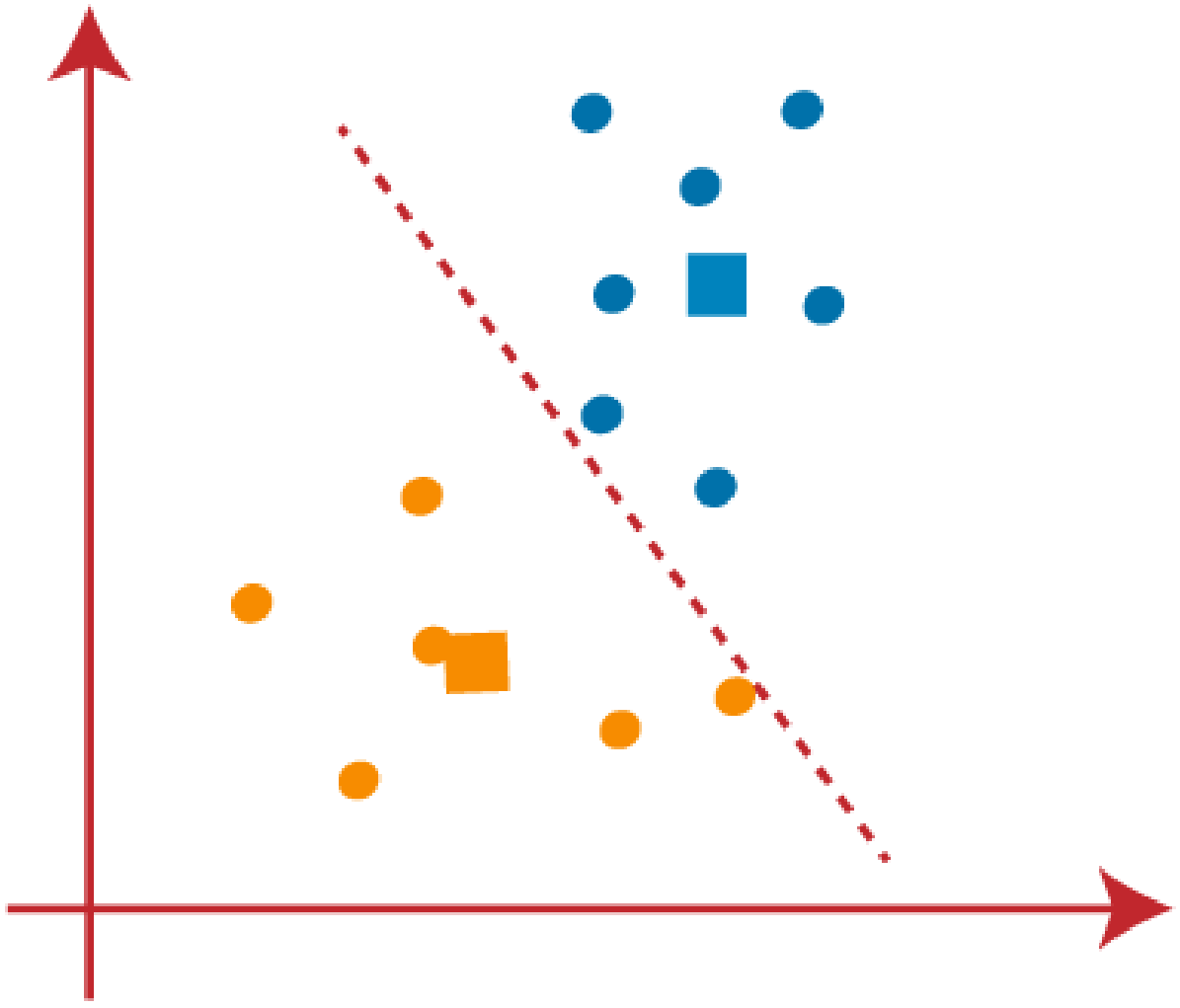


K-means: separation of data points (K=2)

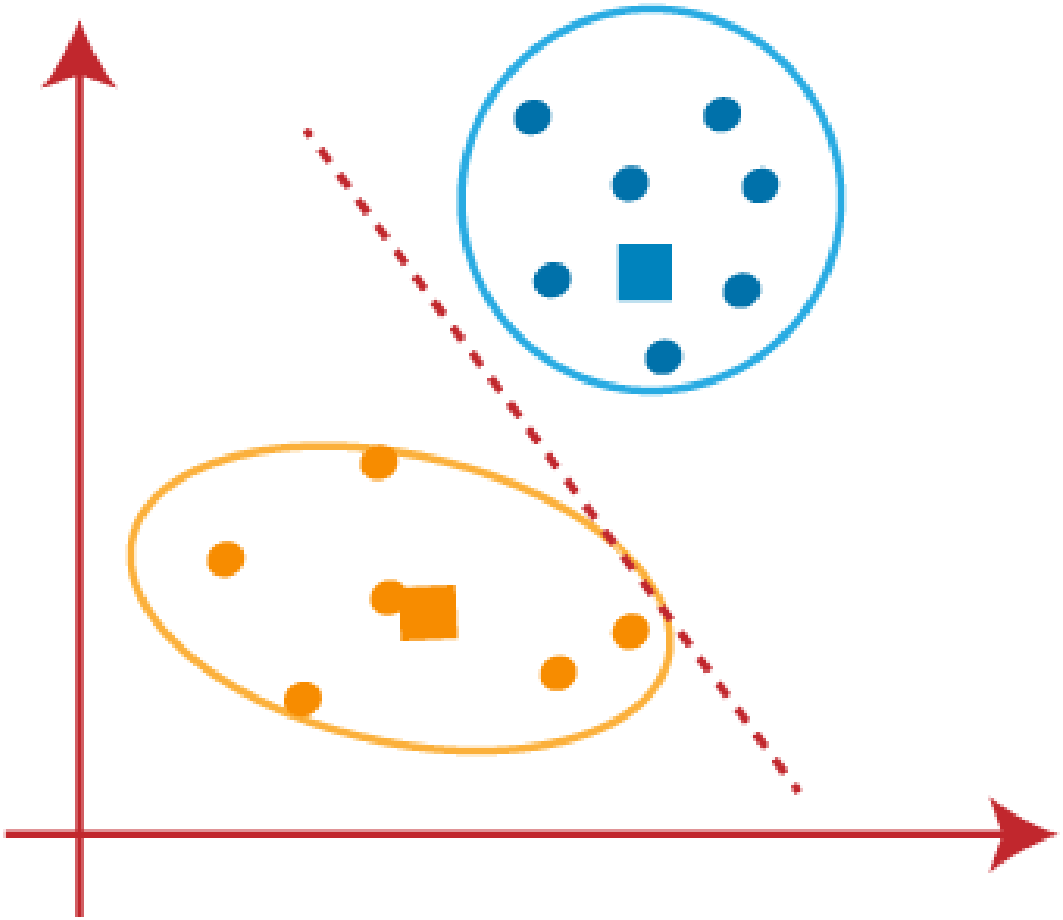


K-means: recomputation of centroids (K=2)

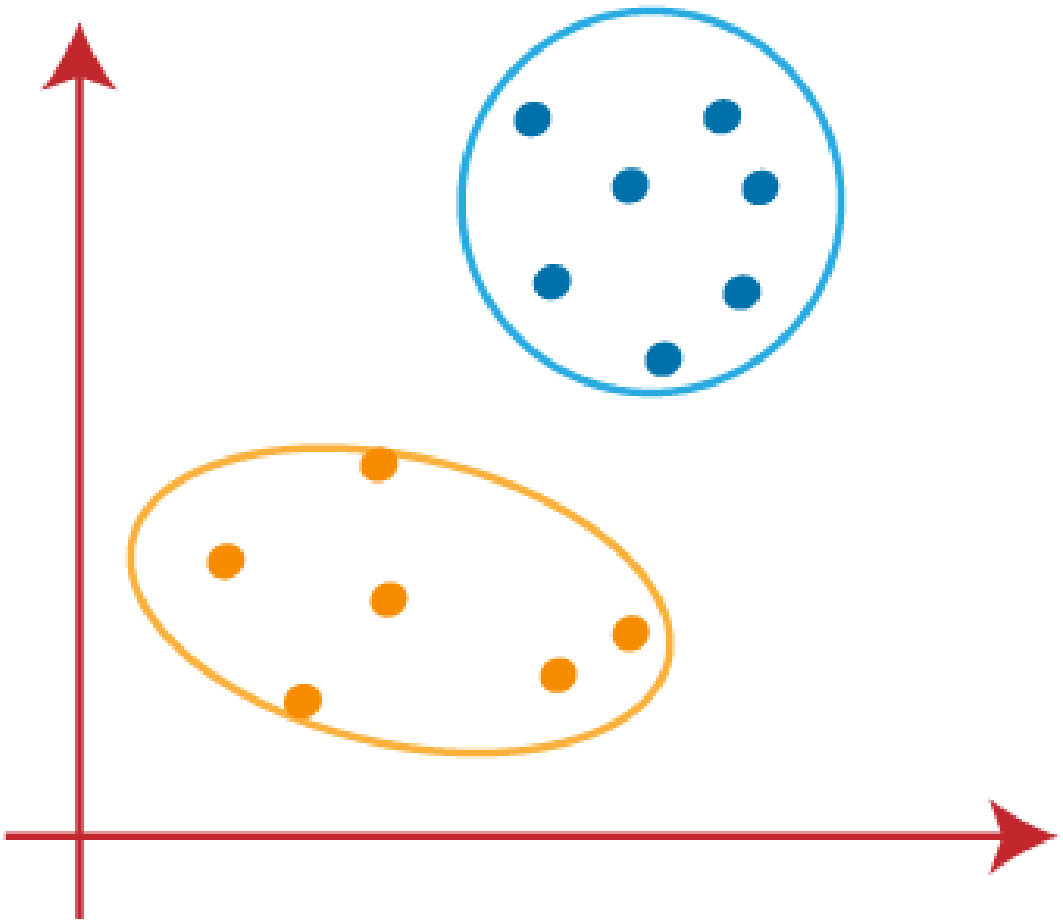
K-means: separation of data points (K=2)



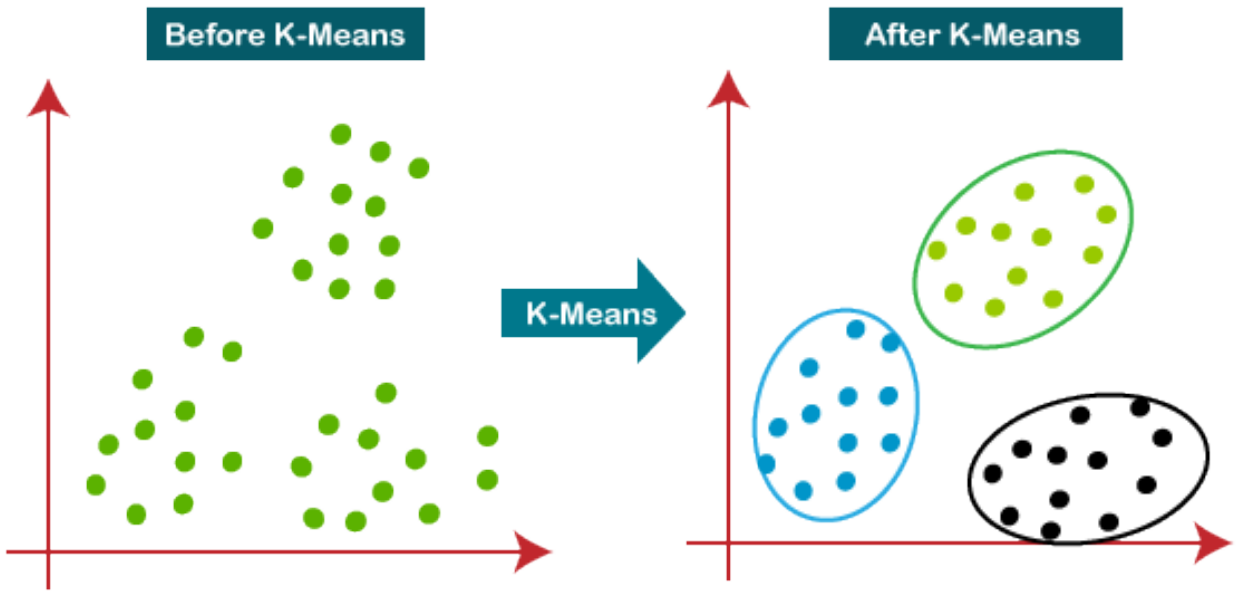
K-means: cluster formation (K=2)



K-means: cluster formation finalized (K=2)



K-means: cluster formation finalized (K=2)[1]



[1] https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning

K-means: elbow method

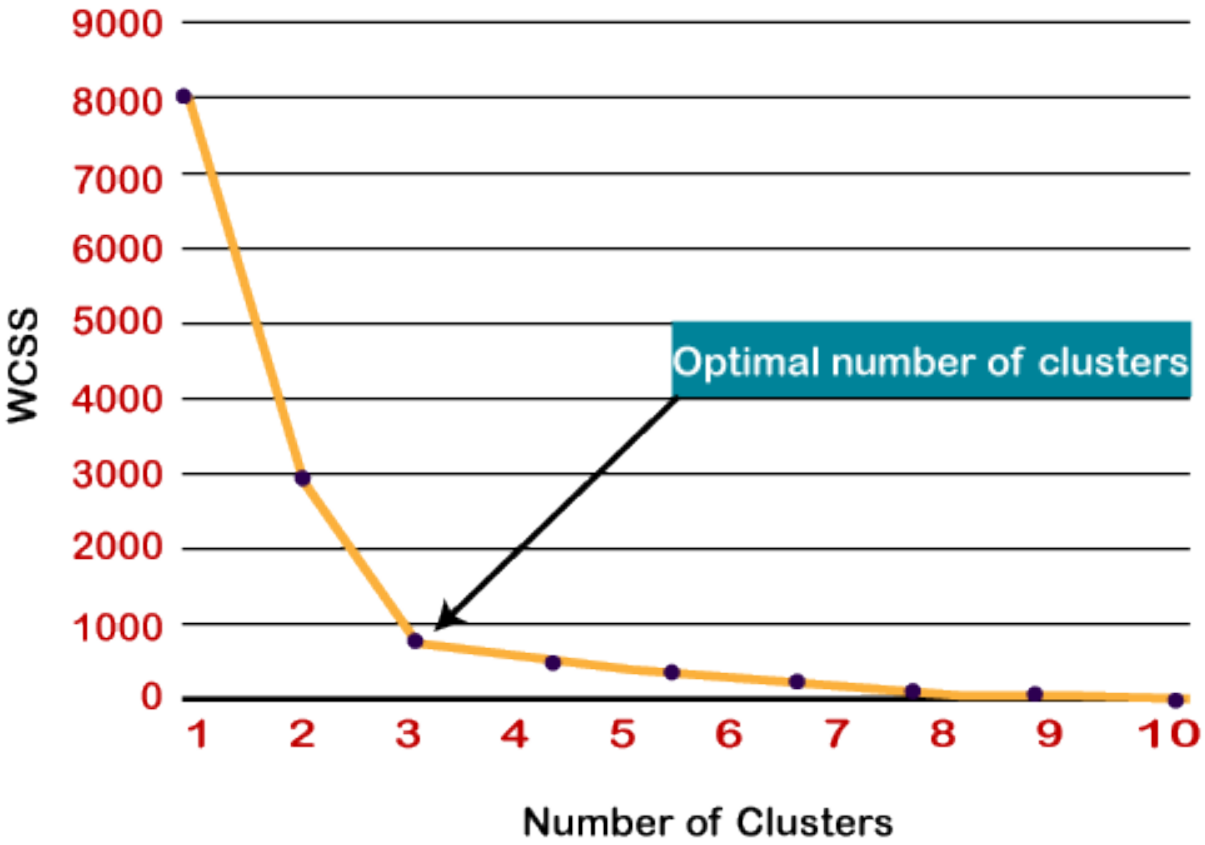
Within Cluster Sum of Squares (WCSS)

$$\text{WCSS} = \sum_{Cluster_i} \sum_{P_i \in Cluster_i} \text{distance}(P_i\ C_i)^2$$

- P_i: a point in a cluster
- C_i: Centroid in cluster i

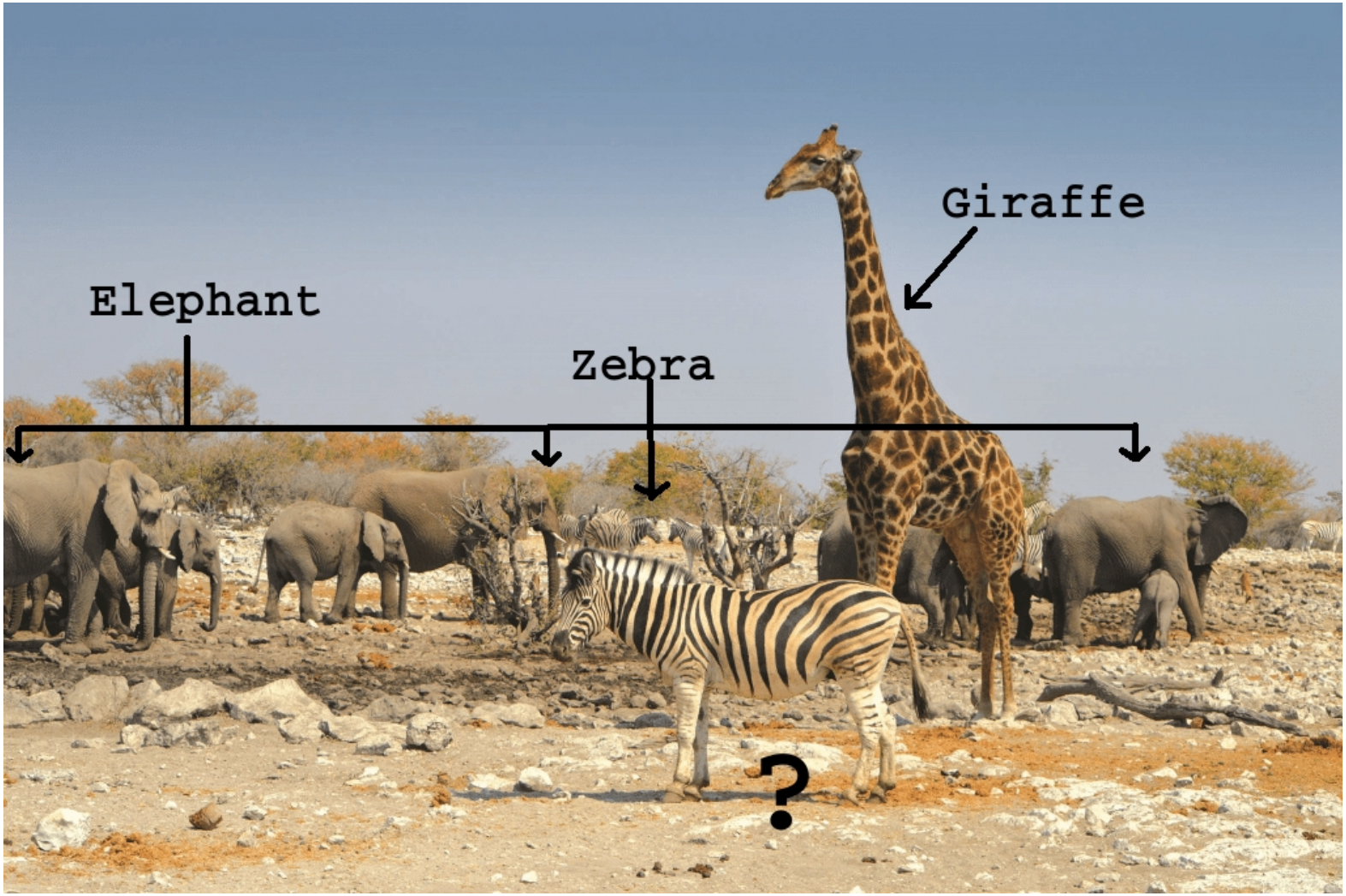
Minimize WCSS and choose the corresponding K value

K-means: elbow method



CLASSIFICATION

Classification Example



Classification

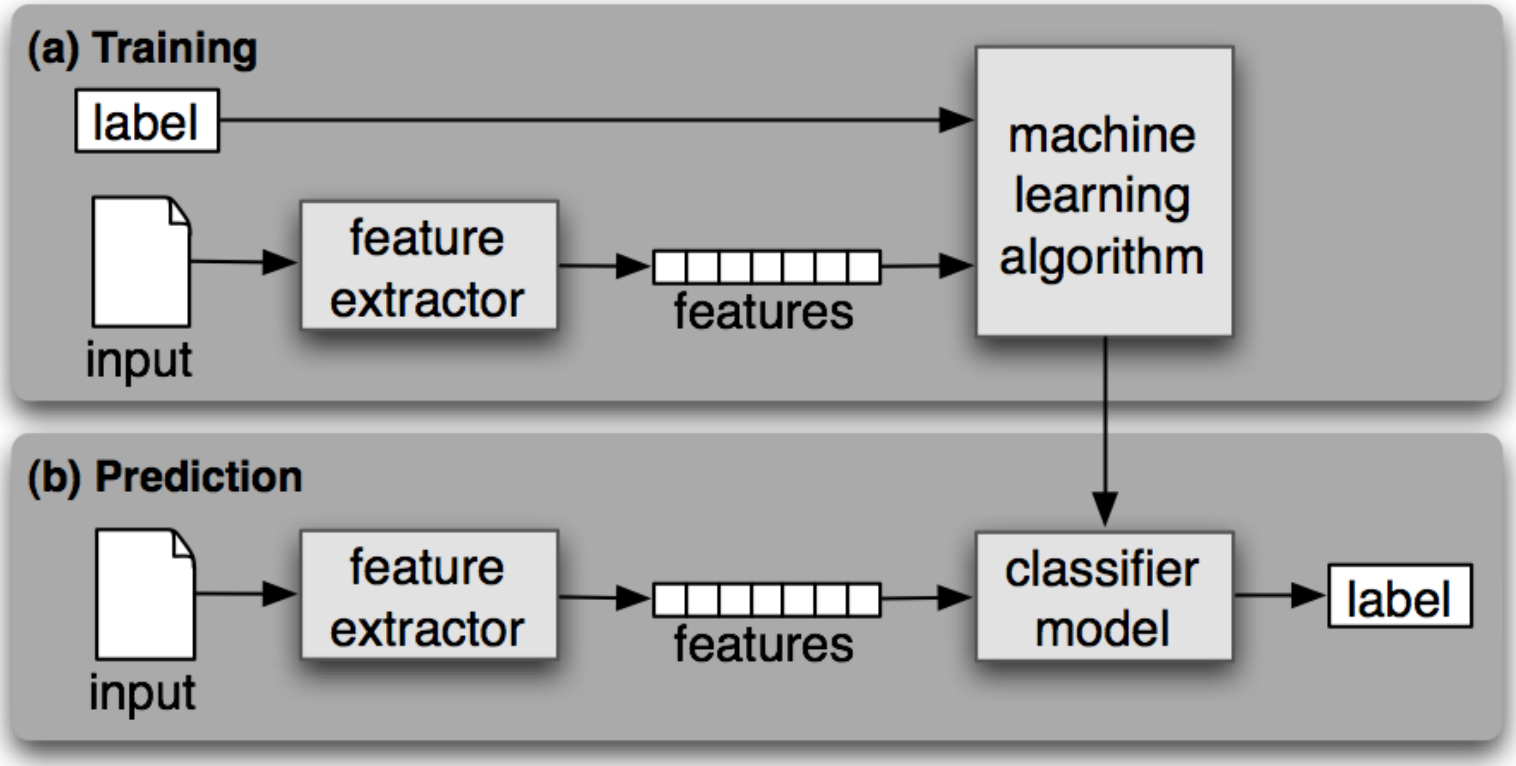
Definition

An applications in which the *training data* comprises examples of the input vectors (known as supervised learning problems) and the aim is to assign each input vector to one of a finite number of discrete categories (classes) is called a classification problem

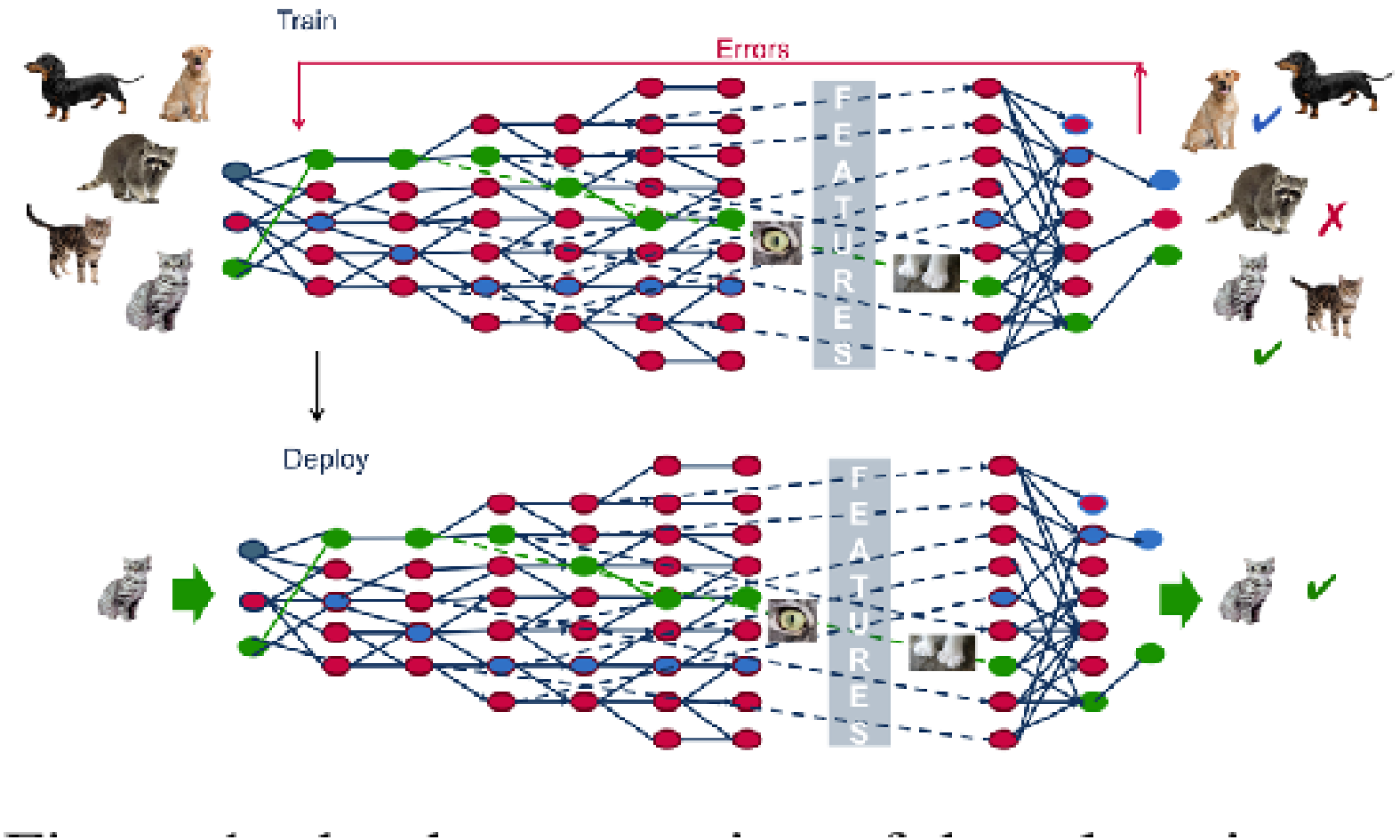
Phase

- Training: Model *learns* from data points and labels (e.g. animal image and animal class)
- Test: Given a new image (not a part of training) predicts the label (e.g. given an animal image, predicts its class)

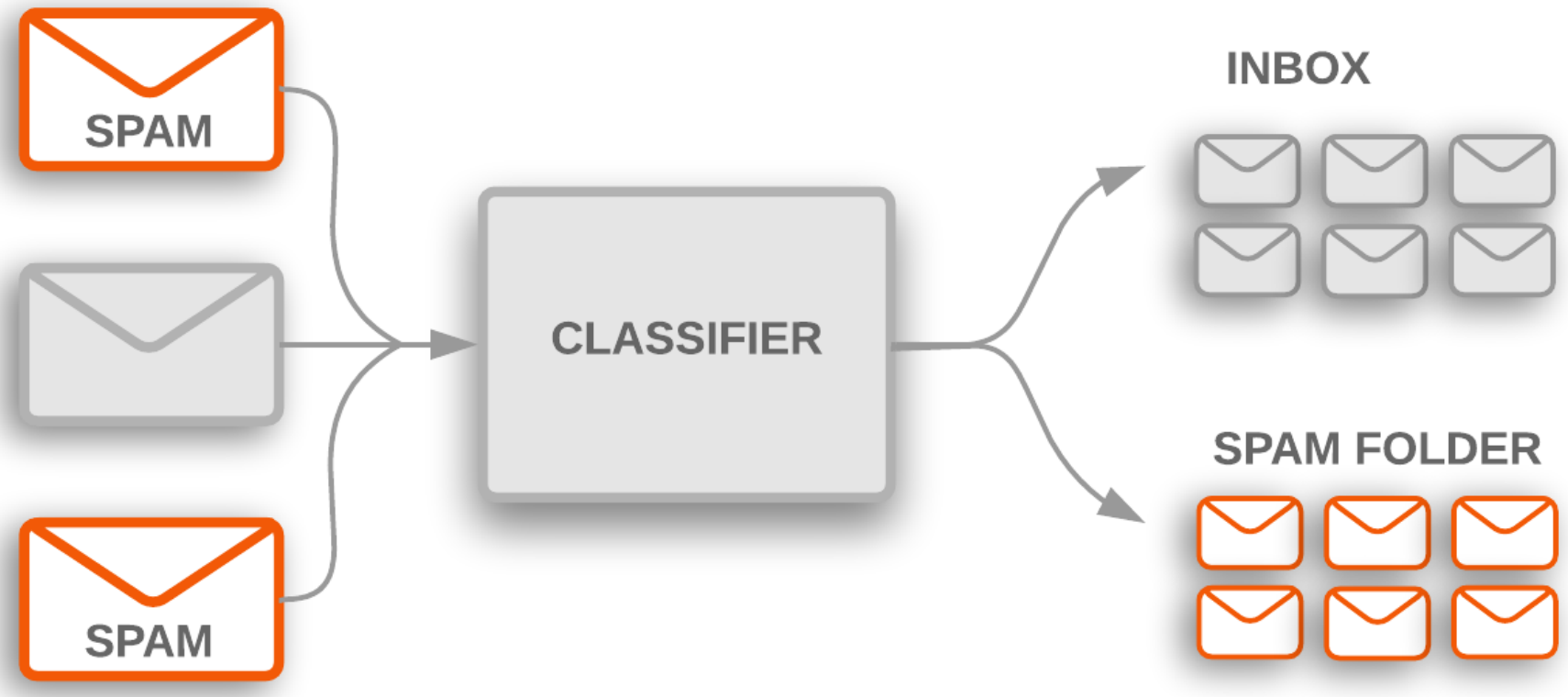
Classification Scheme



Classification



Classification: application



Classification: application

K-NEAREST NEIGHBOUR

K-Nearest Neighbour[2]

Training

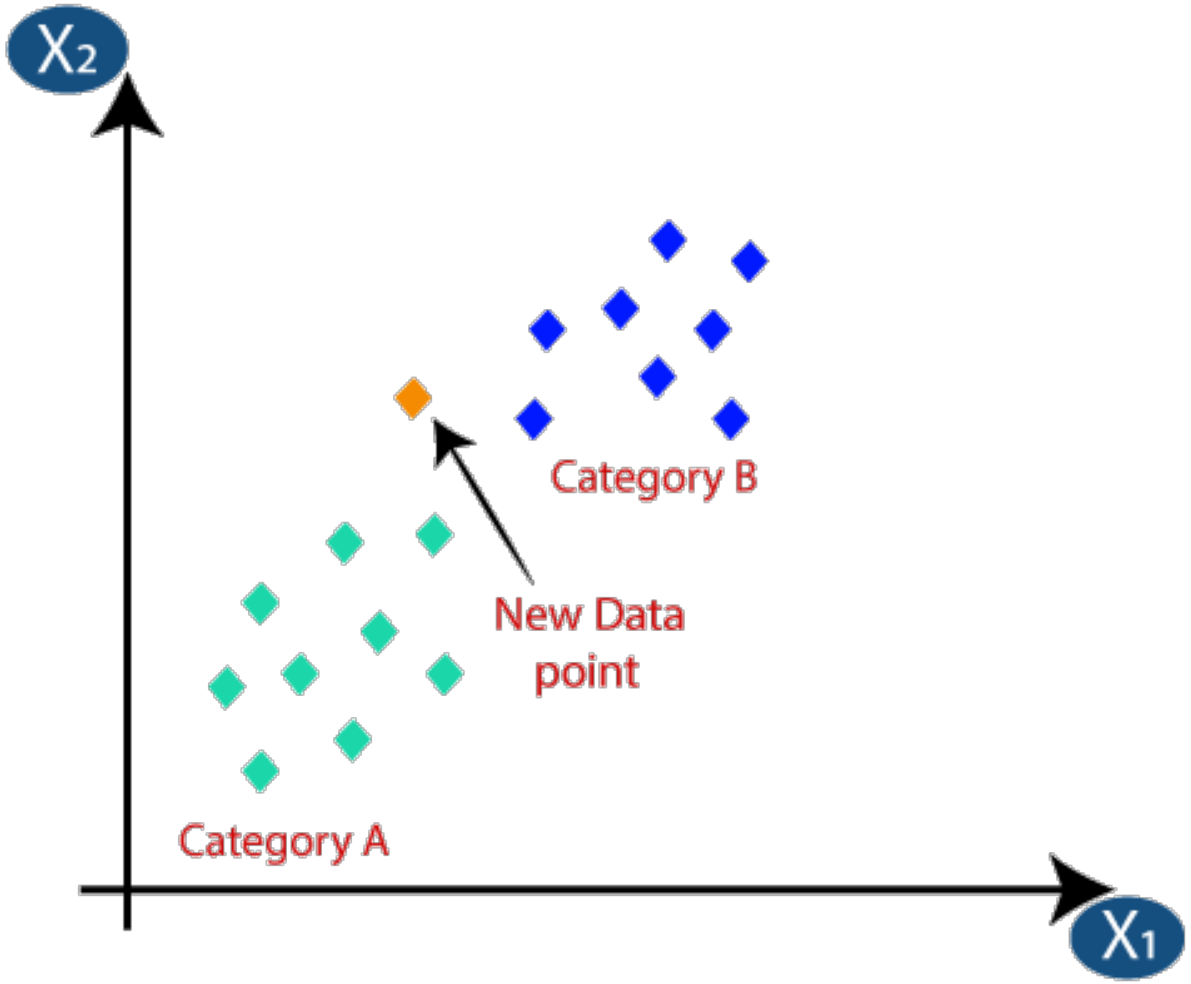
- Just stores the input data with labels
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

Test

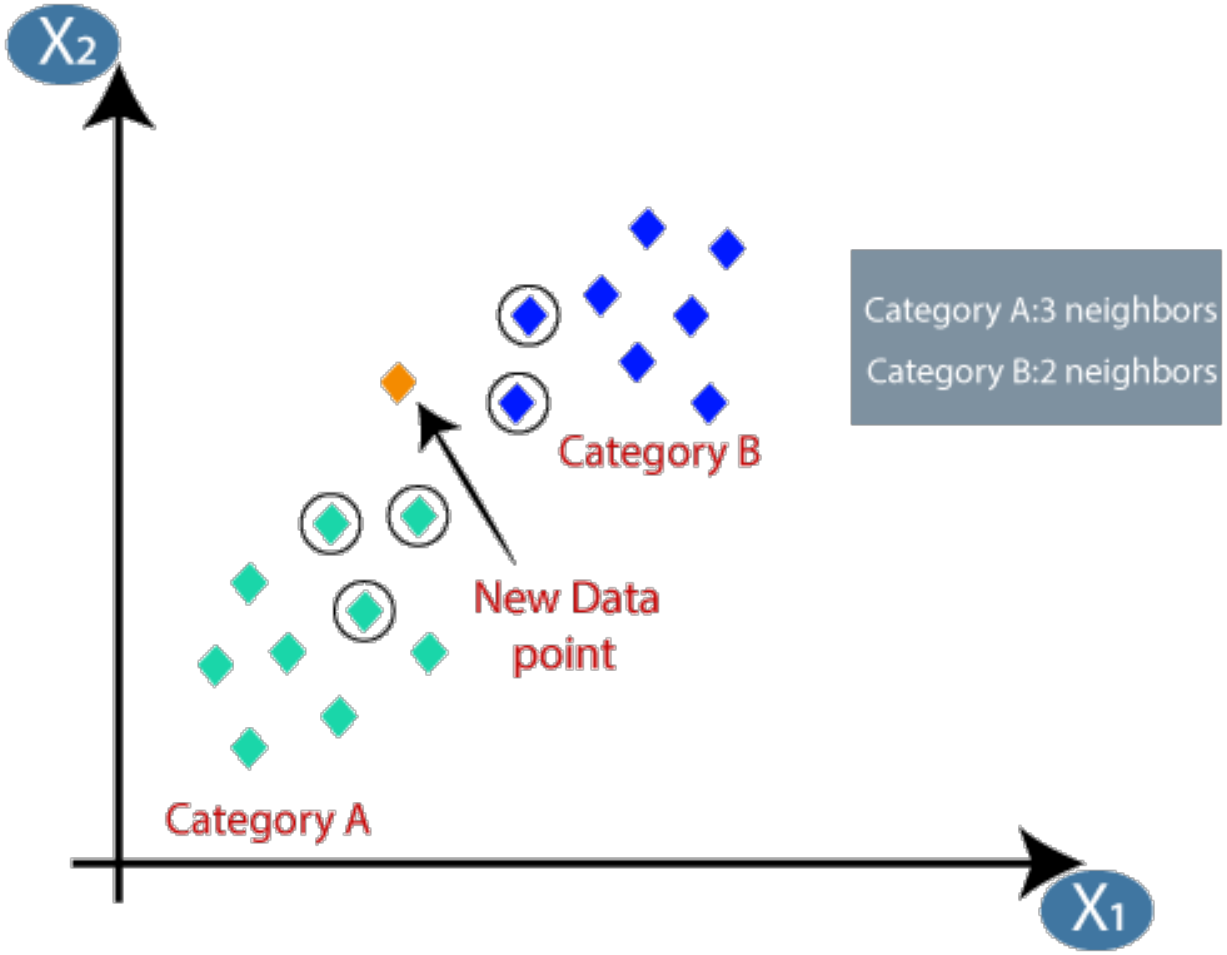
1. Step-1: Select the number K of the neighbours
2. Step-2: Calculate the distance of K number of neighbours with the new data point
3. Step-3: Take the K nearest neighbours as per the calculated distance.
4. Step-4: Among these K neighbors, count the number of the data points in each class/category.
5. Step-5: Assign the new data point to that class/category for which the number of the neighbor is maximum.

[2] https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

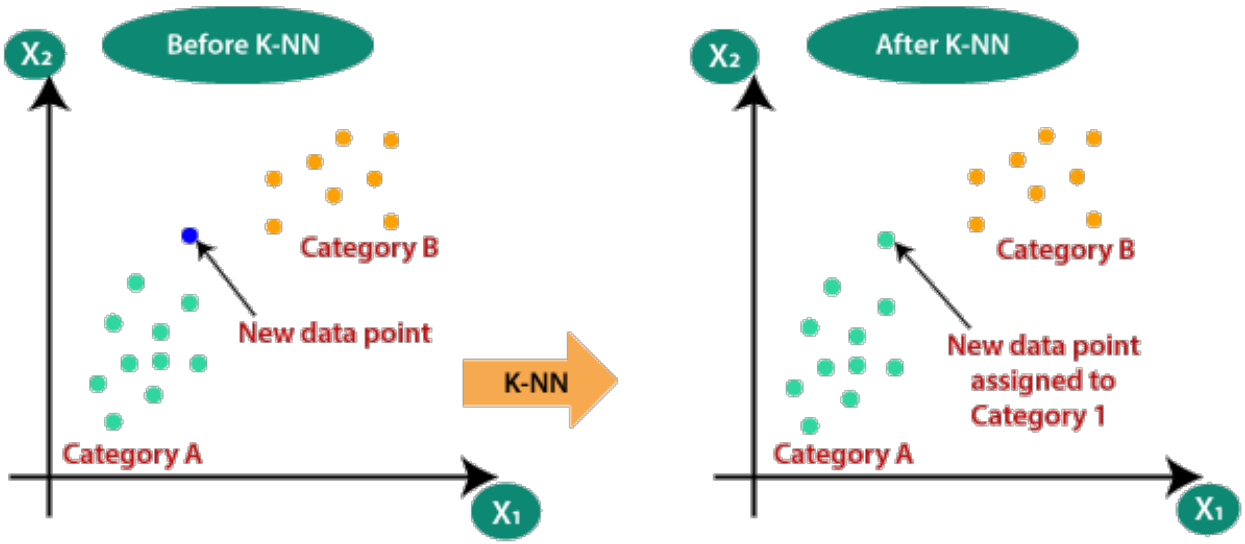
K-Nearest Neighbour



K-Nearest Neighbour



K-Nearest Neighbour



K-Nearest Neighbour

Advantage

Simple

Disadvantage

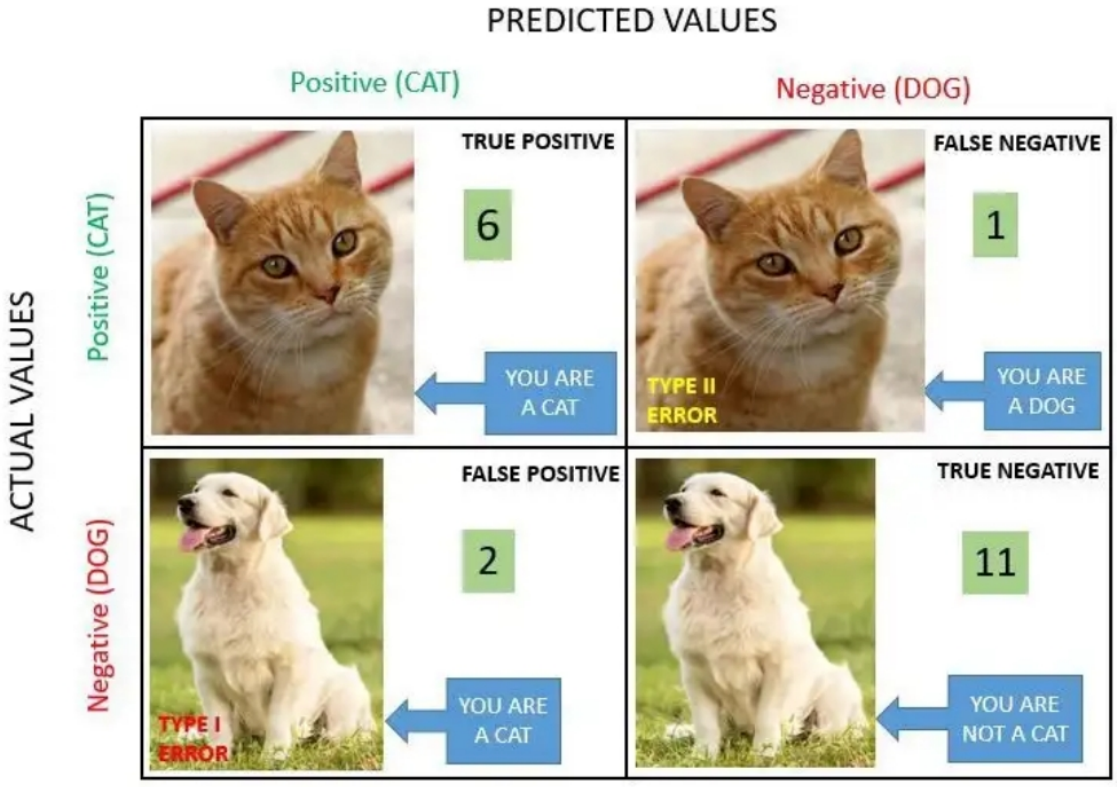
- Determining K can be a challenge
- Computationally expensive as involves the computation of distance of the new data point with all training points

Confusion Matrix[3]

		Predicted	
		Negative **(N)** -	Positive **(P)** +
Actual	Negative -	True Negative **(TN)**	**False Positive (FP) Type I Error**
	Positive +	**False Negative (FN) Type II Error**	True Positive **(TP)**

[3] https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

Confusion Matrix[4]



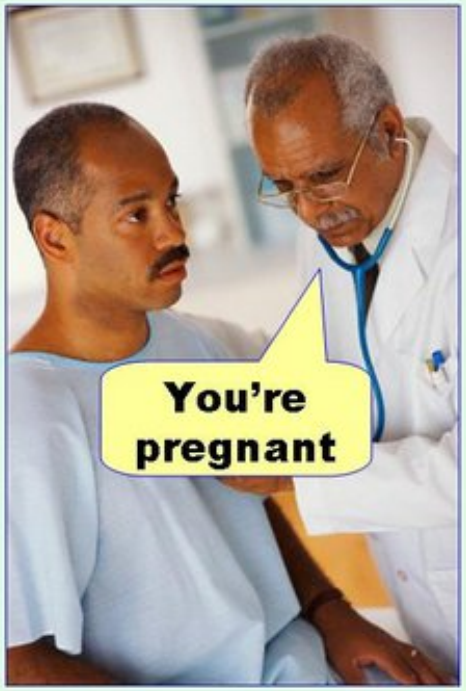
Classification Task

Positive: CAT, Negative: NOT CAT (DOG)

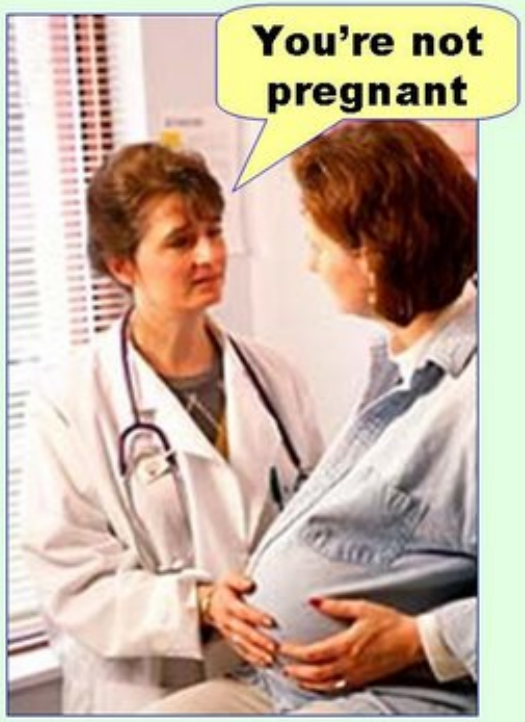
[4] https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

Confusion Matrix: Viral Example

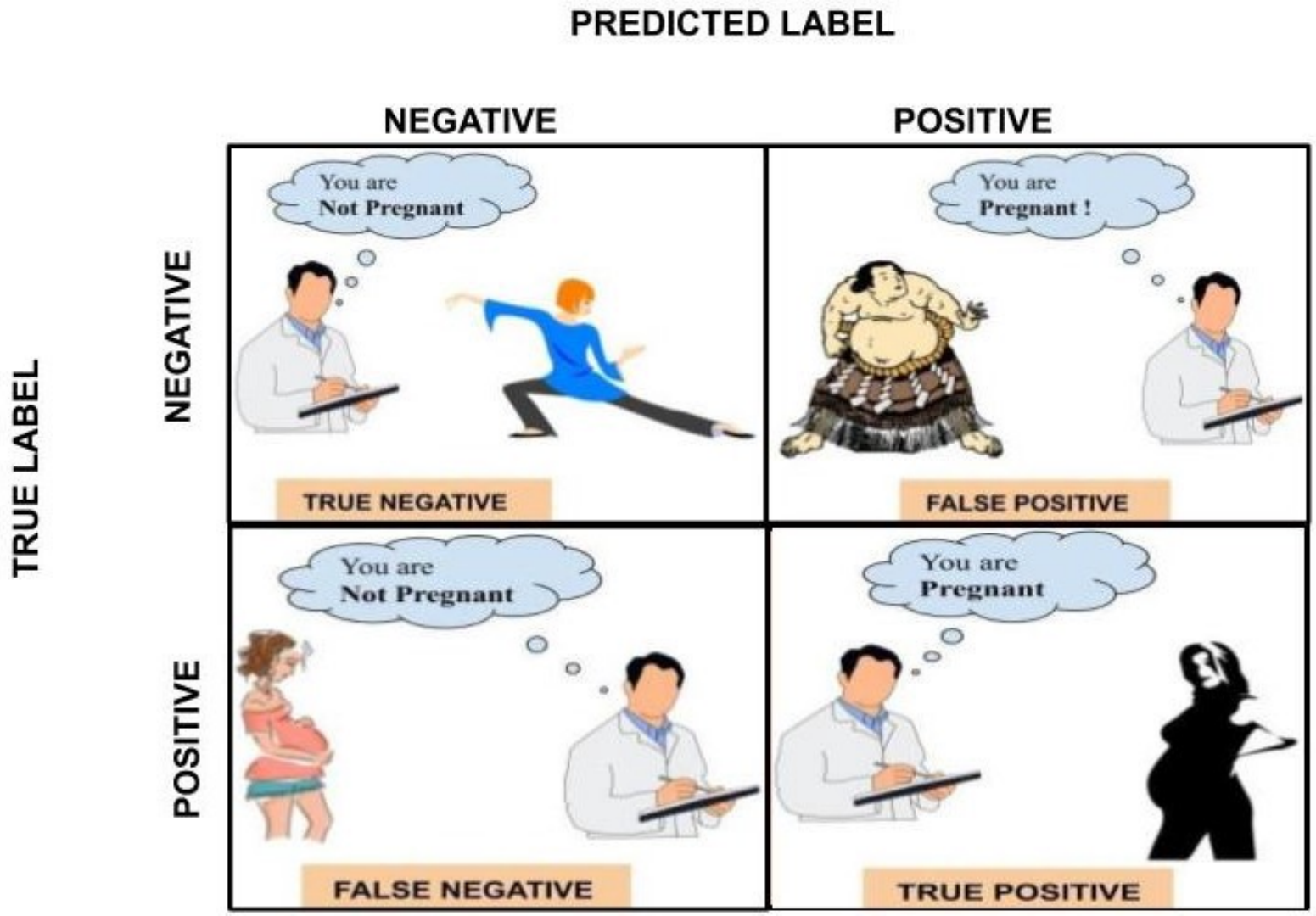
Type I error
(false positive)



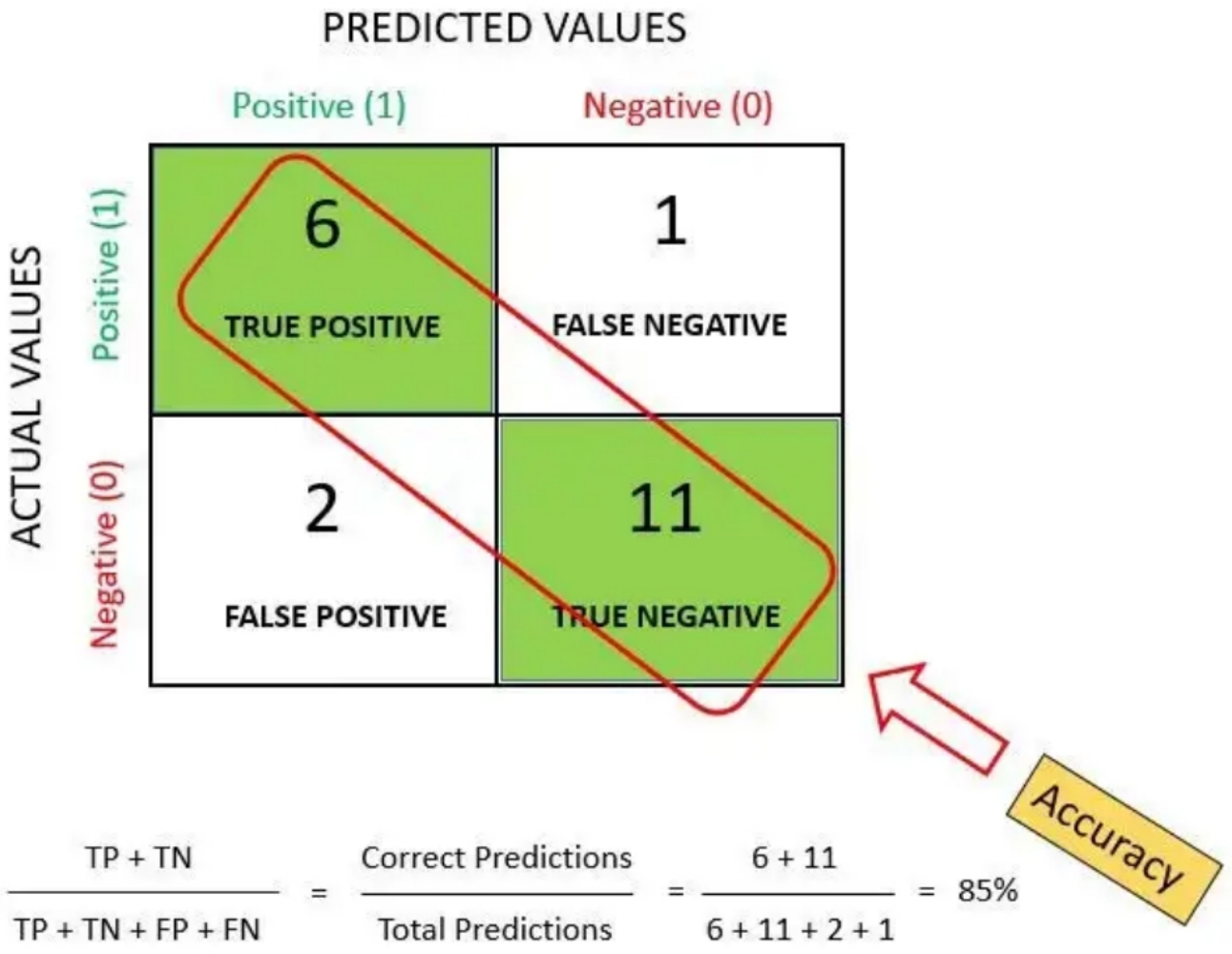
Type II error
(false negative)



Confusion Matrix: Comic Example

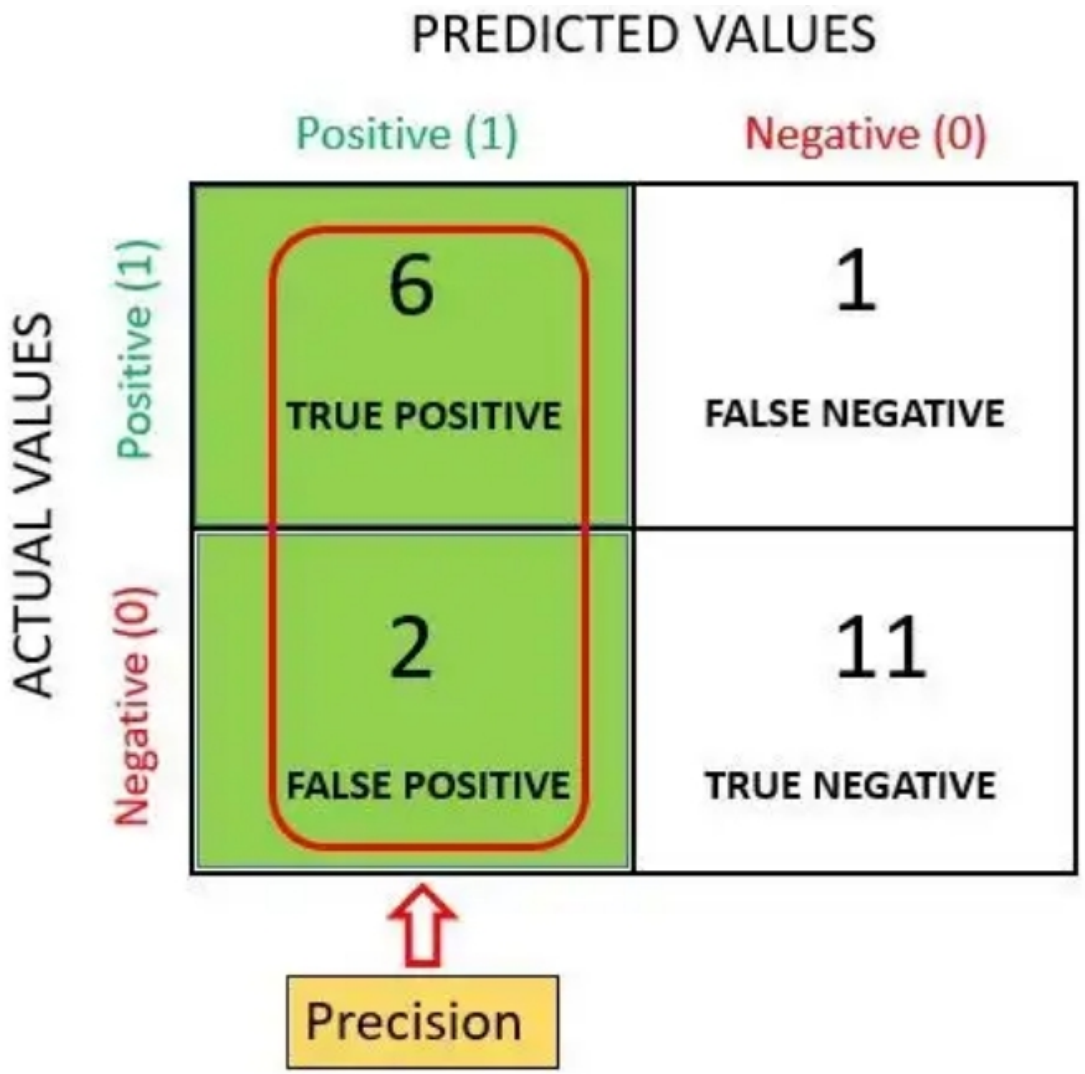


Confusion Matrix: Accuracy



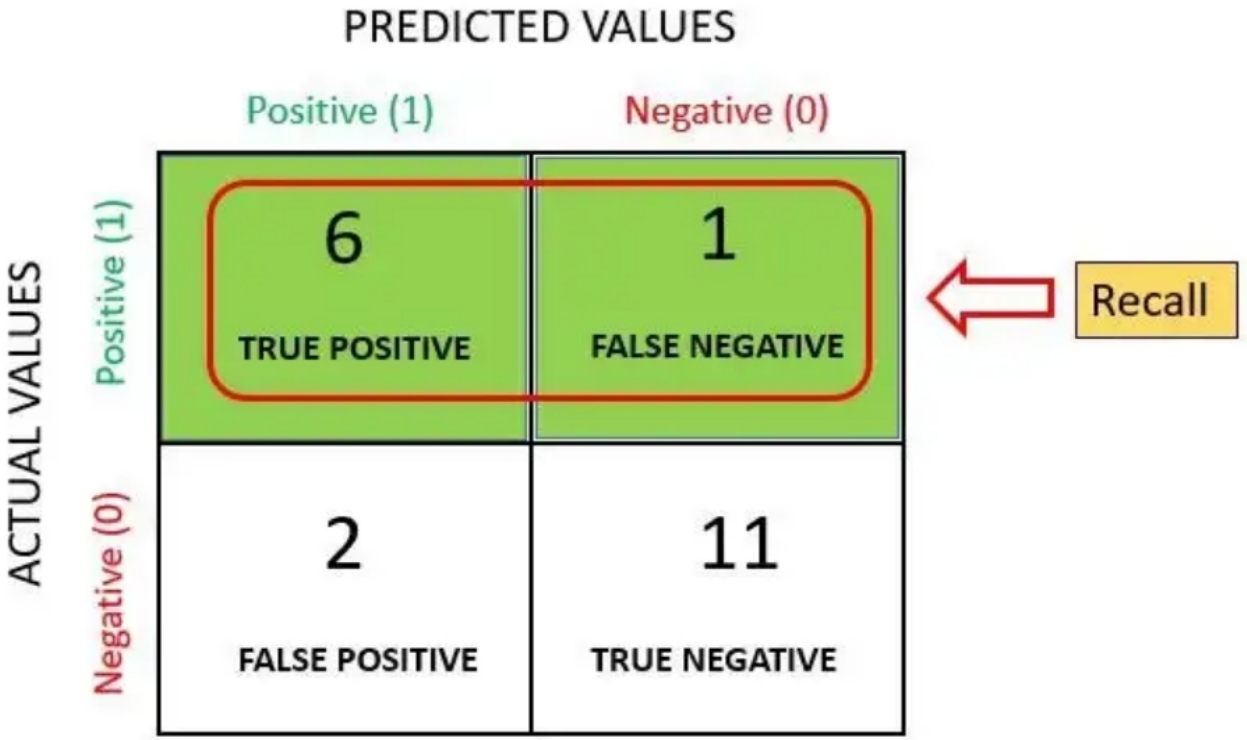
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} = \frac{6 + 11}{6 + 11 + 2 + 1} = 85\%$$

Confusion Matrix: Precision



$$\text{Precision} = \frac{TP}{TP + FP} = \frac{\text{Predictions Actually Positive}}{\text{Total Predicted positive}} = \frac{6}{6+2} = 0.75$$

Confusion Matrix: Recall



$$\text{Recall} = \frac{TP}{TP + FN} = \frac{\text{Predictions Actually Positive}}{\text{Total Actual positive}} = \frac{6}{6 + 1} = 0.85$$

Confusion Matrix: F1-Score

$$\text{F1-Score} = 2 * \frac{(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} = 2 * \frac{(0.85 * 0.75)}{(0.85 + 0.75)} = 0.79$$

THANK YOU !!!