Machine Learning

Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

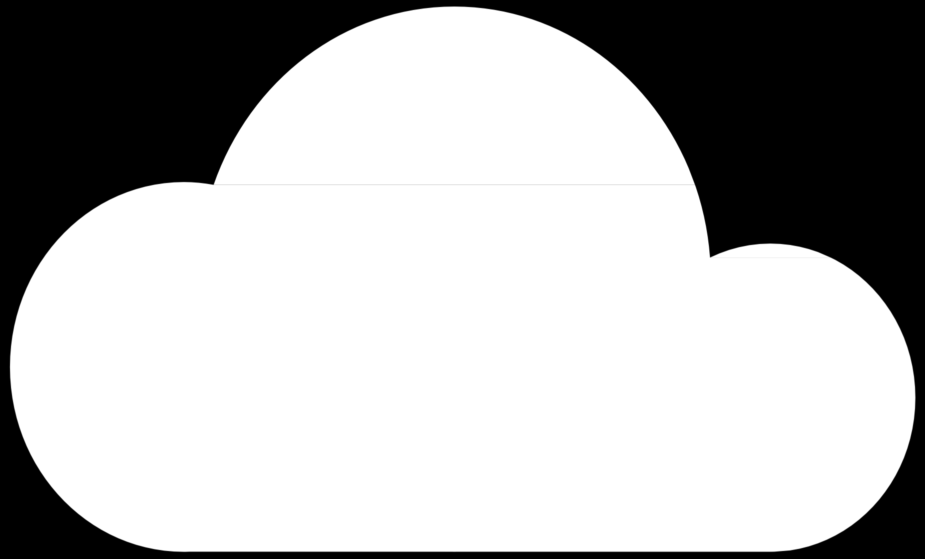
Supervised Learning

supervised learning

given a data set of input-output pairs, learn a function to map inputs to outputs

classification

supervised learning task of learning a function mapping an input point to a discrete category



Date	Humidity (relative humidity)	Pressure (sea level, mb)	Rain

Date	Humidity (relative humidity)	Pressure (sea level, mb)	Rain
January 1	93%	999.7	Rain
January 2	49%	1015.5	No Rain
January 3	79%	1031.1	No Rain
January 4	65%	984.9	Rain
January 5	90%	975.2	Rain

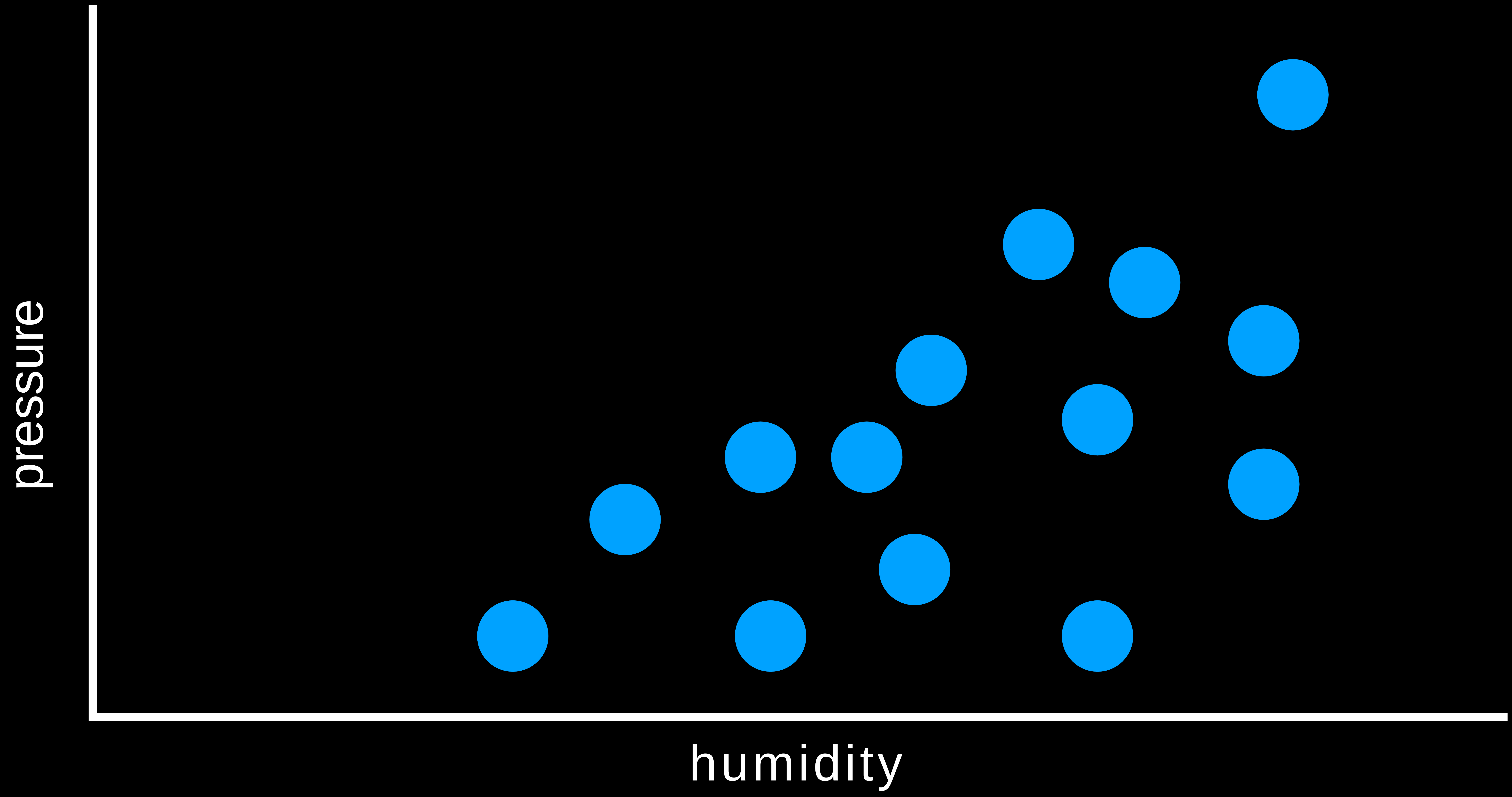
$$f(\mathit{humidity}, \mathit{pressure})$$

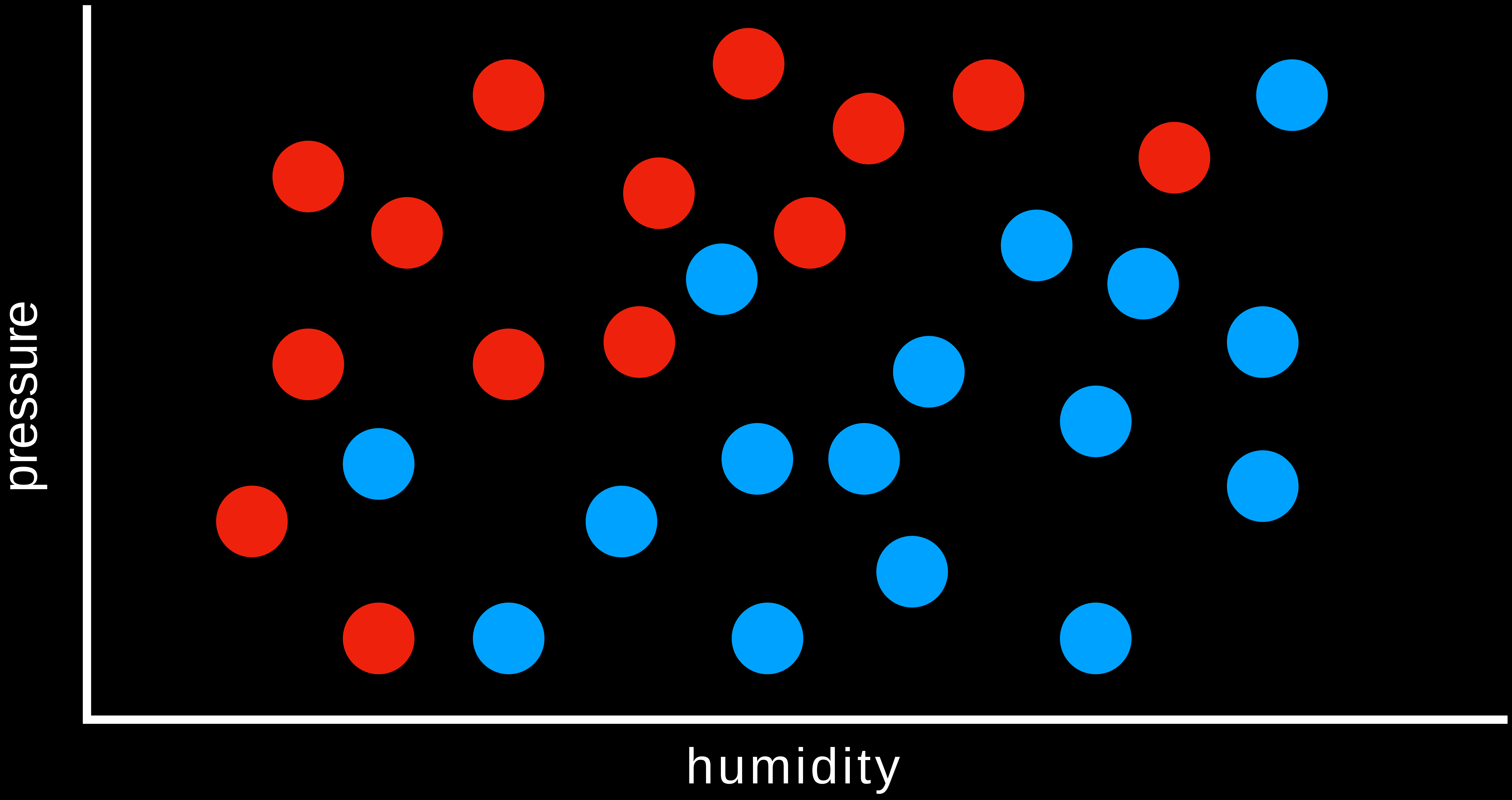
$$f(93, 999.7) = \text{Rain}$$

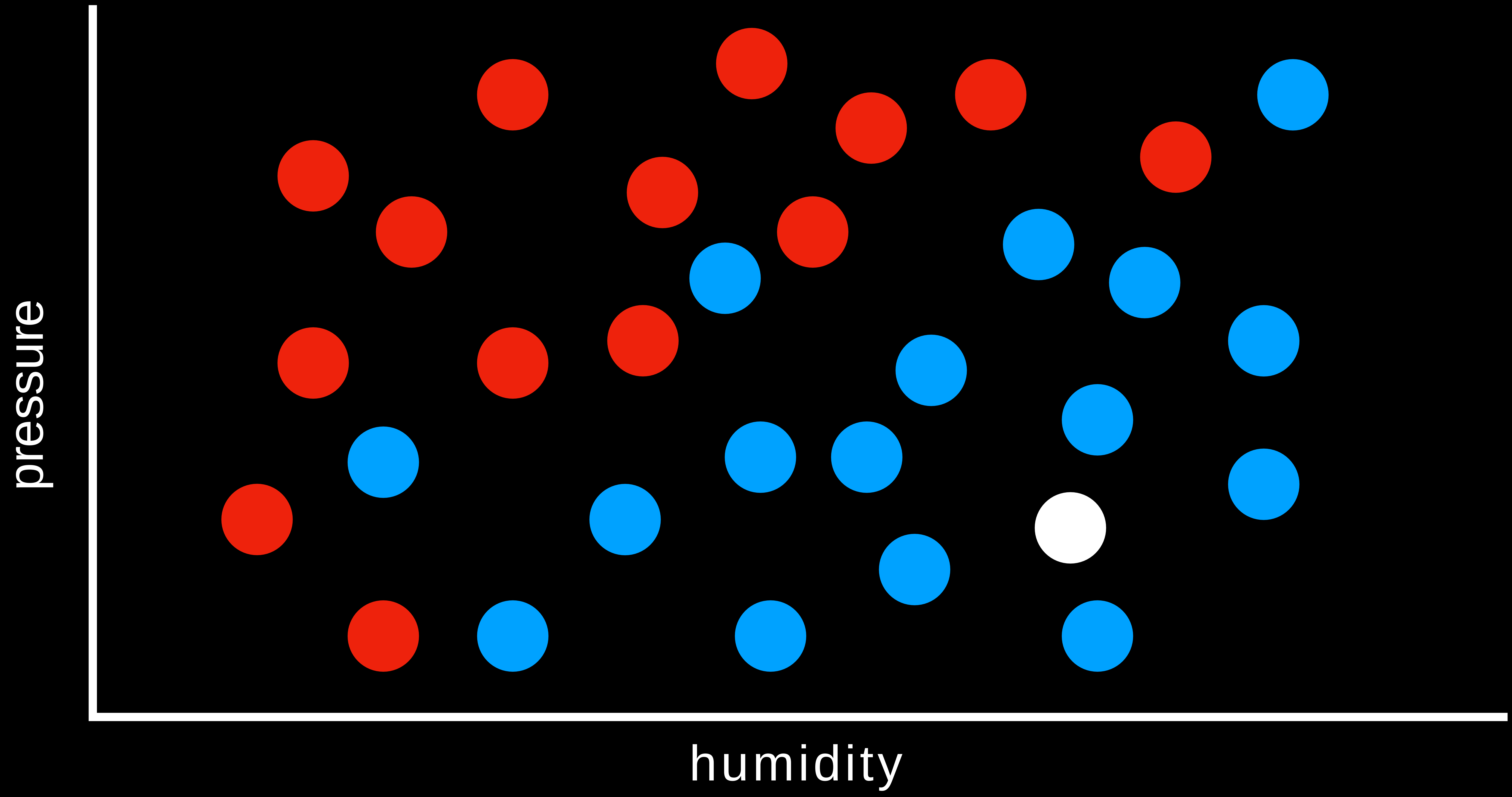
$$f(49, 1015.5) = \text{No Rain}$$

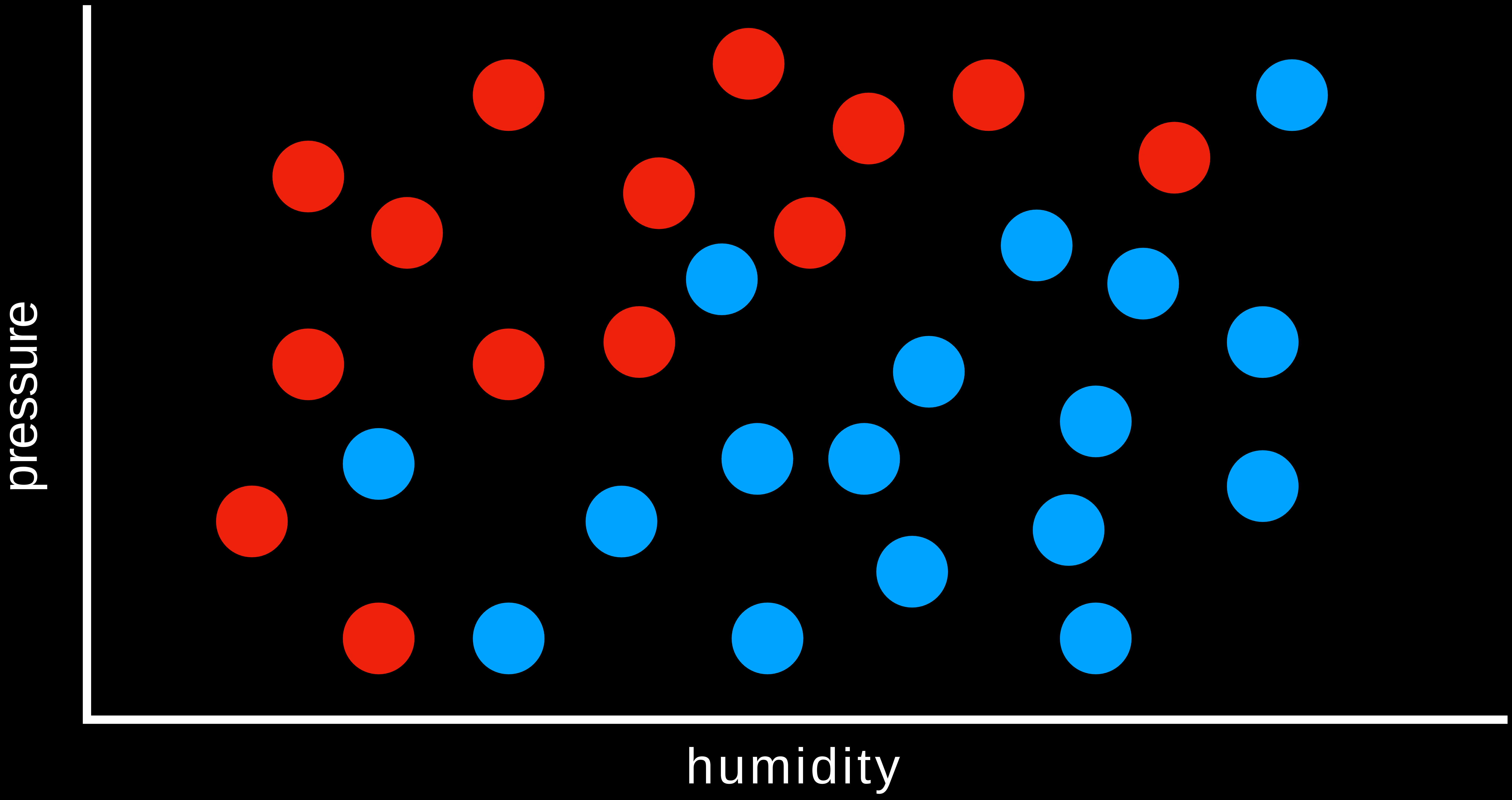
$$f(79, 1031.1) = \text{No Rain}$$

pressure
humidity


pressure
humidity

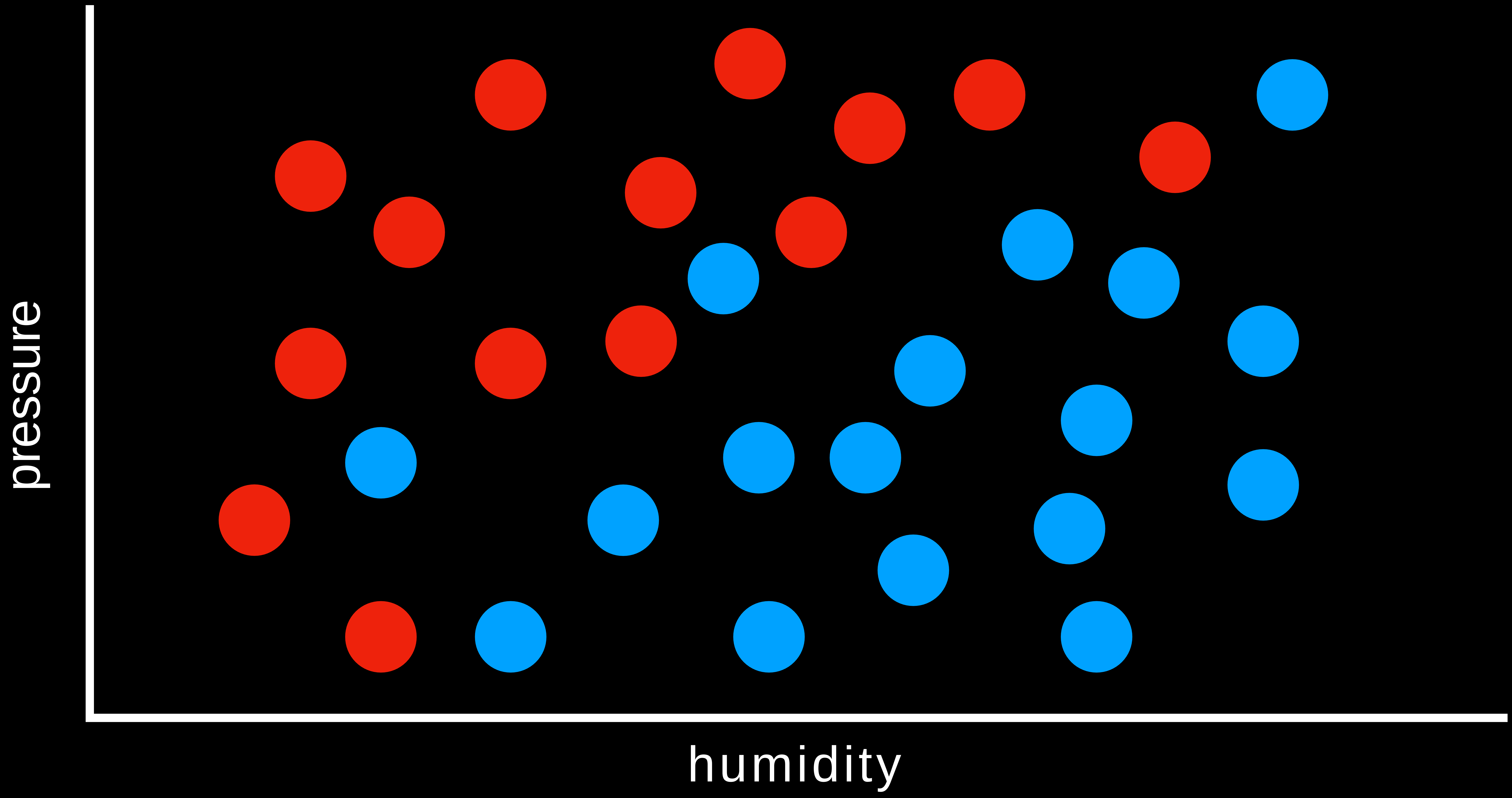

pressure
humidity

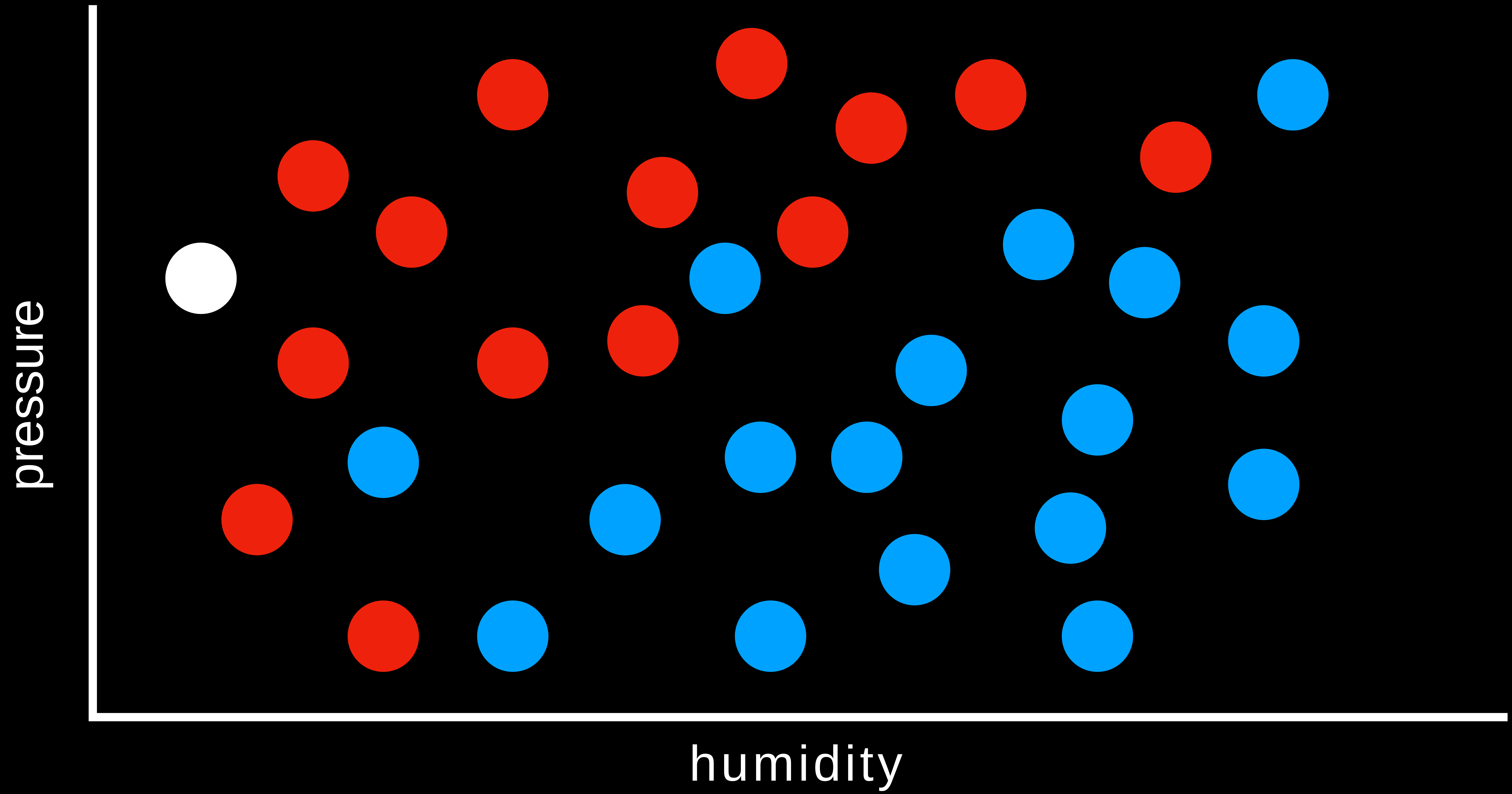

pressure
humidity

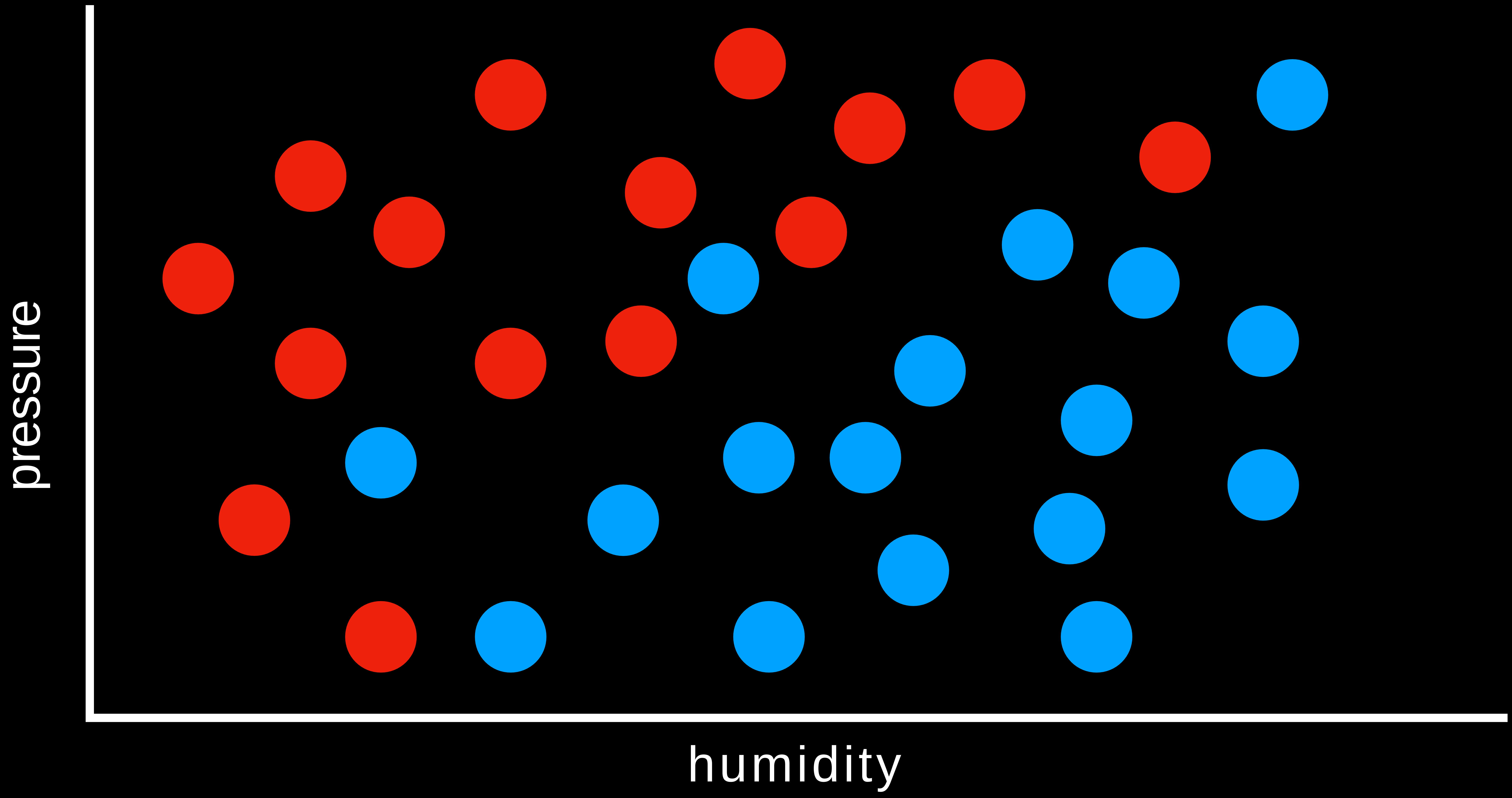

pressure
humidity

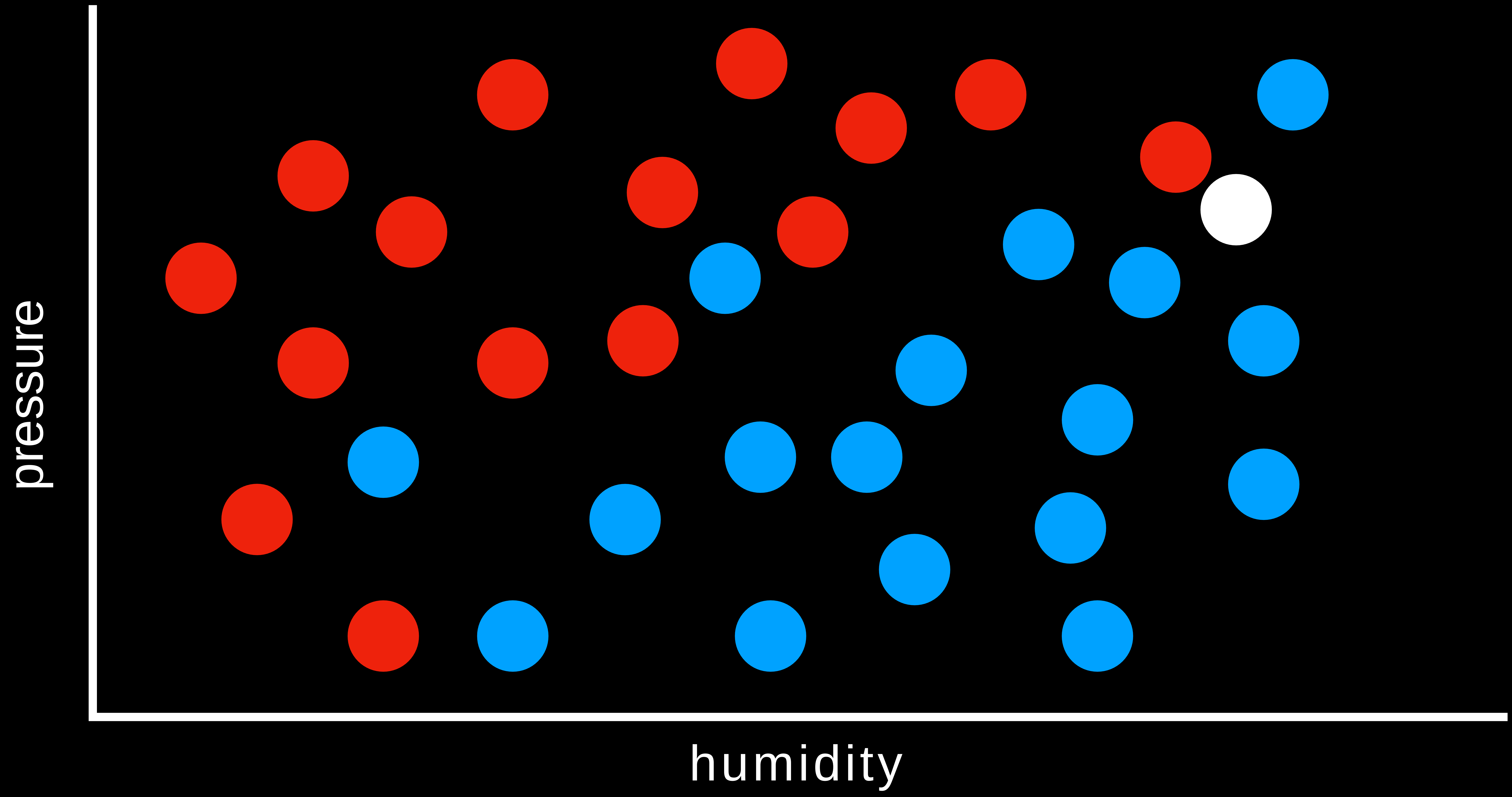
nearest-neighbor classification

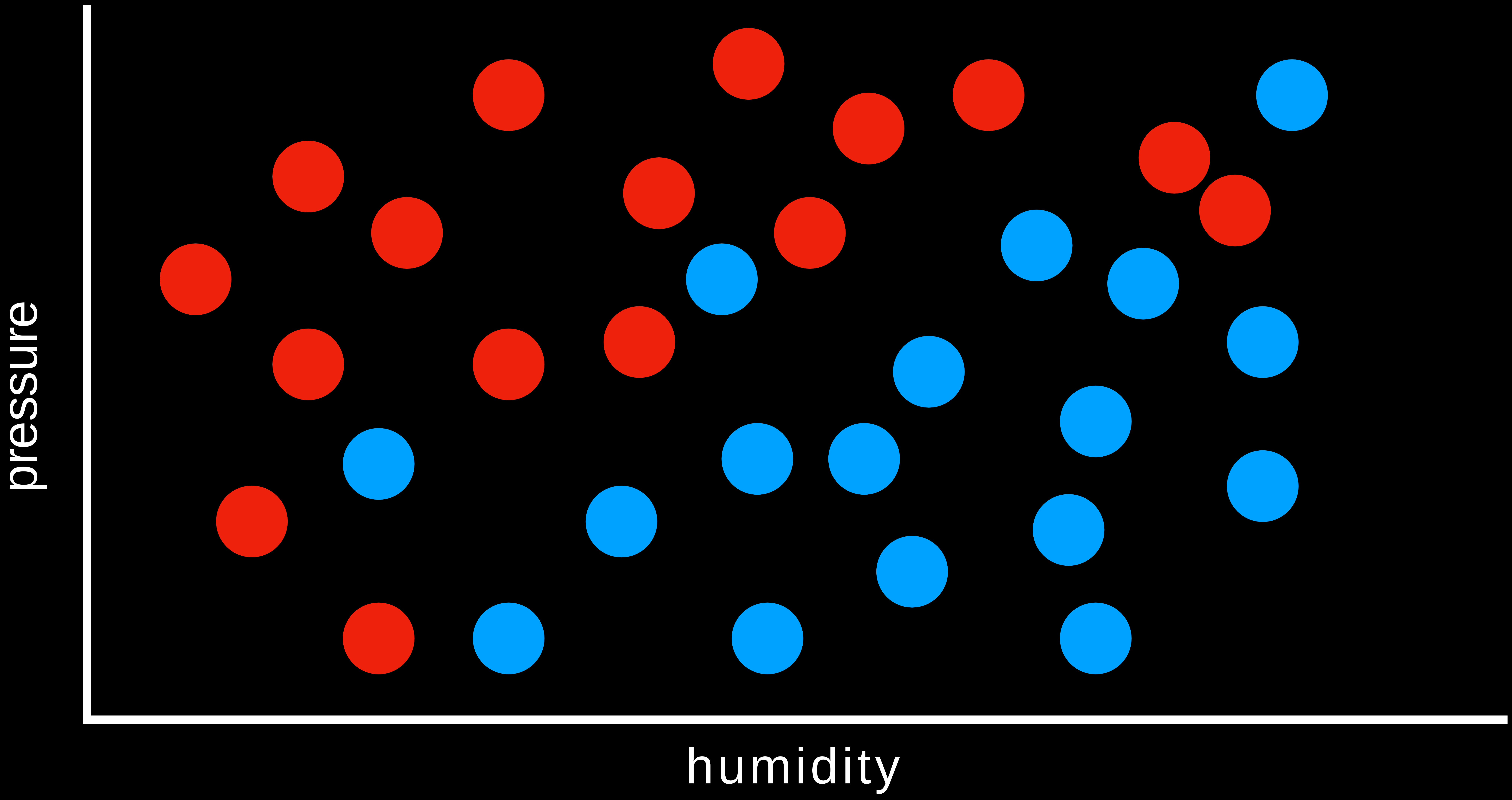
algorithm that, given an input, chooses the class of the nearest data point to that input

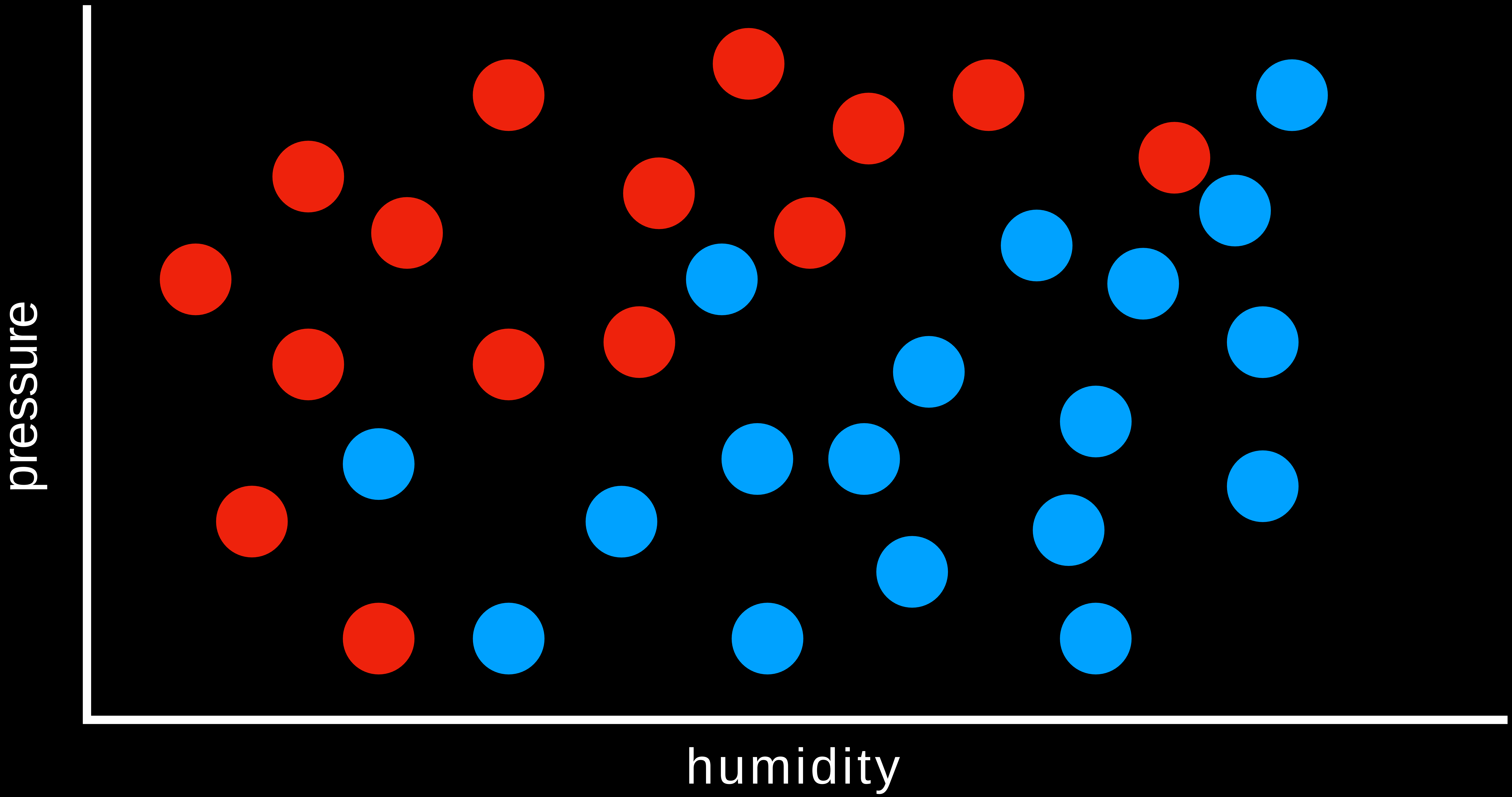

pressure
humidity


pressure
humidity


pressure
humidity

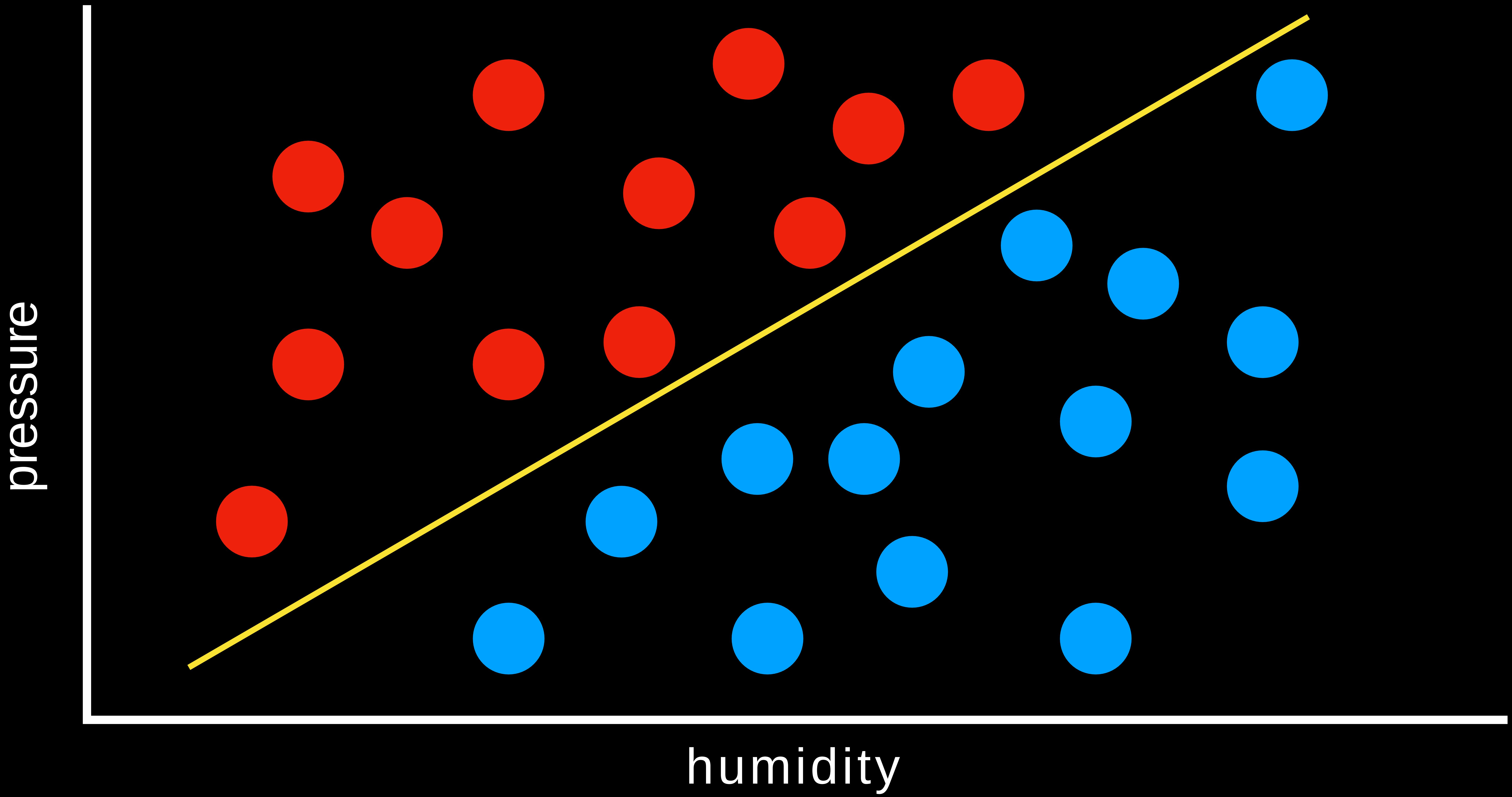

pressure
humidity

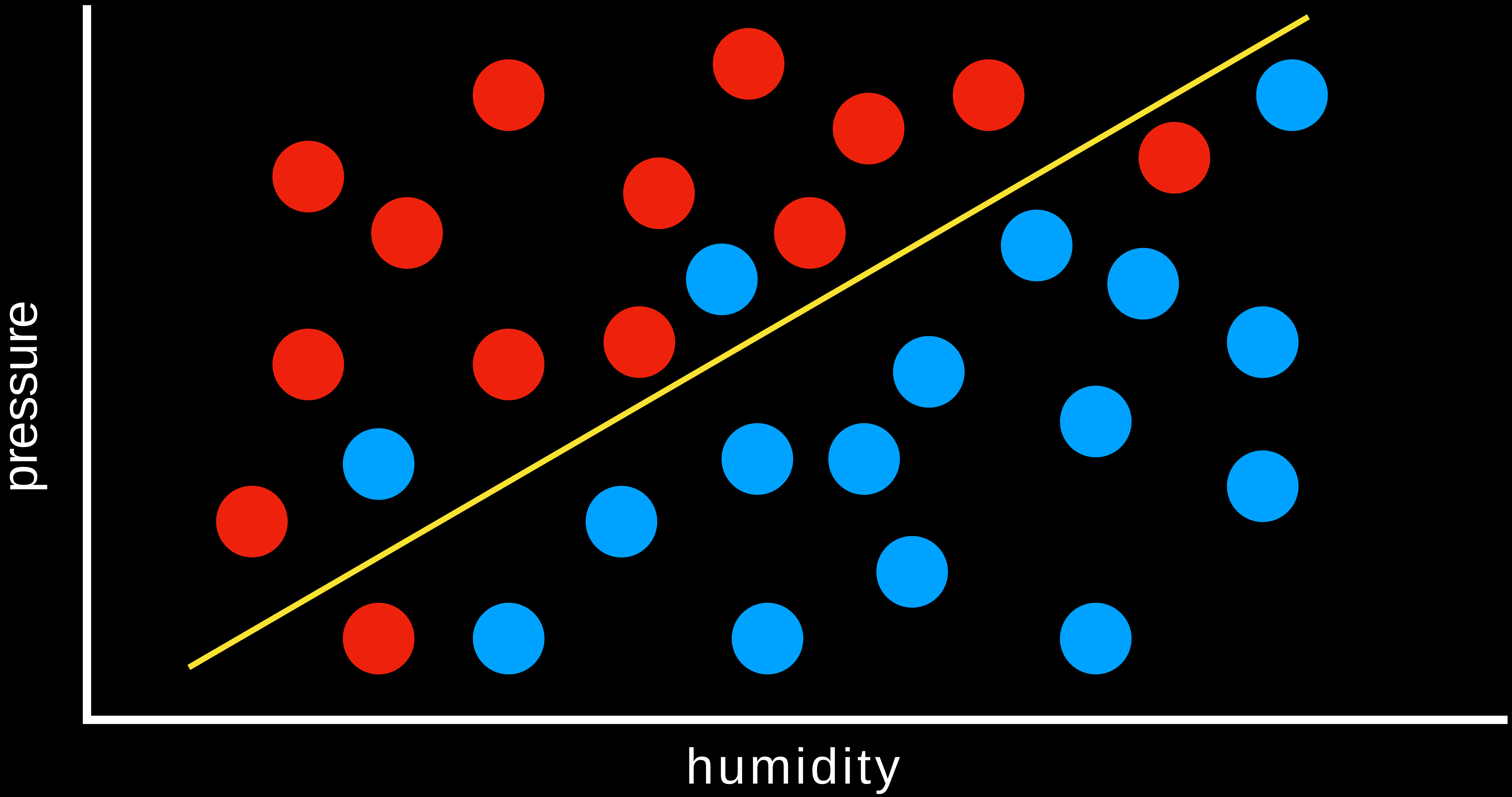

pressure
humidity


pressure
humidity

k-nearest-neighbor classification

algorithm that, given an input, chooses the most common class out of the k nearest data points to that input


pressure
humidity

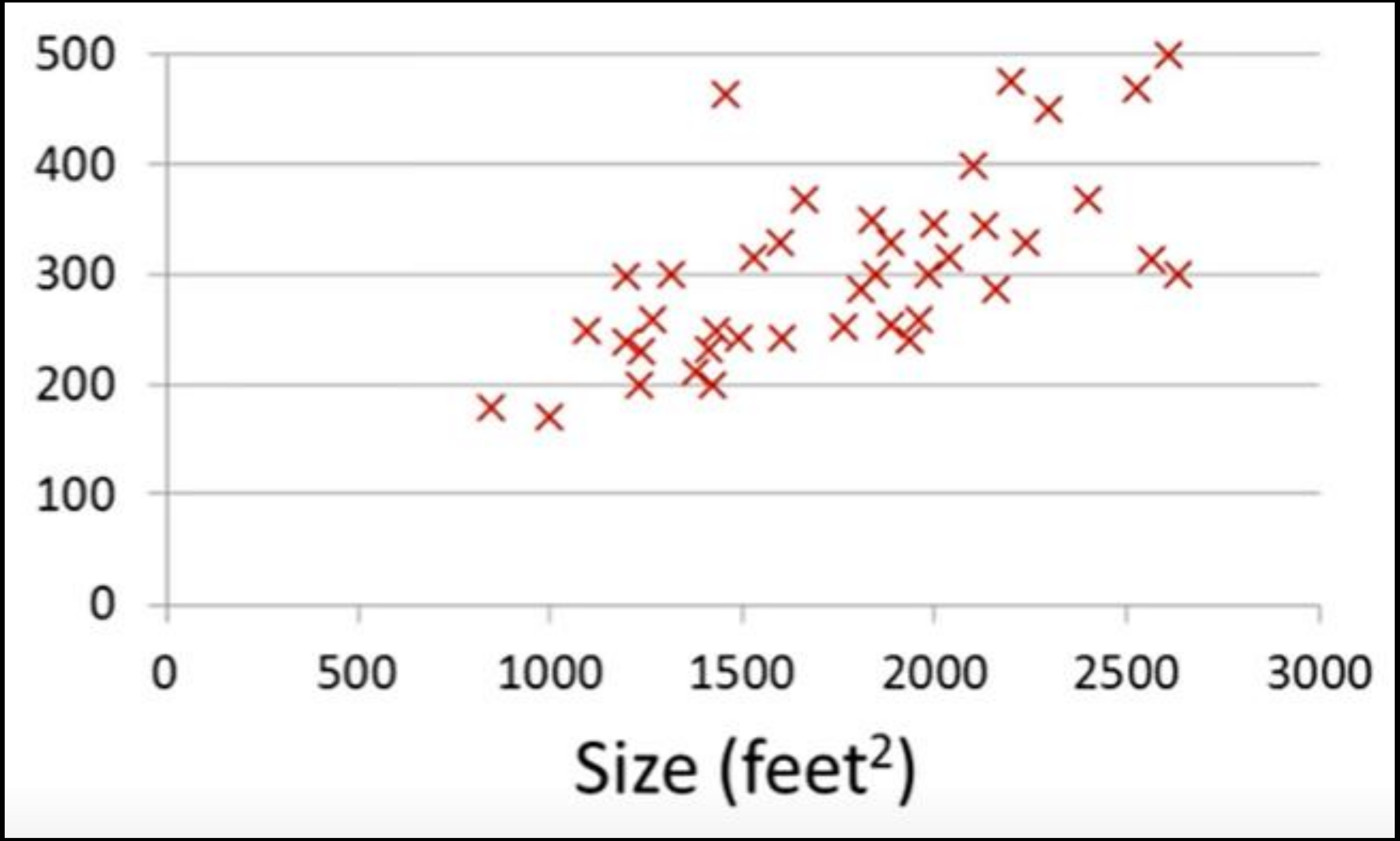

pressure
humidity

Linear Regression

- Linear regression is a linear model, e.g. a model that assumes a linear relationship b/w the input variables (x) and the single output variable (y).
- The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

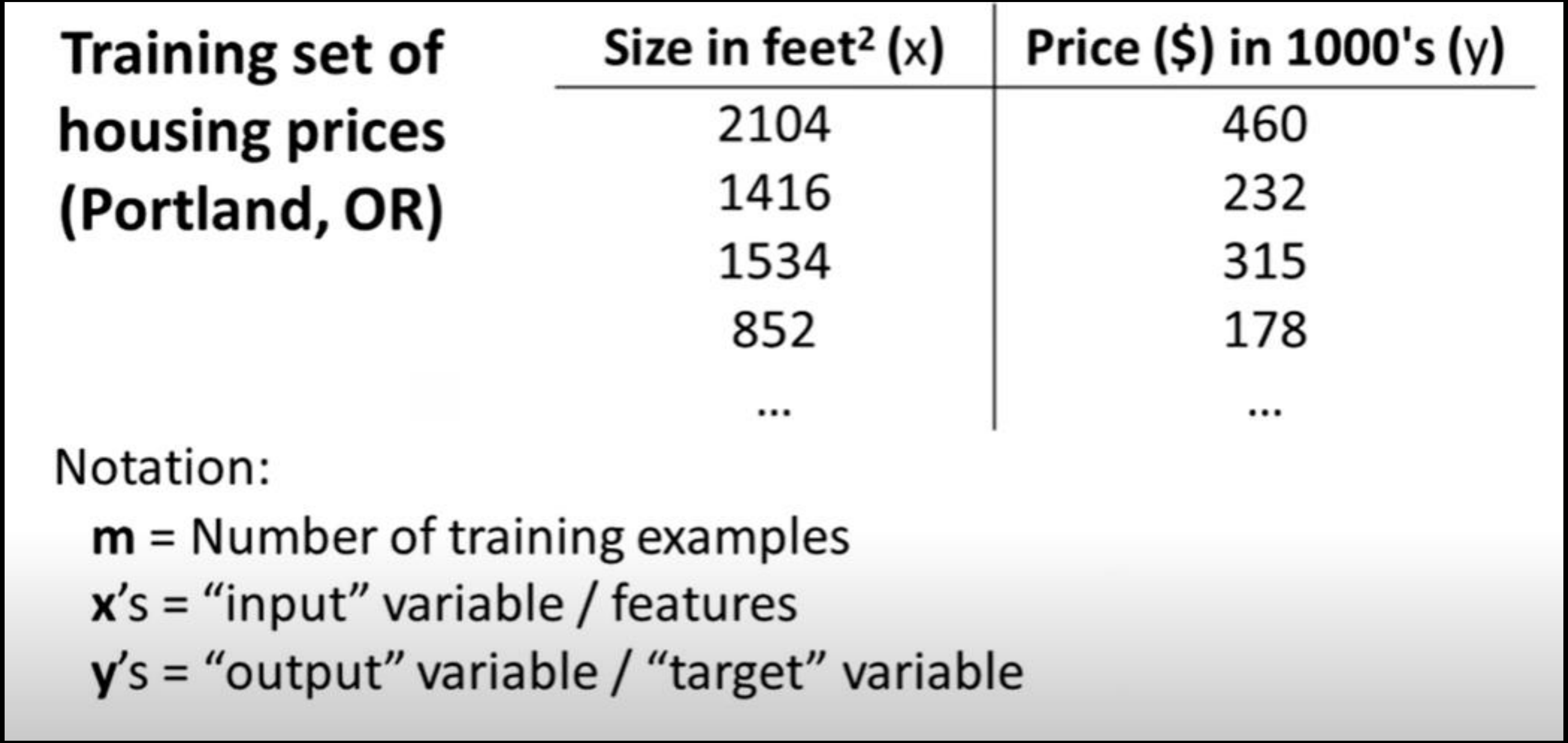
Cont..

Example: Housing Prices



Cont..

Example: Housing Prices



Training set of housing prices (Portland, OR)

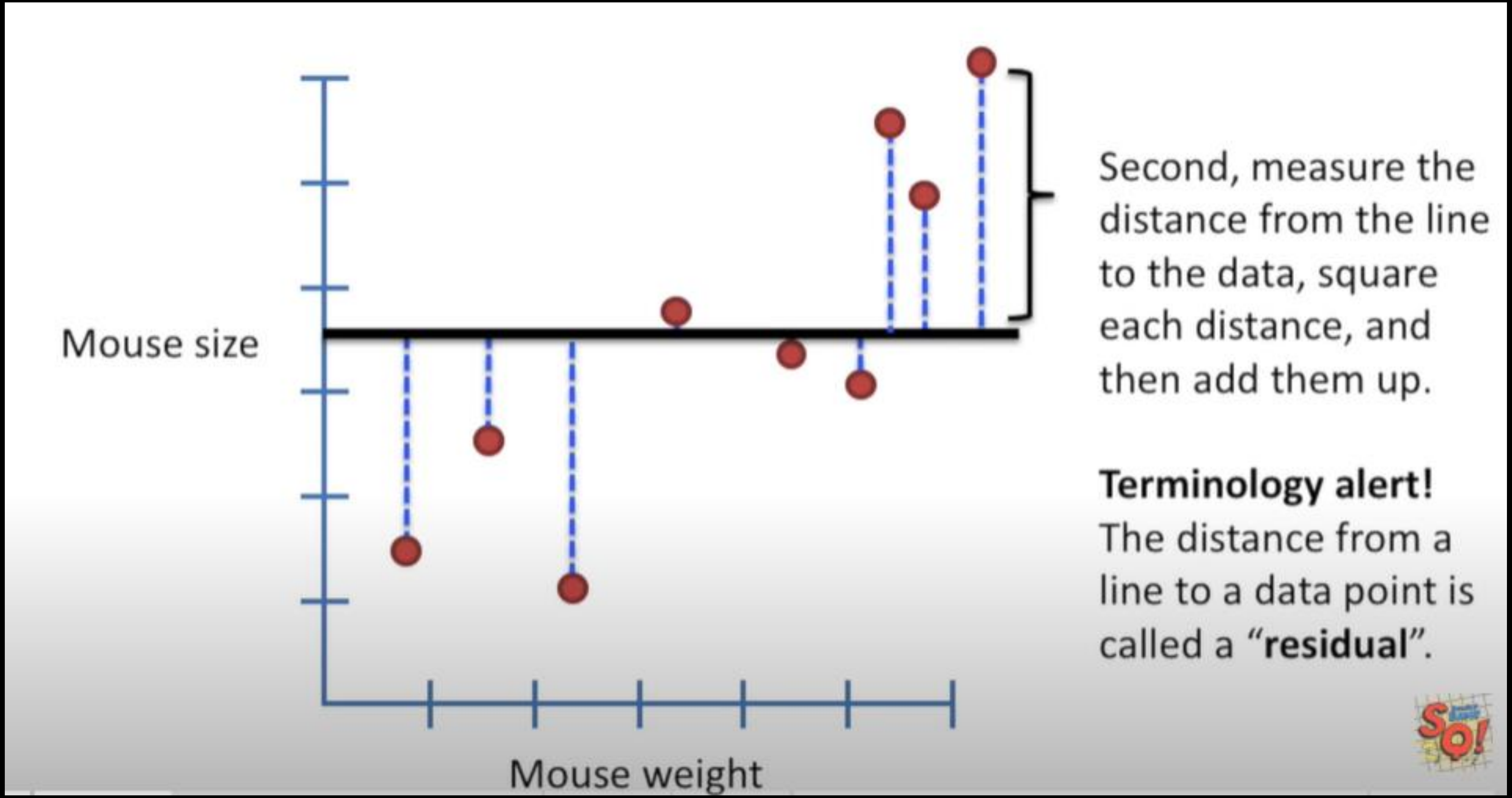
Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Notation:

m = Number of training examples
x's = "input" variable / features
y's = "output" variable / "target" variable

Cont..

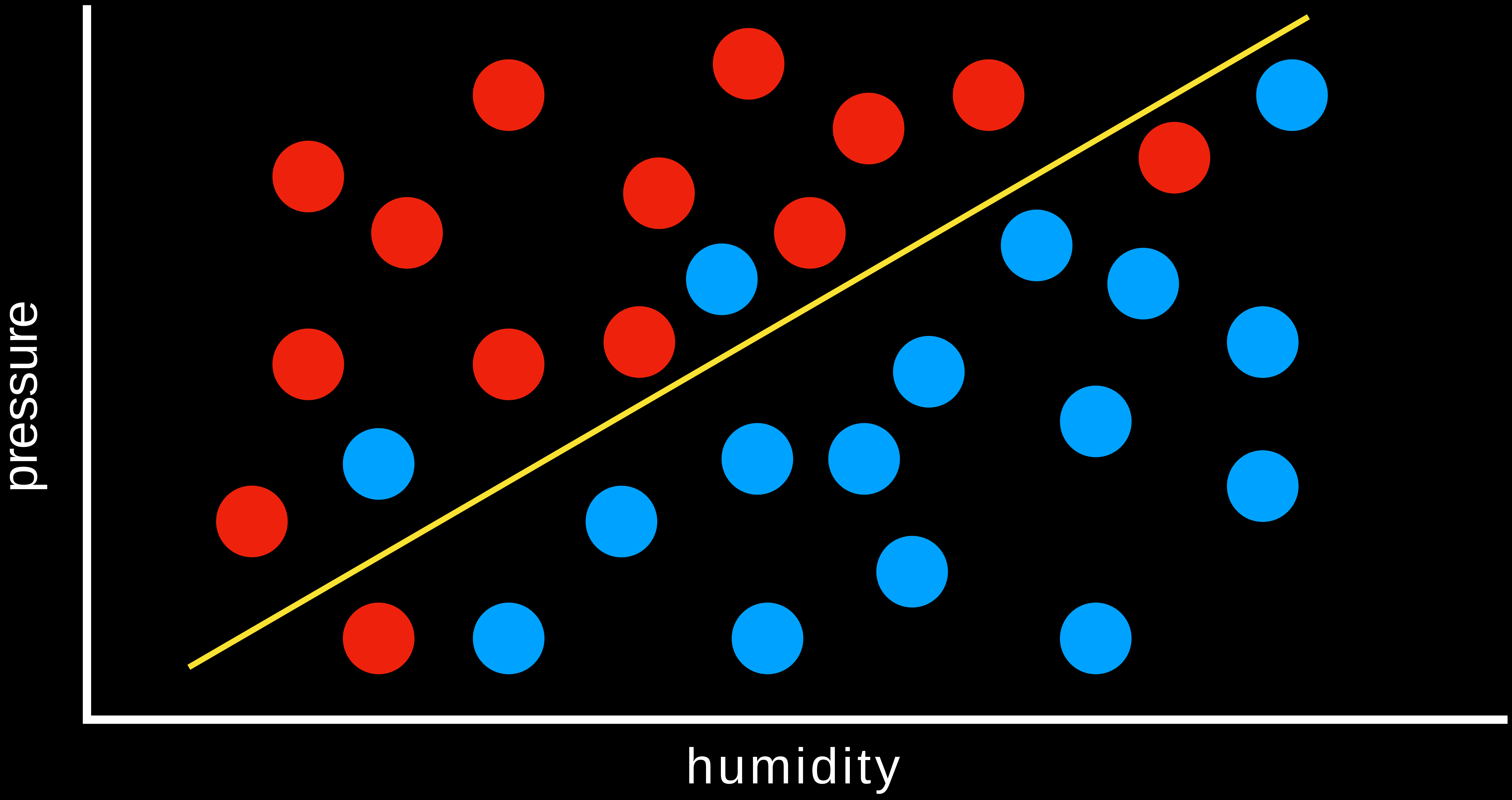
How could we get the best linear line?



Second, measure the distance from the line to the data, square each distance, and then add them up.

Terminology alert! The distance from a line to a data point is called a “**residual**”.

❖**Take the line that has the Least Square Error!**


pressure
humidity

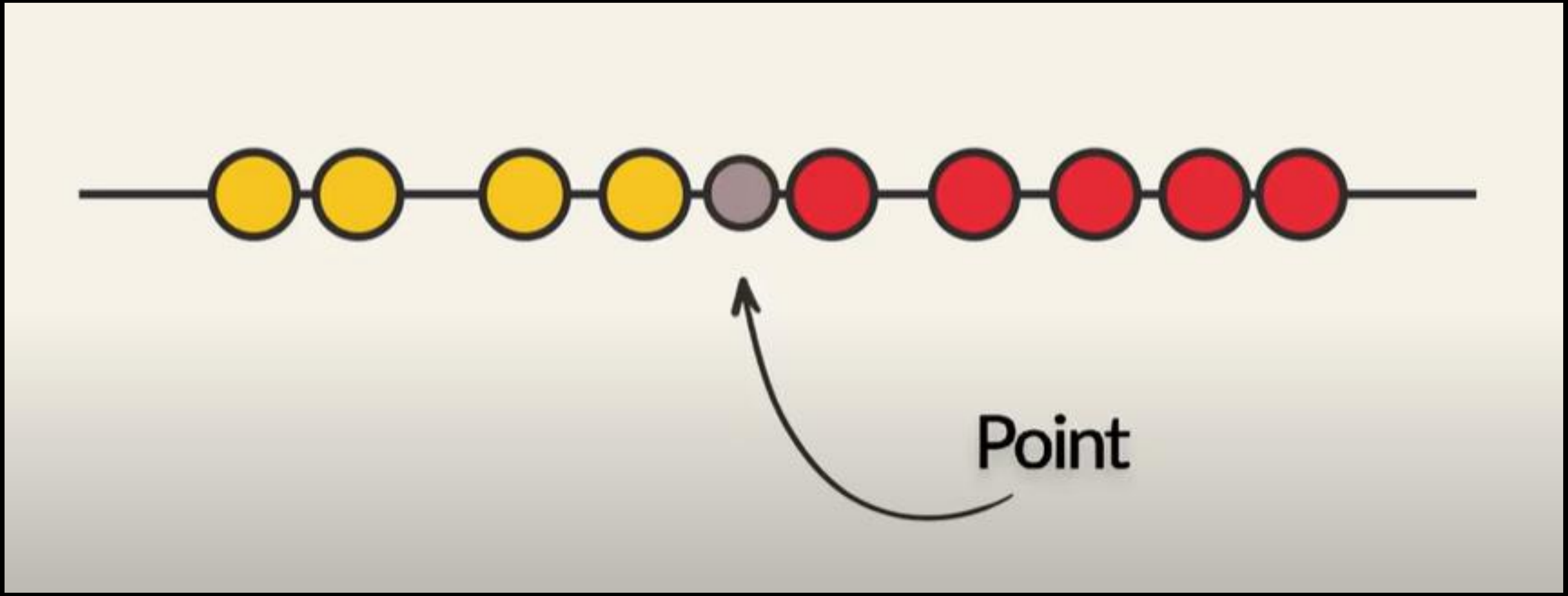
Support Vector Machines

- SVM is an exciting algorithm and the concepts are relatively simple. The classifier separates data points using a **hyperplane** with the largest amount of margin. SVM finds an optimal hyperplane which helps in classifying new data points.

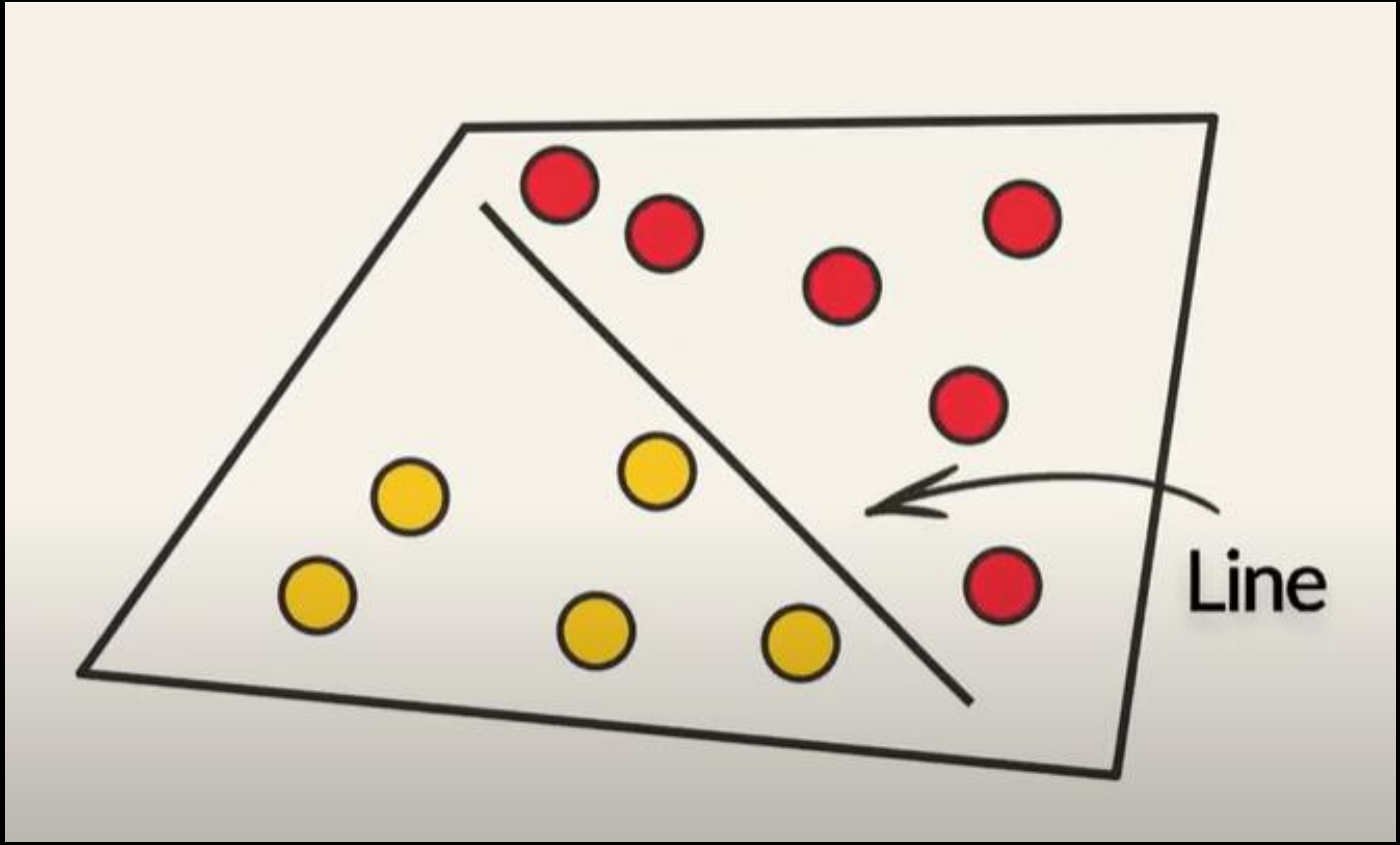
- The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes.

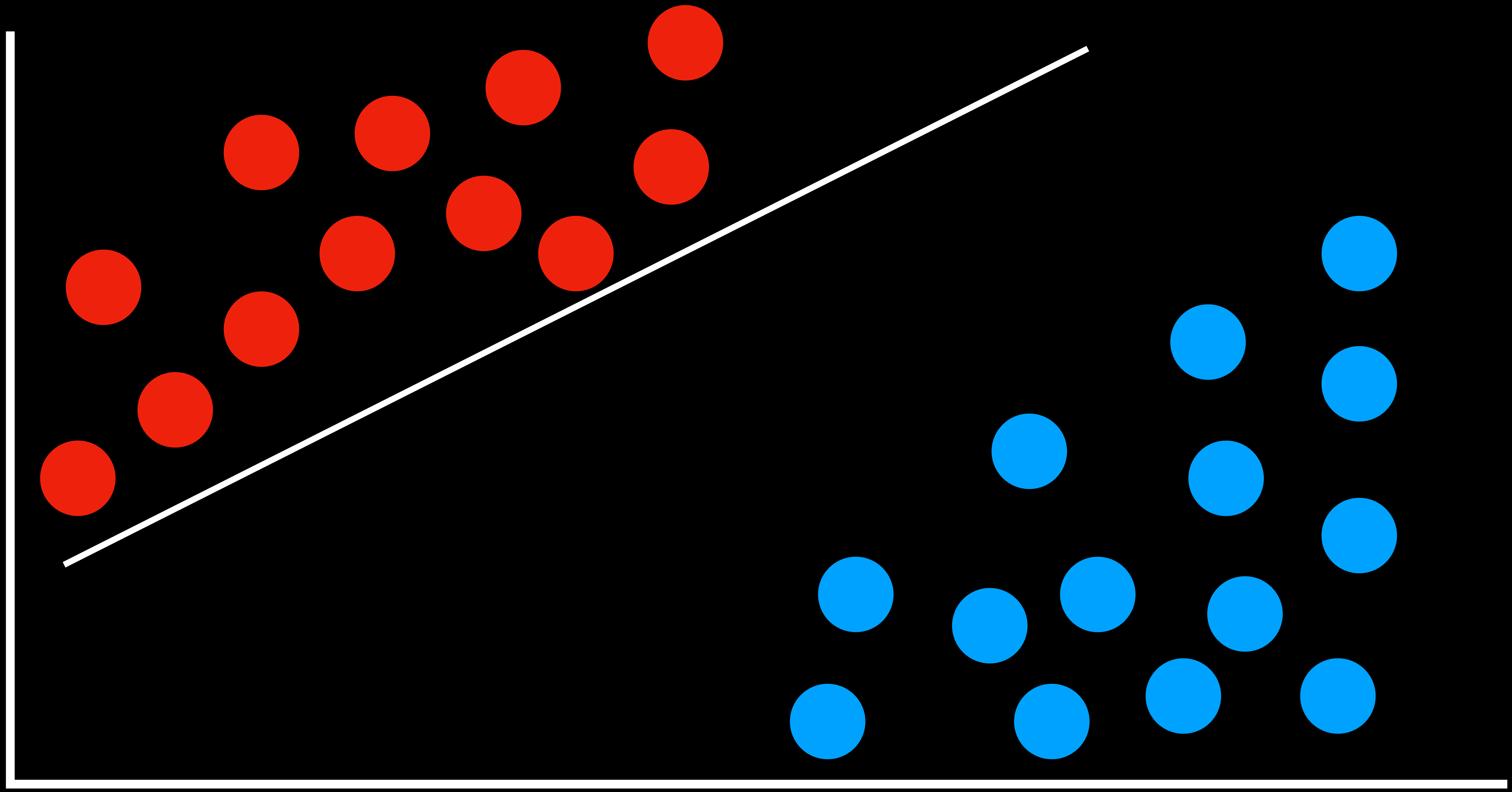
Support Vector Machines is all about finding the Optimal Hyperplane.

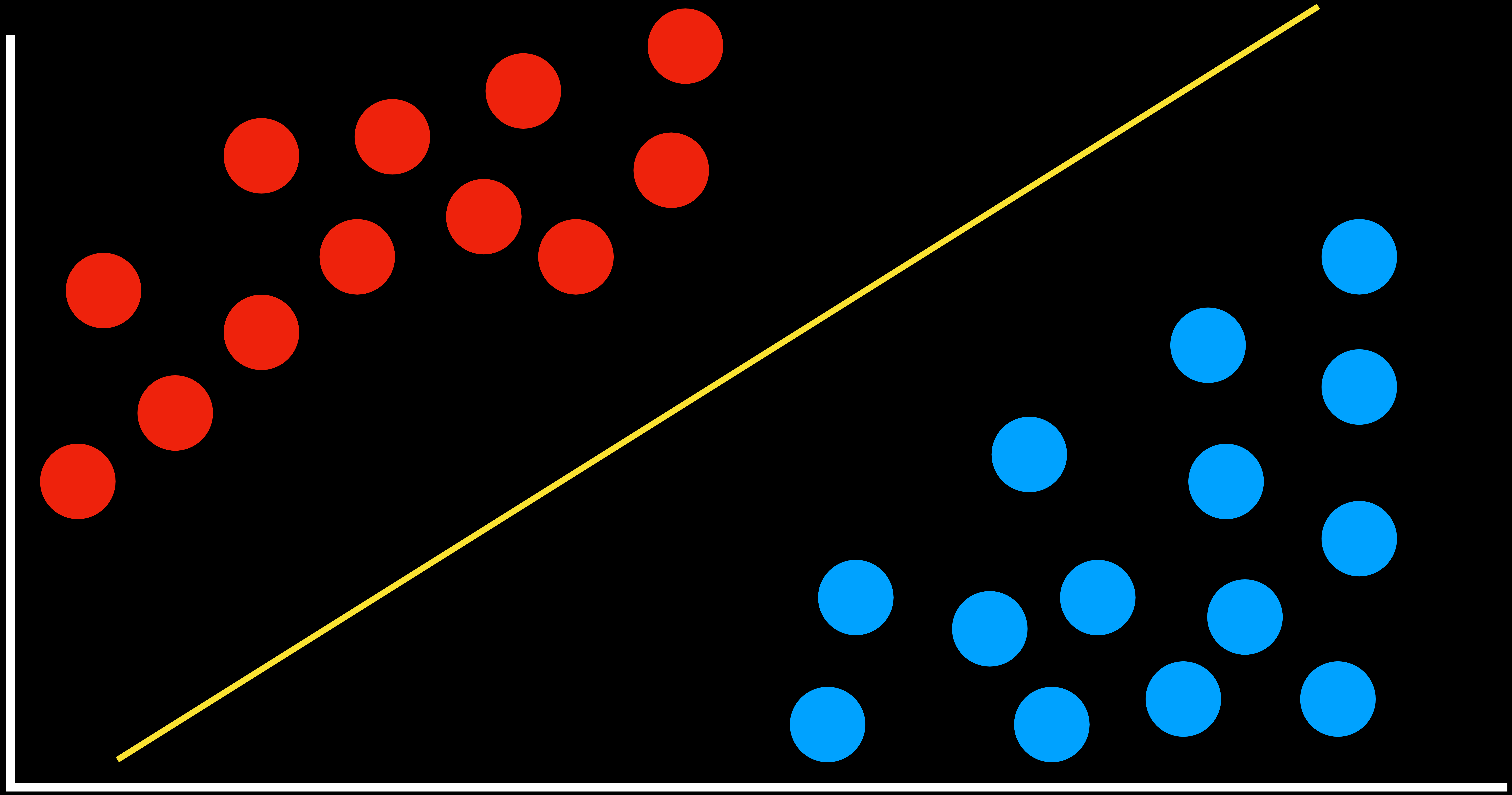
In one dimensional space the hyperplane is a point.



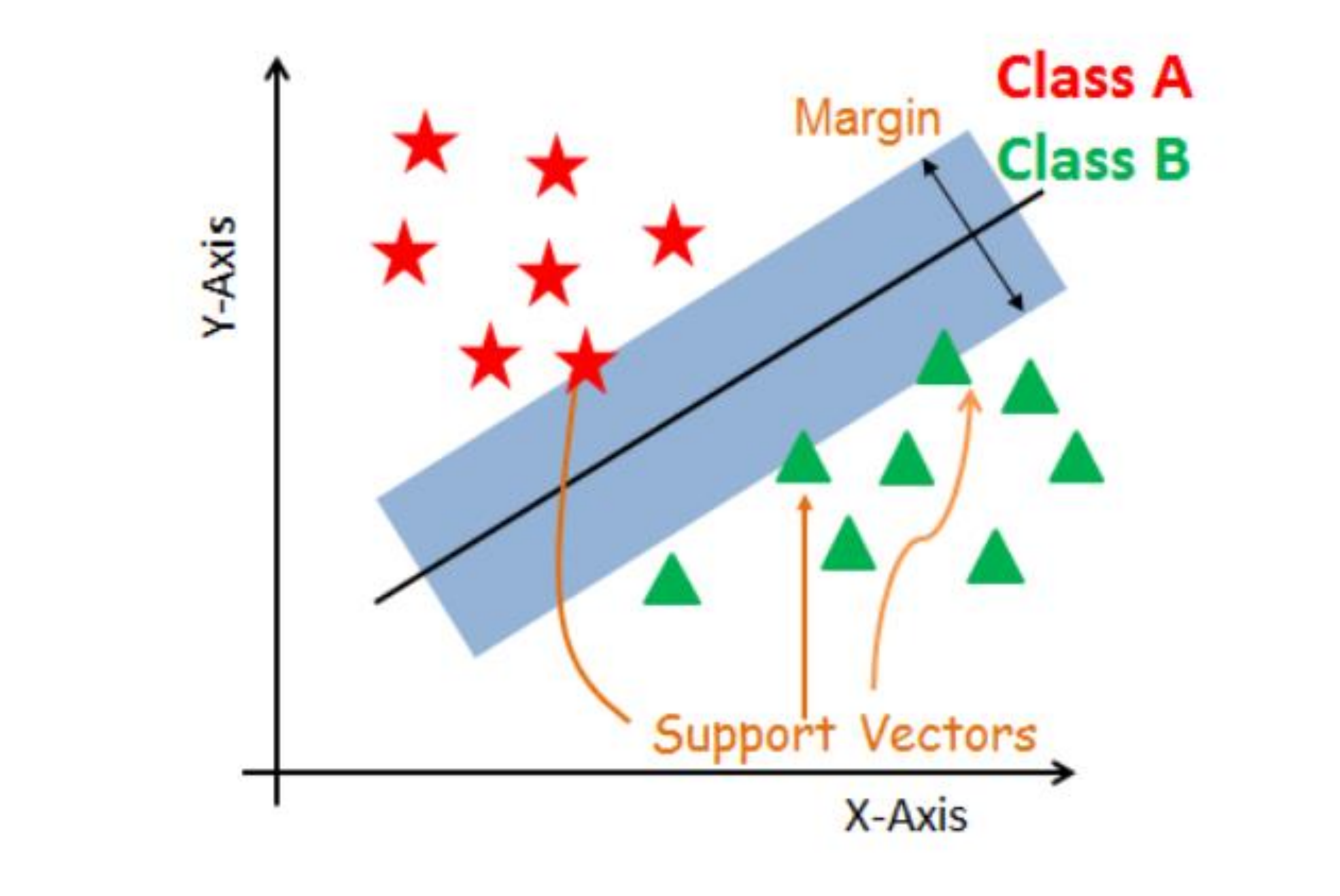
In two dimensional spaces the hyperplane is a Line.







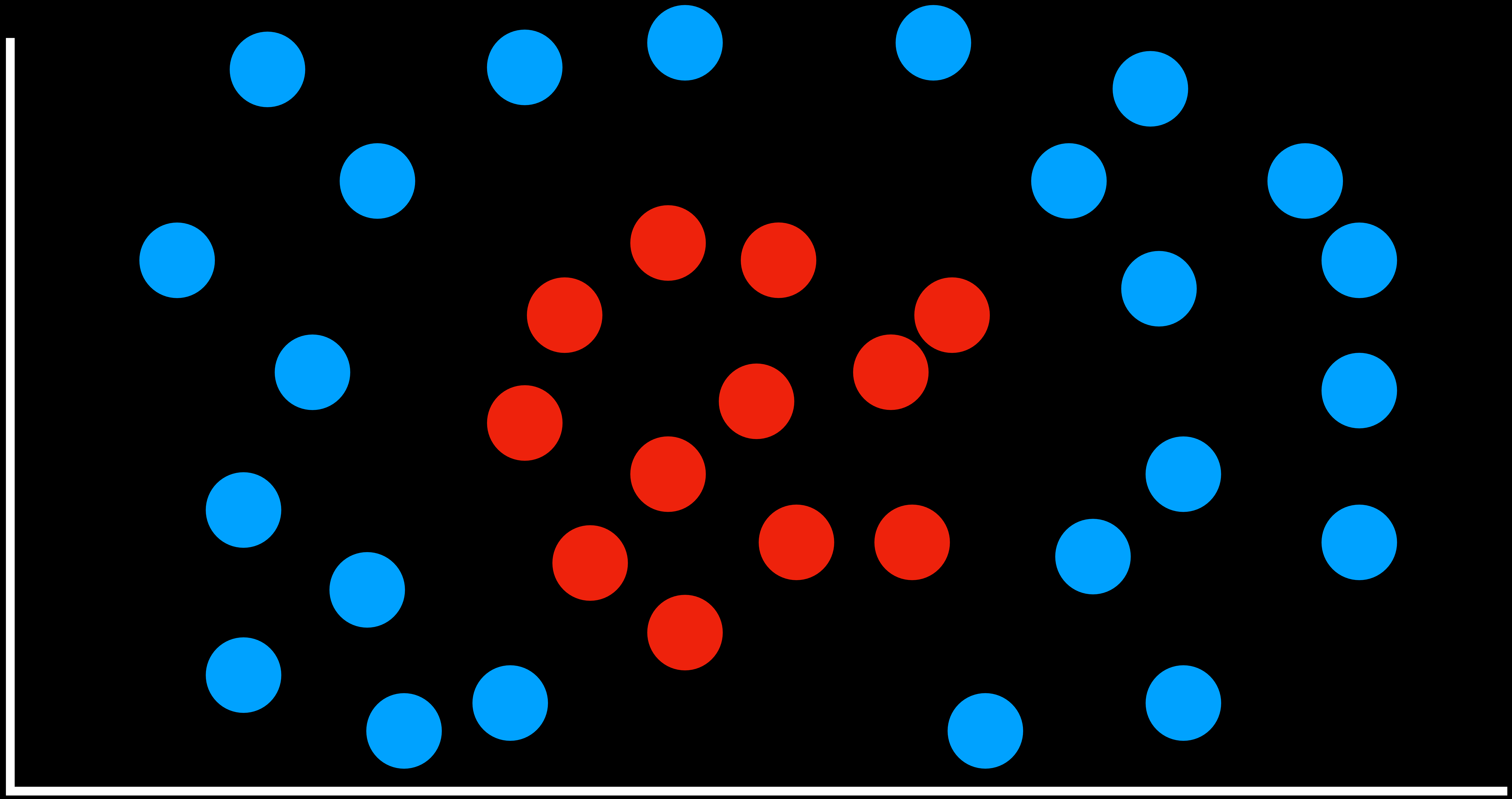
Cont..

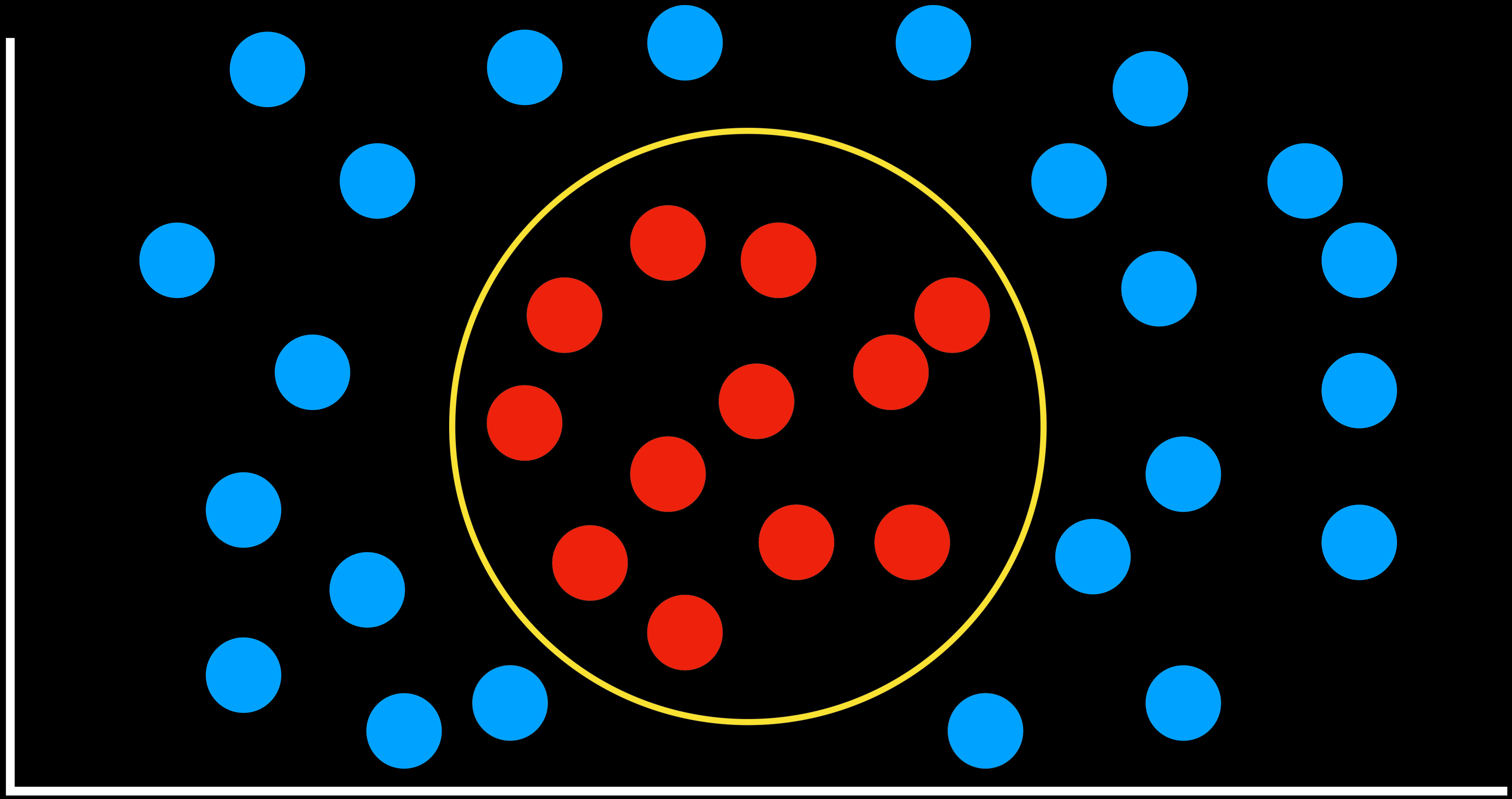


The best or optimal line that can separate the two classes is the line that as the **largest margin**. This is called the **Maximal-Margin hyperplane.**

maximum margin separator

boundary that maximizes the distance between any of the data points





Evaluating Hypotheses

loss function

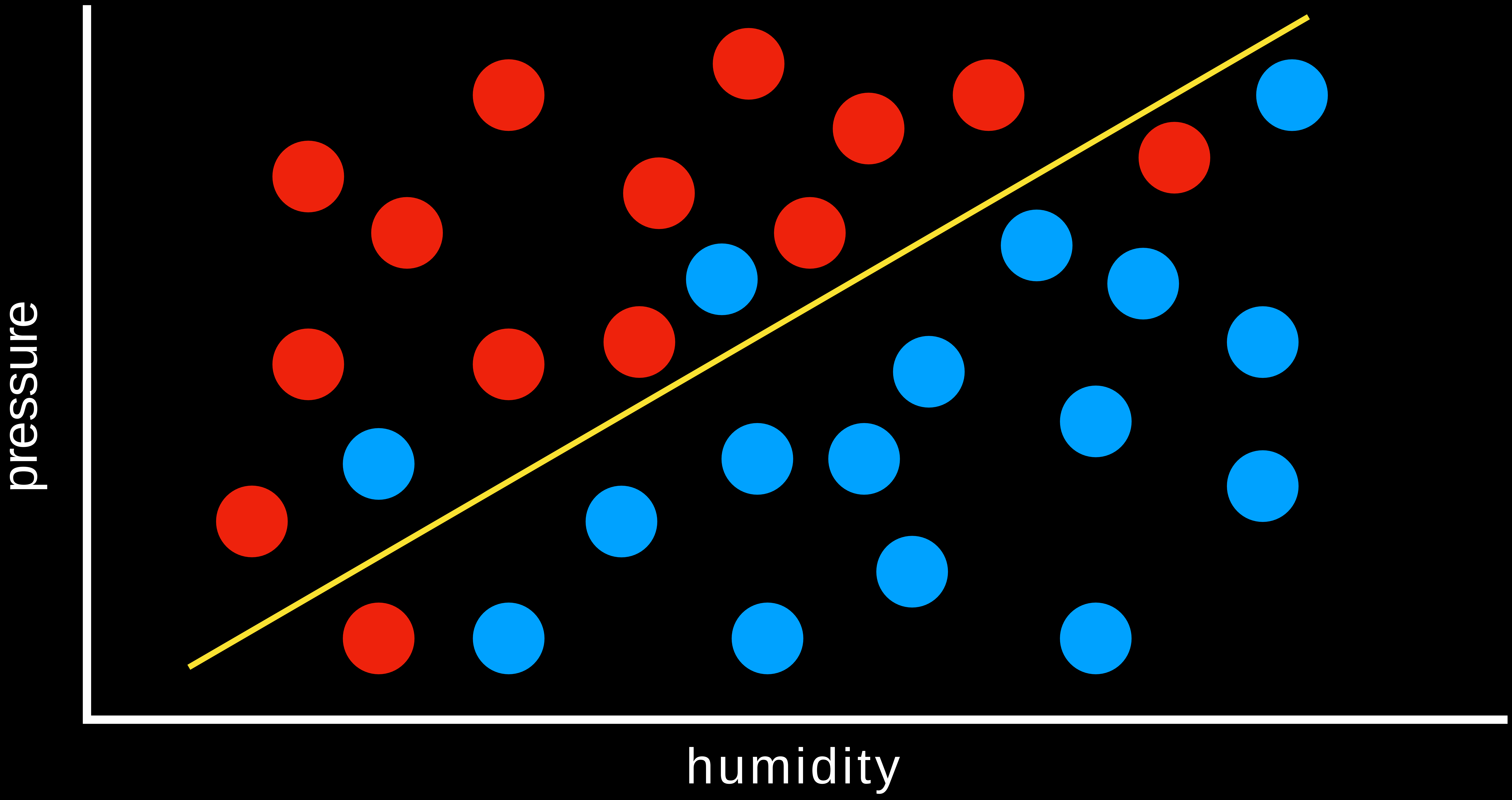
function that expresses how poorly our hypothesis performs

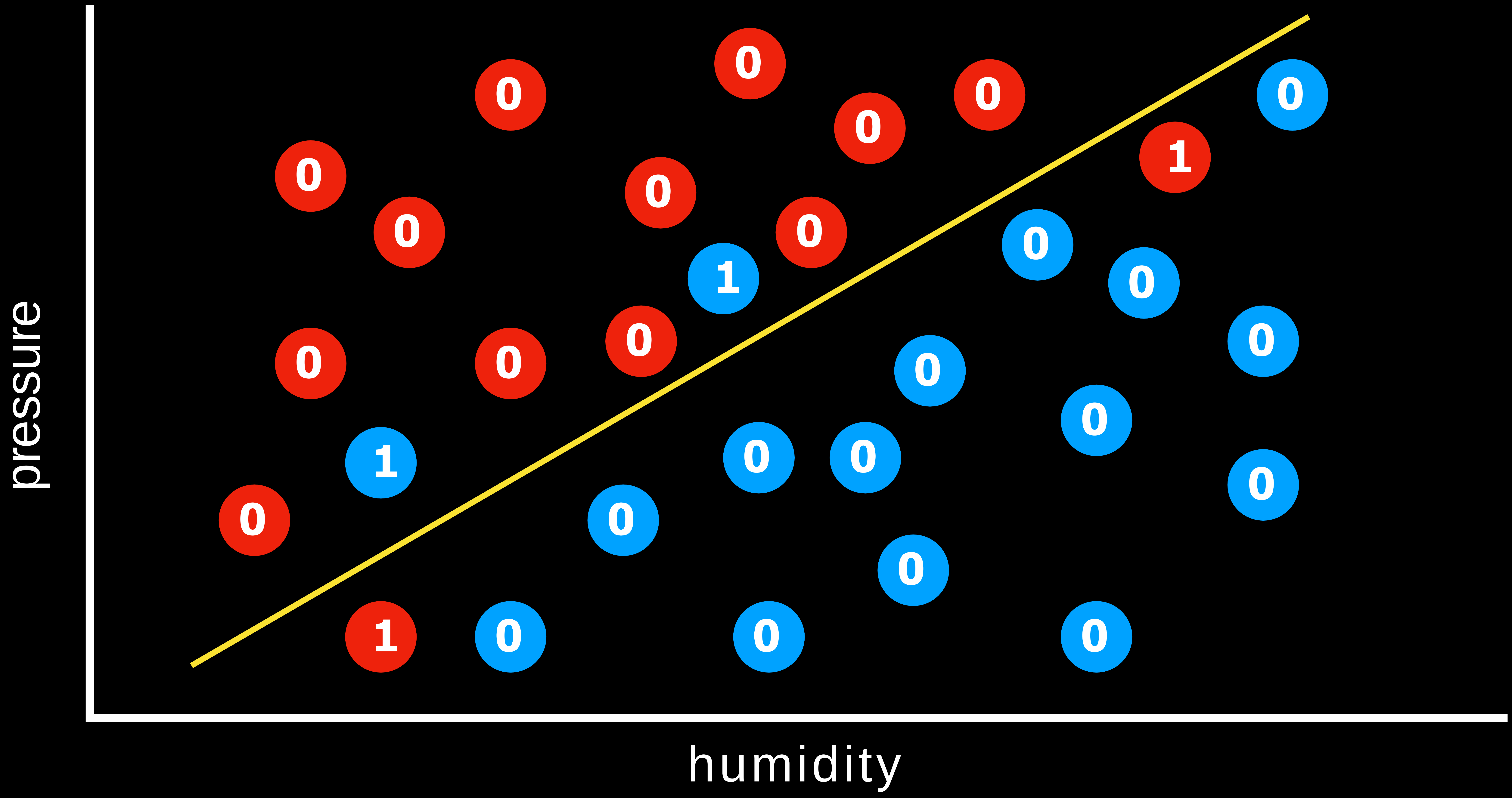
0-1 loss function

L(actual, predicted) =

0 if actual = predicted,

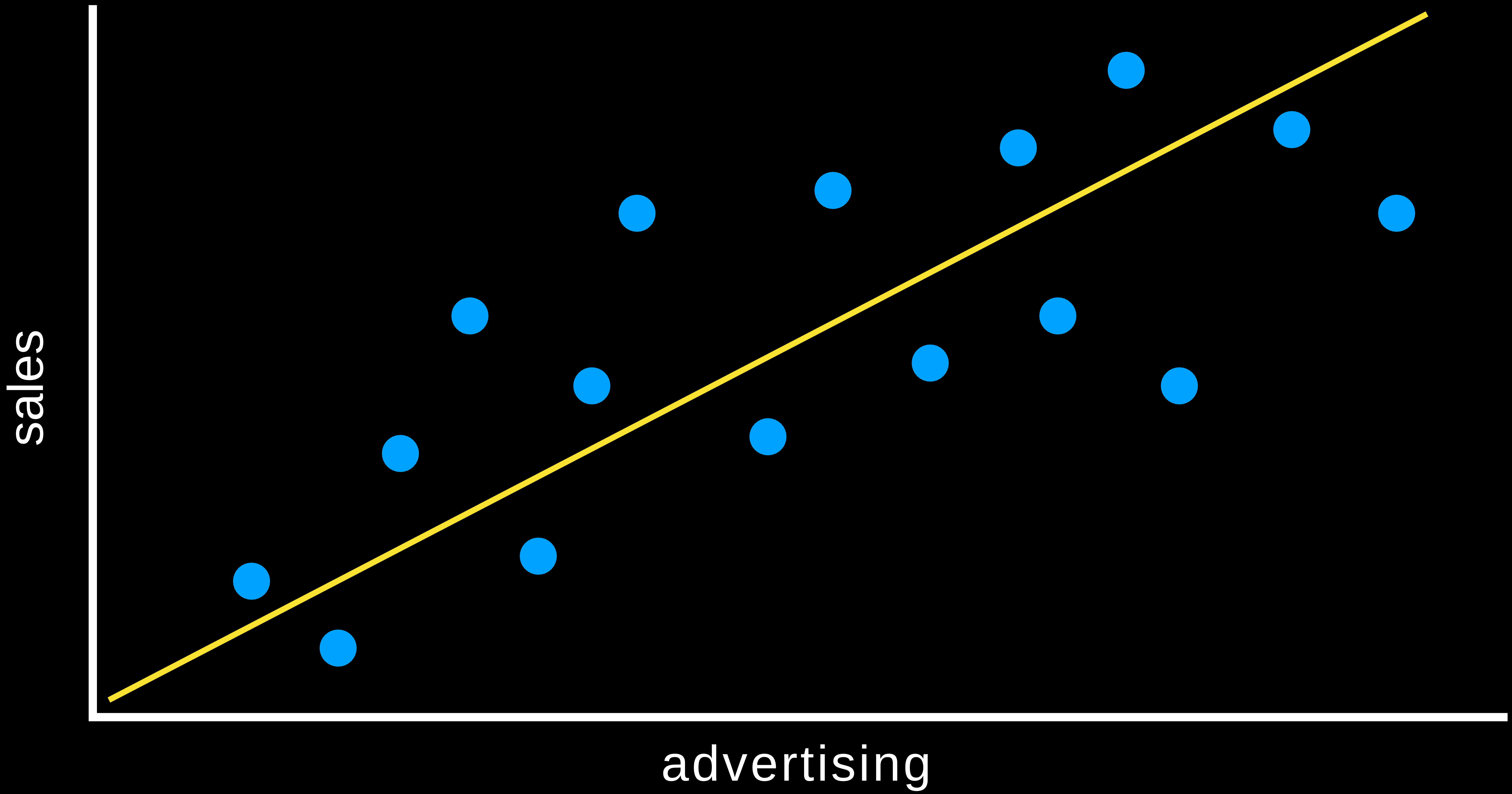
1 otherwise

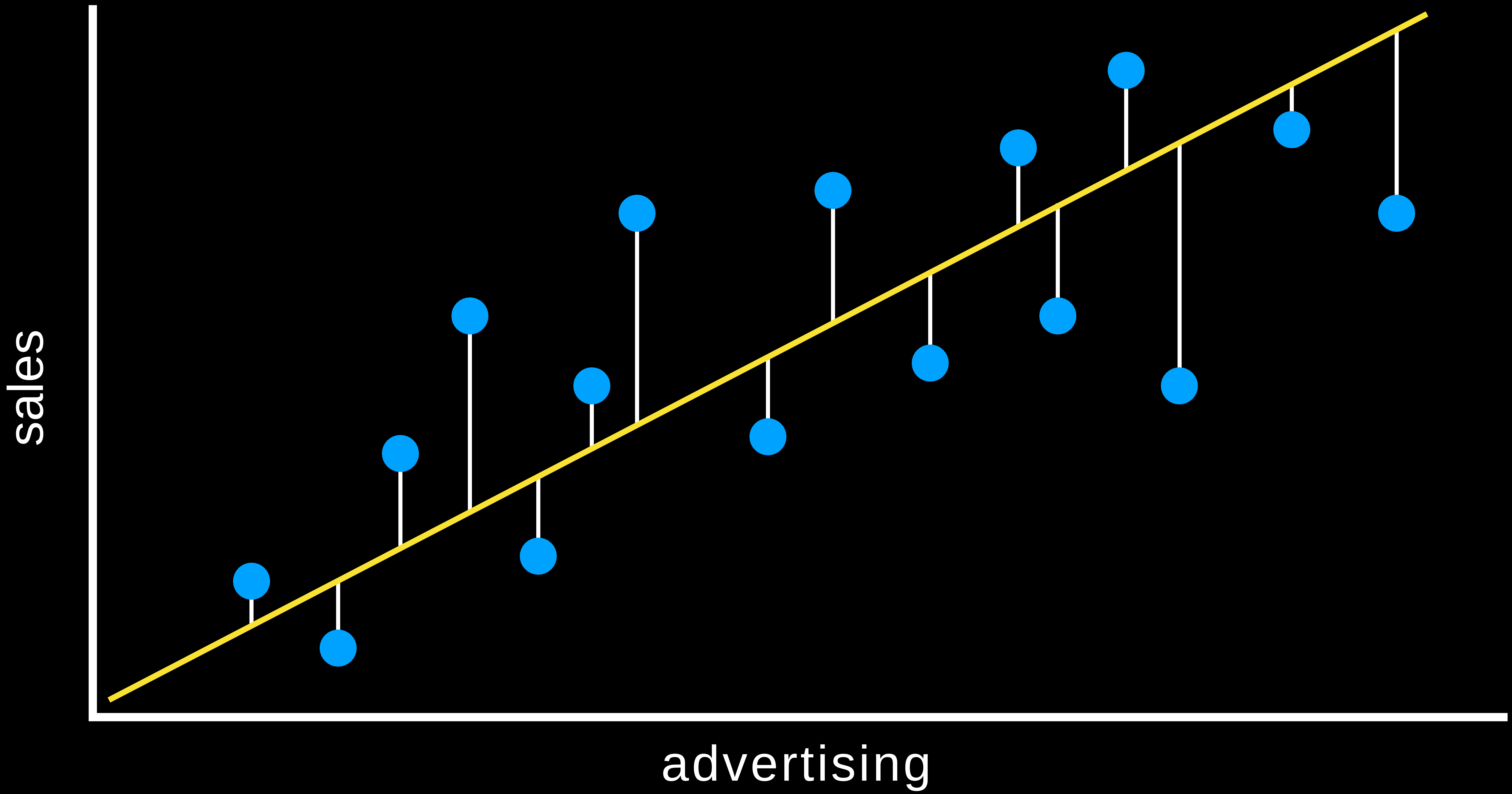

pressure
humidity

pressure
humidity
0
0
0
0
0
0
0
0
0
1
0
1
0
0
0
0
0
0
1
0
0
0
0
0
0
0
0
0
0
1
0
0
0

L_1 loss function

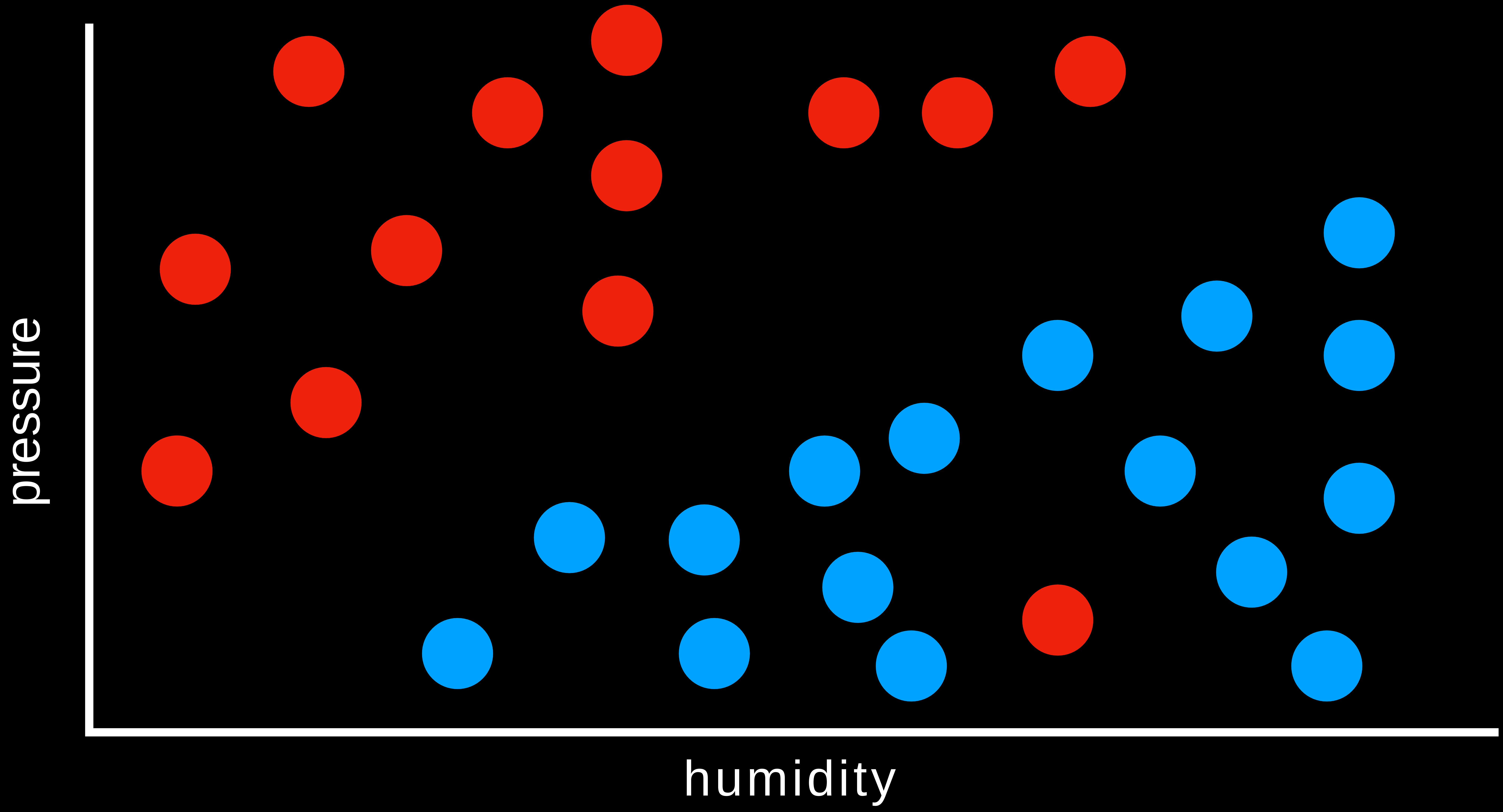
$$L(\text{actual}, \text{predicted}) = |\text{ actual - predicted }|$$

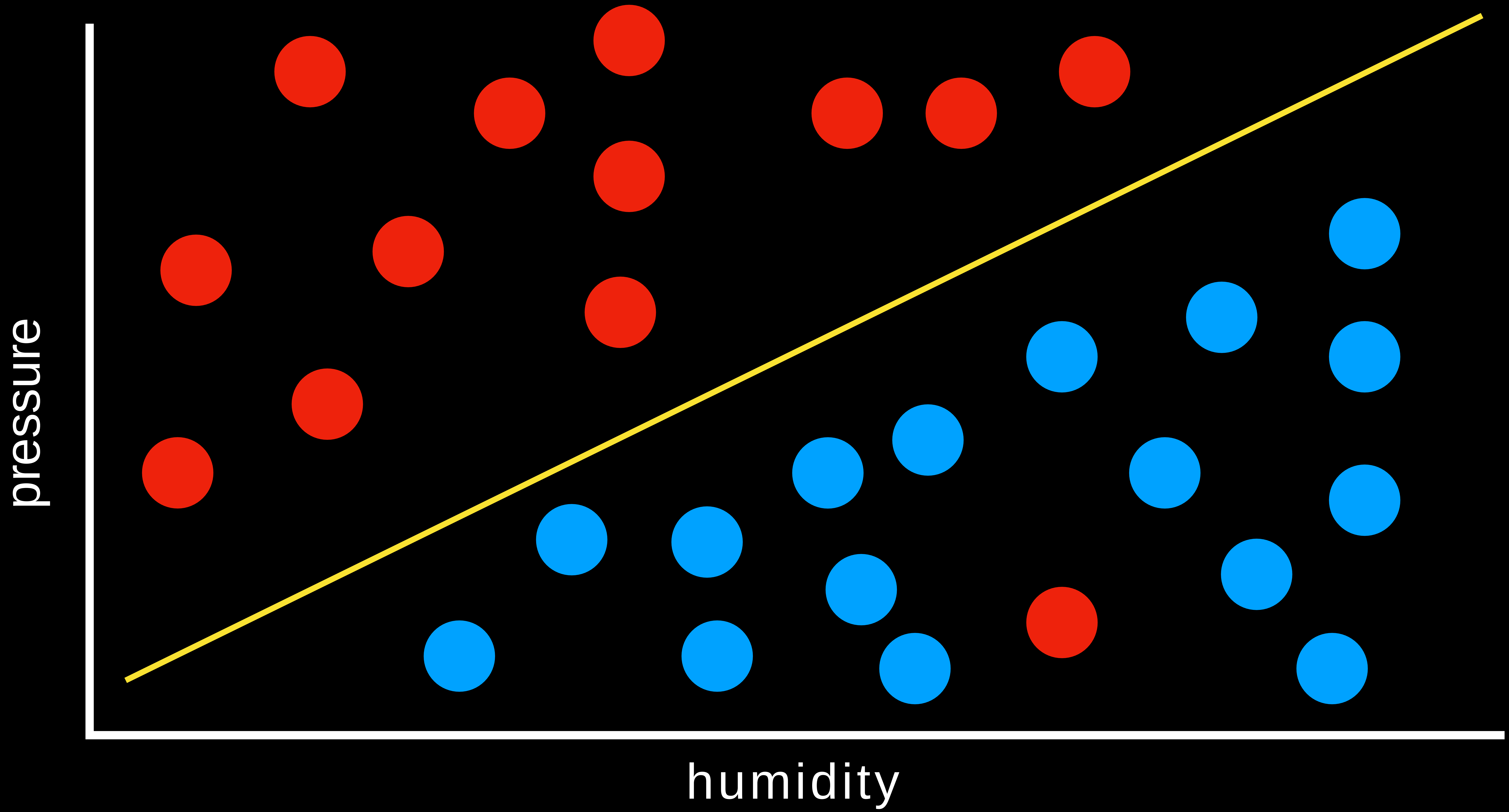

sales
advertising

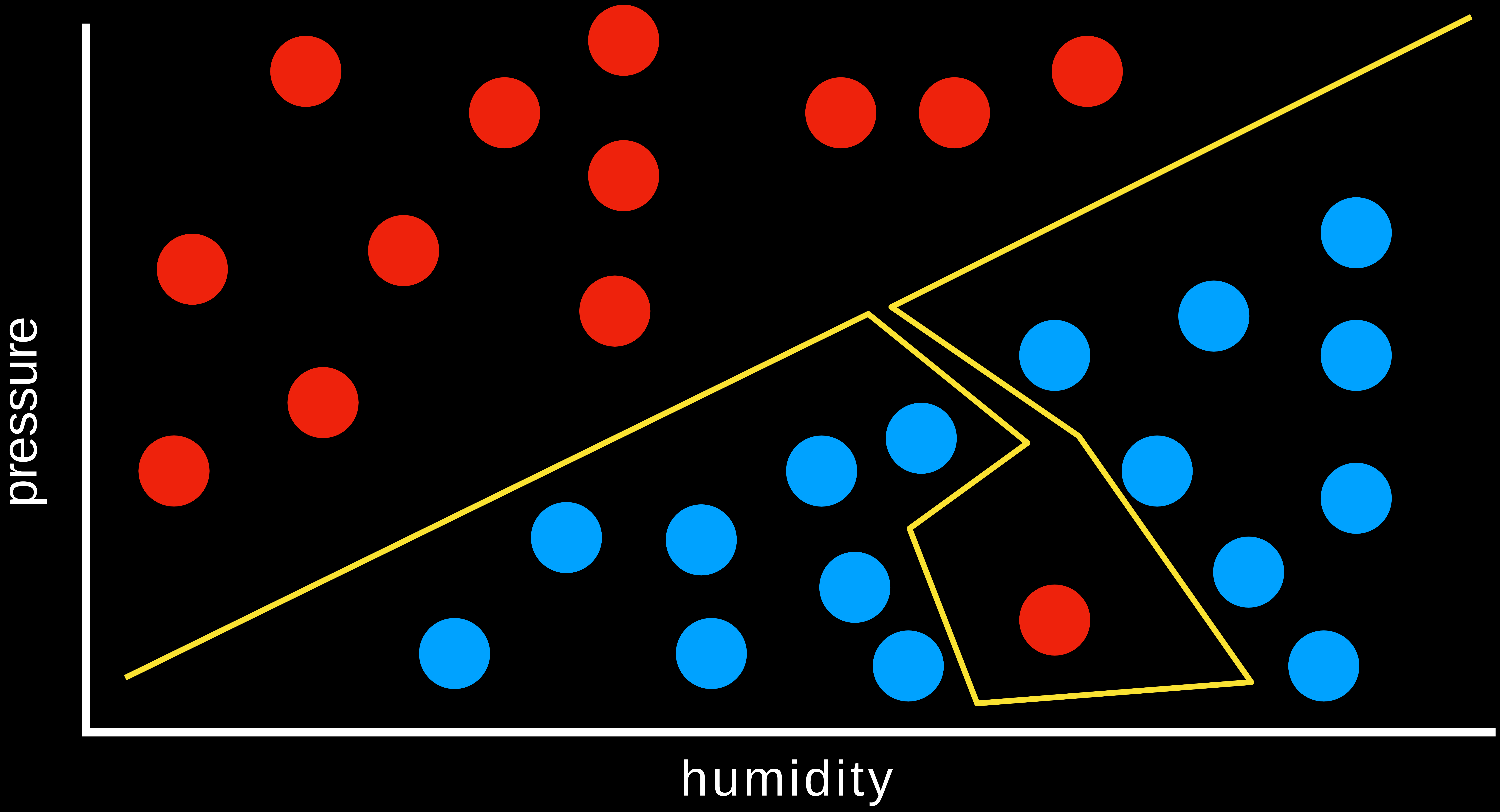

sales
advertising

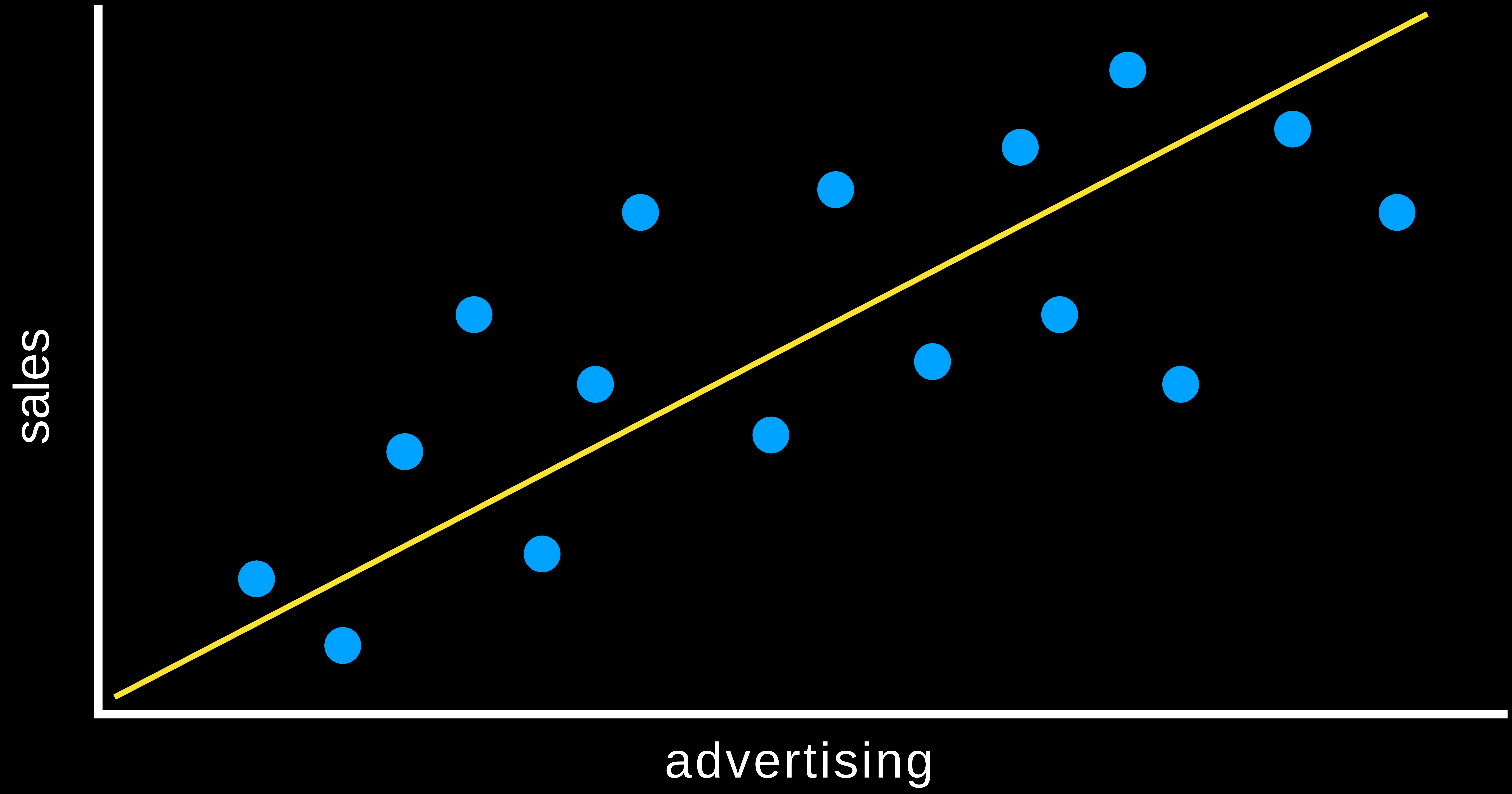
overfitting

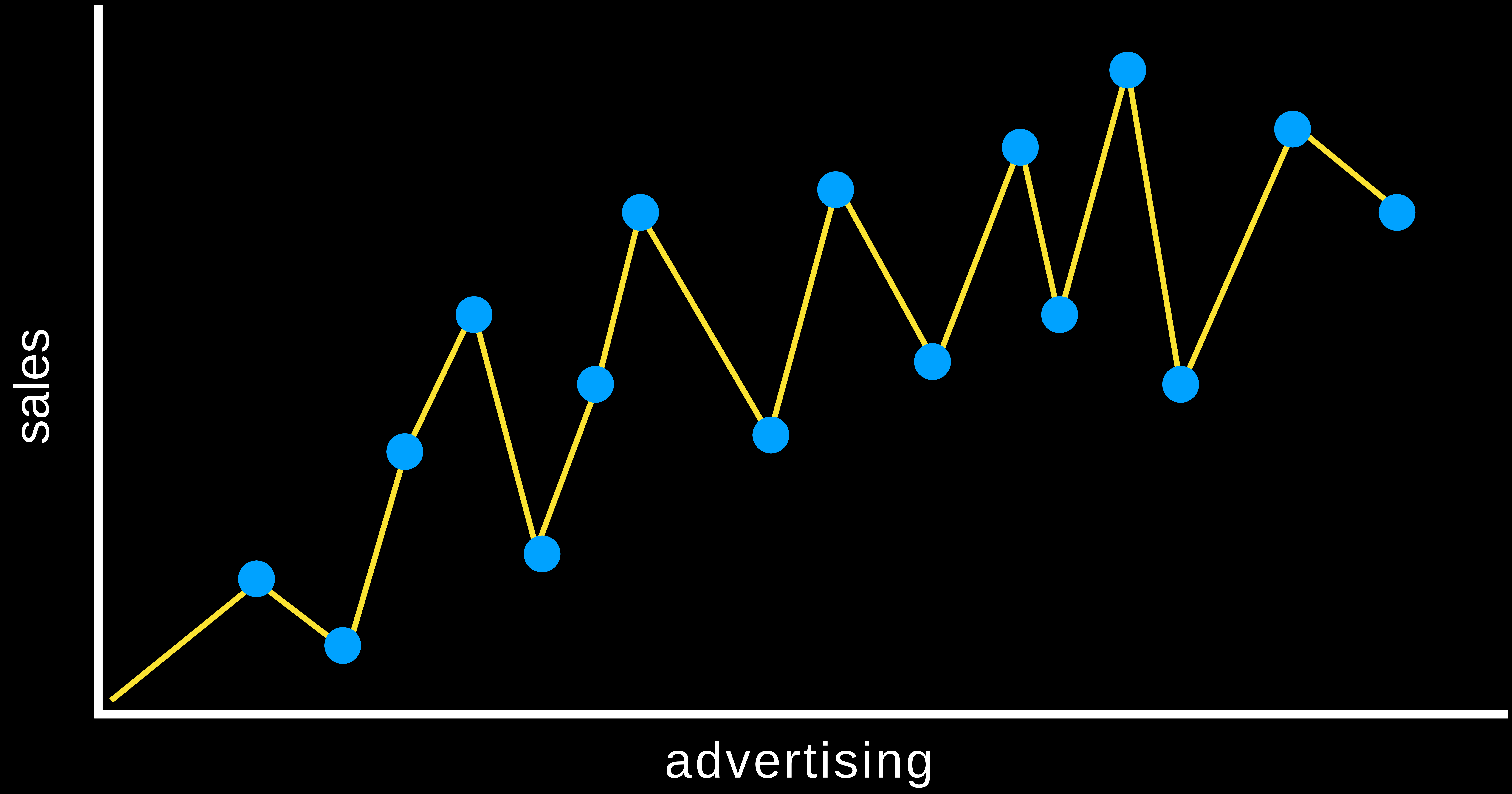
a model that fits too closely to a particular data set and therefore may fail to generalize to future data


pressure
humidity


pressure
humidity


pressure
humidity


sales
advertising


sales
advertising

holdout cross-validation

splitting data into a **training set** and a **test set**, such that learning happens on the training set and is evaluated on the test set

k-fold cross-validation

splitting data into k sets, and experimenting k times, using each set as a test set once, and using remaining data as training set

Try to learn other Supervised Machine Learning Classification Algorithms including:

- ❑ **Gaussian Naïve Bayes**
- ❑ **Perceptron**
- ❑ **Decision Tree**
- ❑ **Random Forest**
- ❑ **Others**

scikit-learn

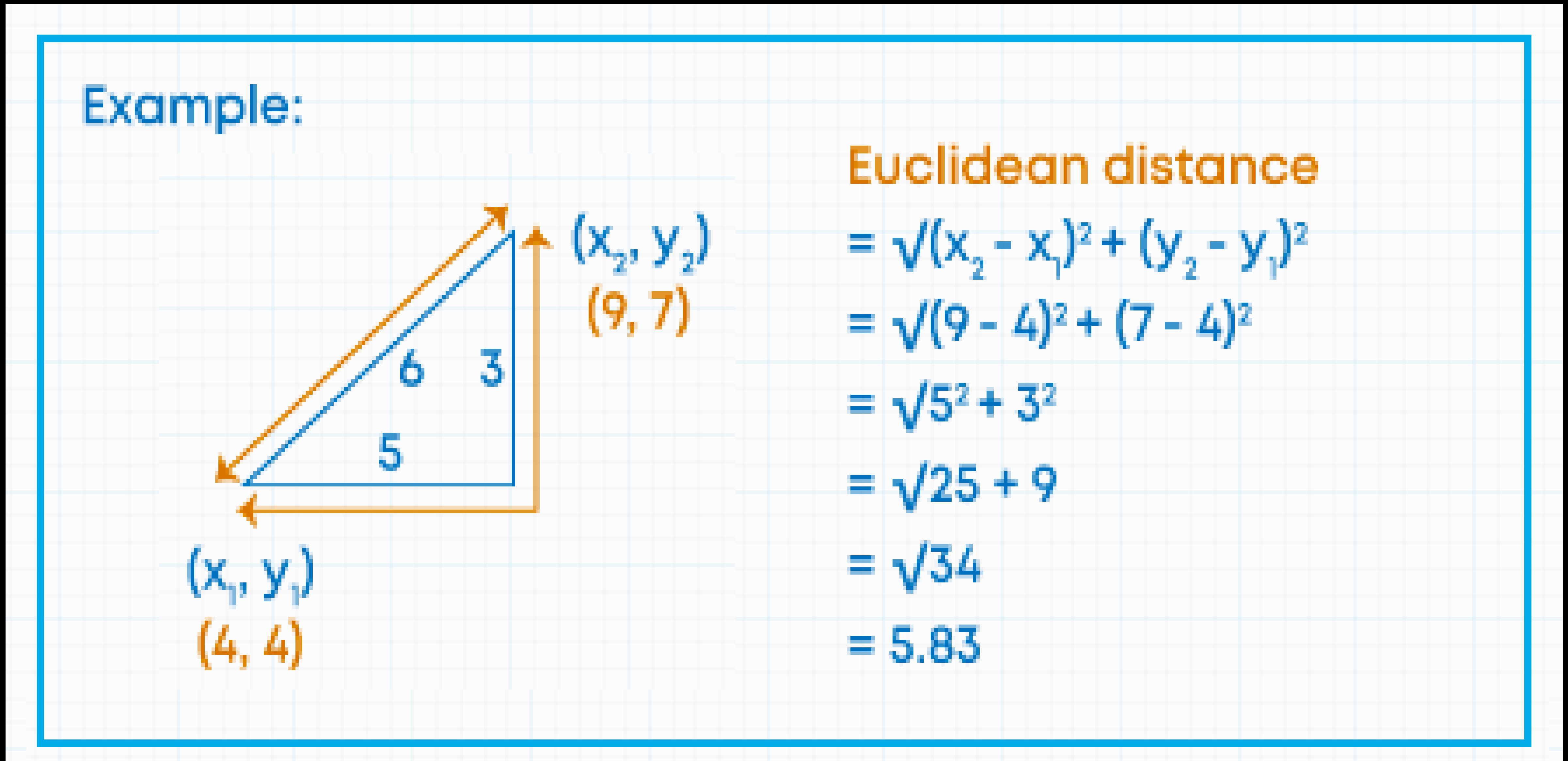
Unsupervised Learning

unsupervised learning

given input data without any additional feedback, learn patterns

Distance Metric (Measure)

- Euclidean Distance (ED)
  - The ED is the most widely used distance measure when the variables are continuous (either interval or ratio scale).
  - The ED between two points calculates the length of a segment connecting the two points. It is the most evident way of representing the distance between two points.



Example:

Euclidean distance

$$= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$= \sqrt{(9 - 4)^2 + (7 - 4)^2}$$

$$= \sqrt{5^2 + 3^2}$$

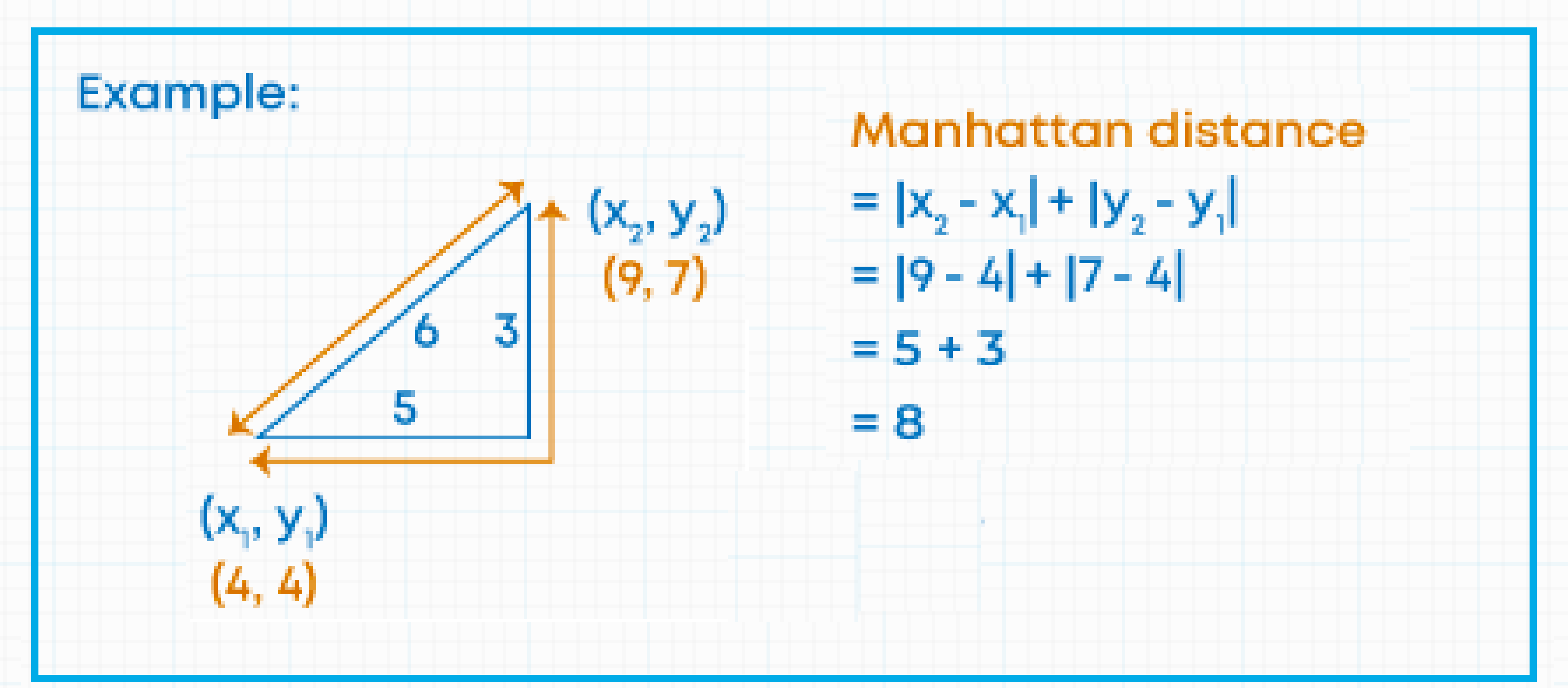
$$= \sqrt{25 + 9}$$

$$= \sqrt{34}$$

$$= 5.83$$

Cont..

- Manhattan Distance (MD)
  - ❑ The distance between two points in a grid-based on a strictly horizontal and vertical path. The Manhattan distance is the simple sum of the horizontal and vertical components.
  - ❑ In nutshell, we can say Manhattan distance is the distance if you had to travel along coordinates only.



Example:

Manhattan distance

$= |x_2 - x_1| + |y_2 - y_1|$

$= |9 - 4| + |7 - 4|$

$= 5 + 3$

$= 8$

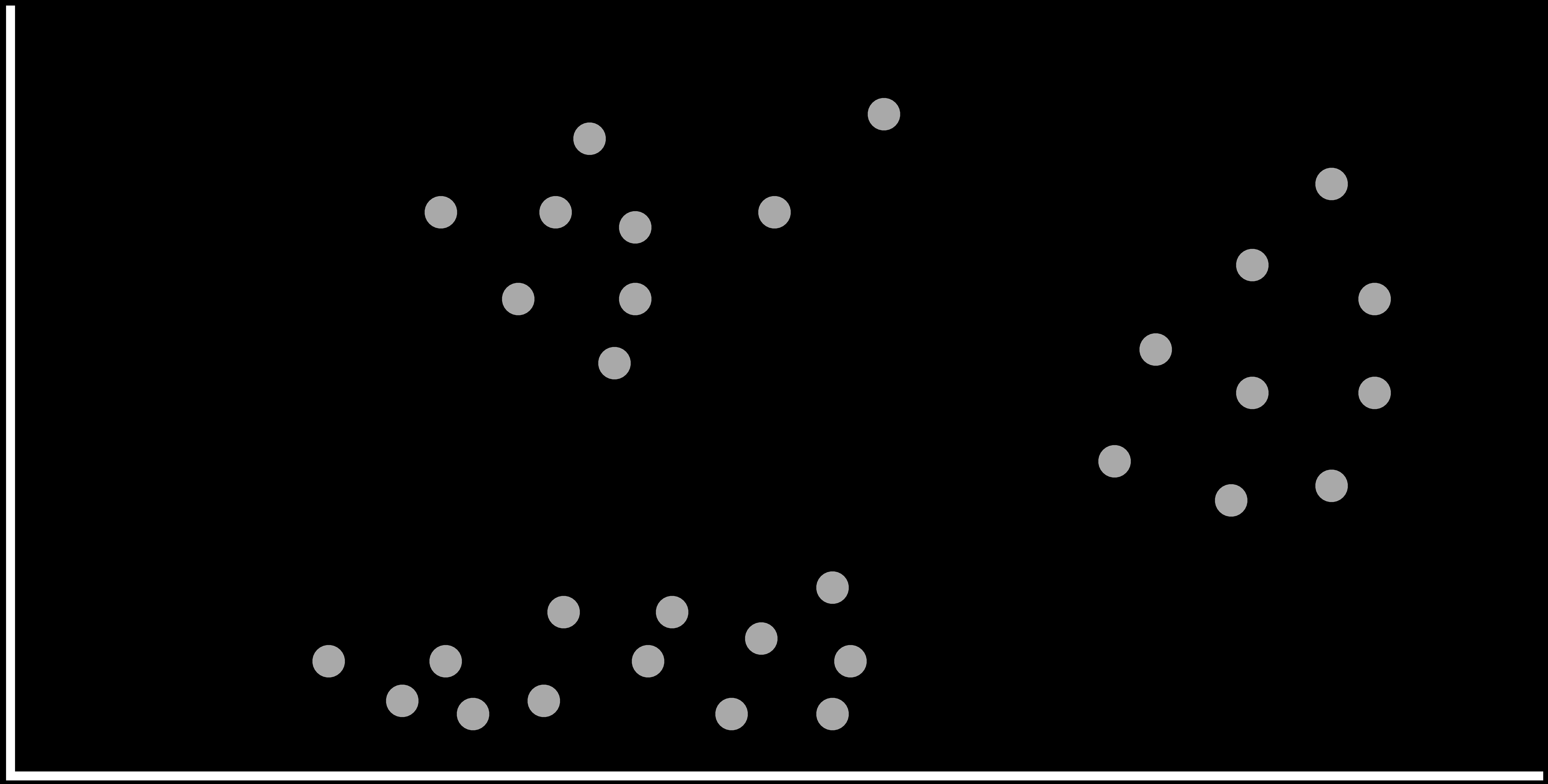
Clustering

clustering

organizing a set of objects into groups in such a way that similar objects tend to be in the same group

Some Clustering Applications

- Genetic research
- Image segmentation
- Market research
- Medical imaging
- Social network analysis.

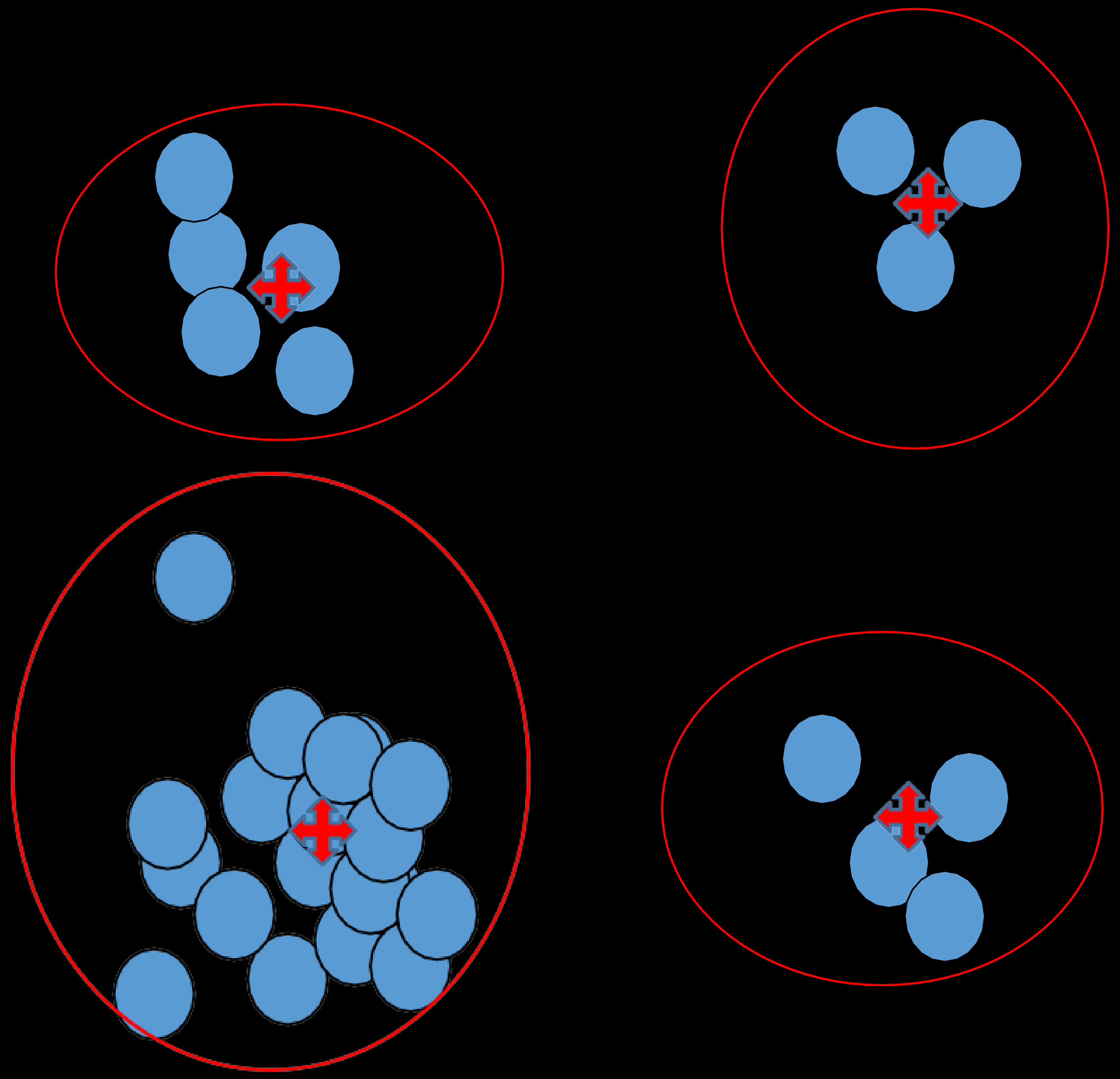


Clustering

- Clustering is a technique for finding similarity groups in data, called **clusters**. I.e.,
    - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning

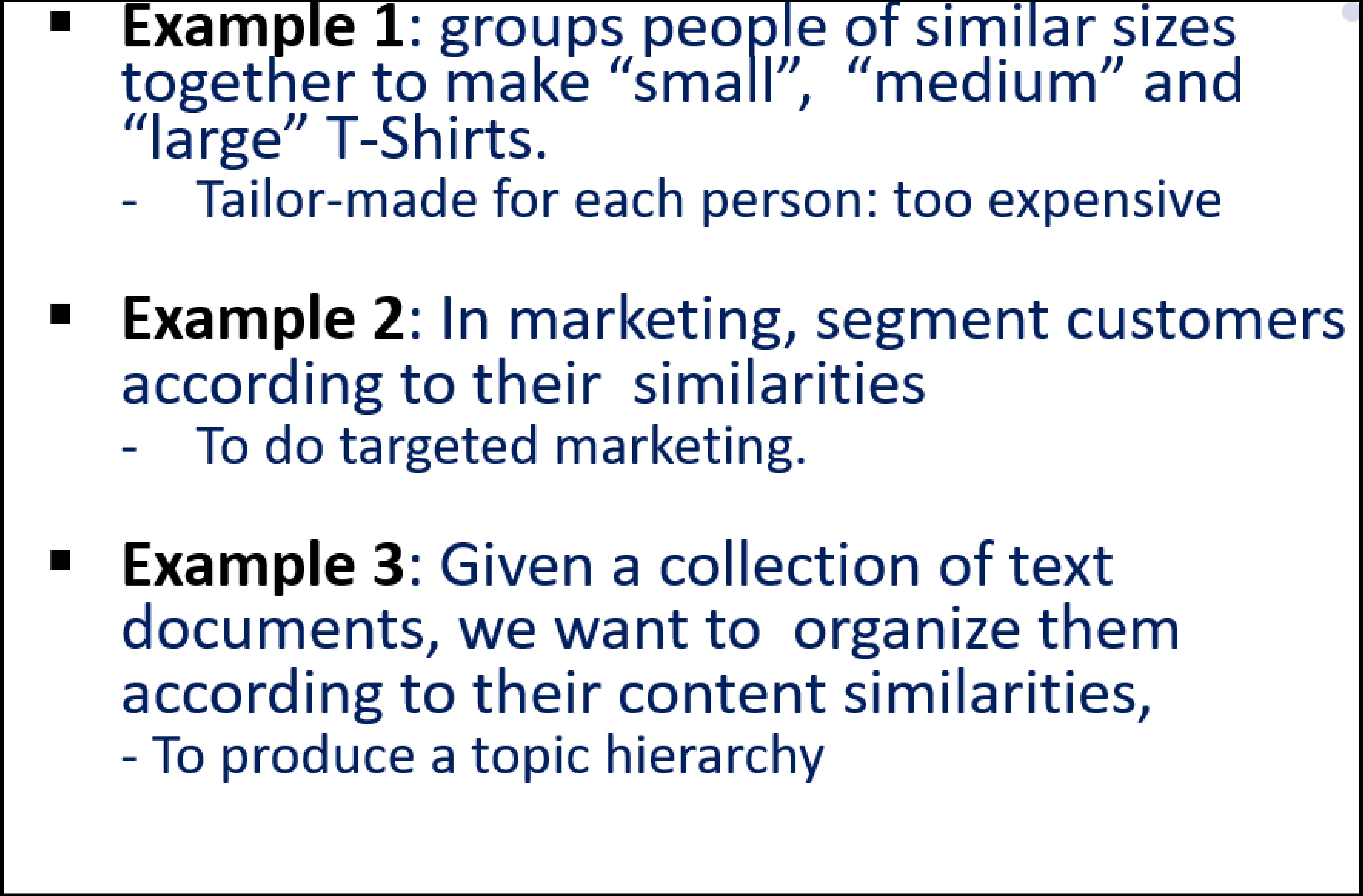
Cont..

This below data set has four natural groups of data points, i.e., 4 natural clusters.



- ❖ A cluster is represented by a single point, known as centroid (or cluster center) of the cluster.
- ❖ Centroid is computed as the mean of all data points in a cluster
- ❖ Cluster boundary is decided by the farthest data point in the cluster.

Cont..



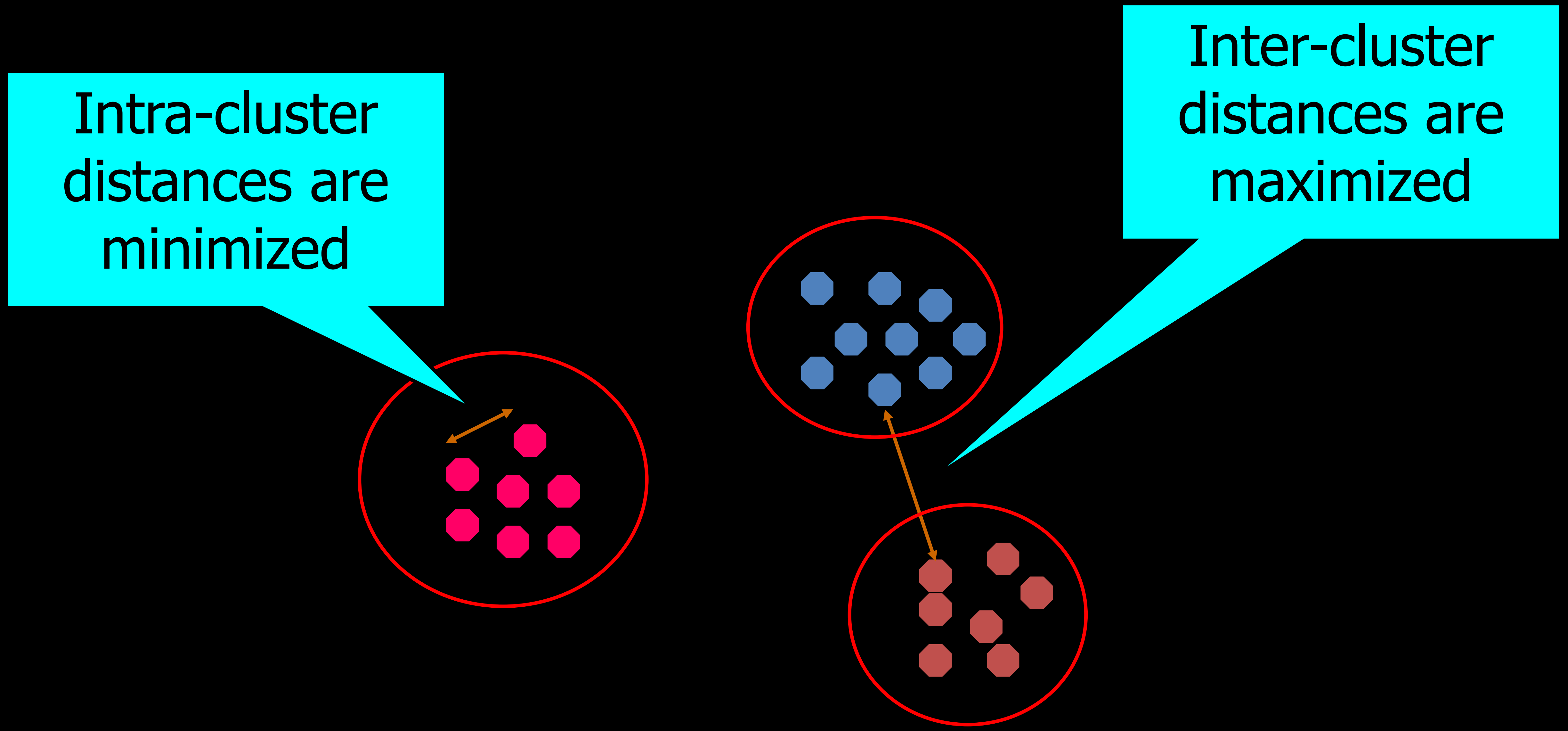
- **Example 1**: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
  - Tailor-made for each person: too expensive

- **Example 2**: In marketing, segment customers according to their similarities
  - To do targeted marketing.

- **Example 3**: Given a collection of text documents, we want to organize them according to their content similarities,
  - To produce a topic hierarchy

What is Cluster Analysis?

Finding groups of objects in data such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Intra-cluster distances are minimized

Inter-cluster distances are maximized

Types of Clustering

❑ **Partitional Clustering**

K-Means Clustering

K-Medoid Clustering

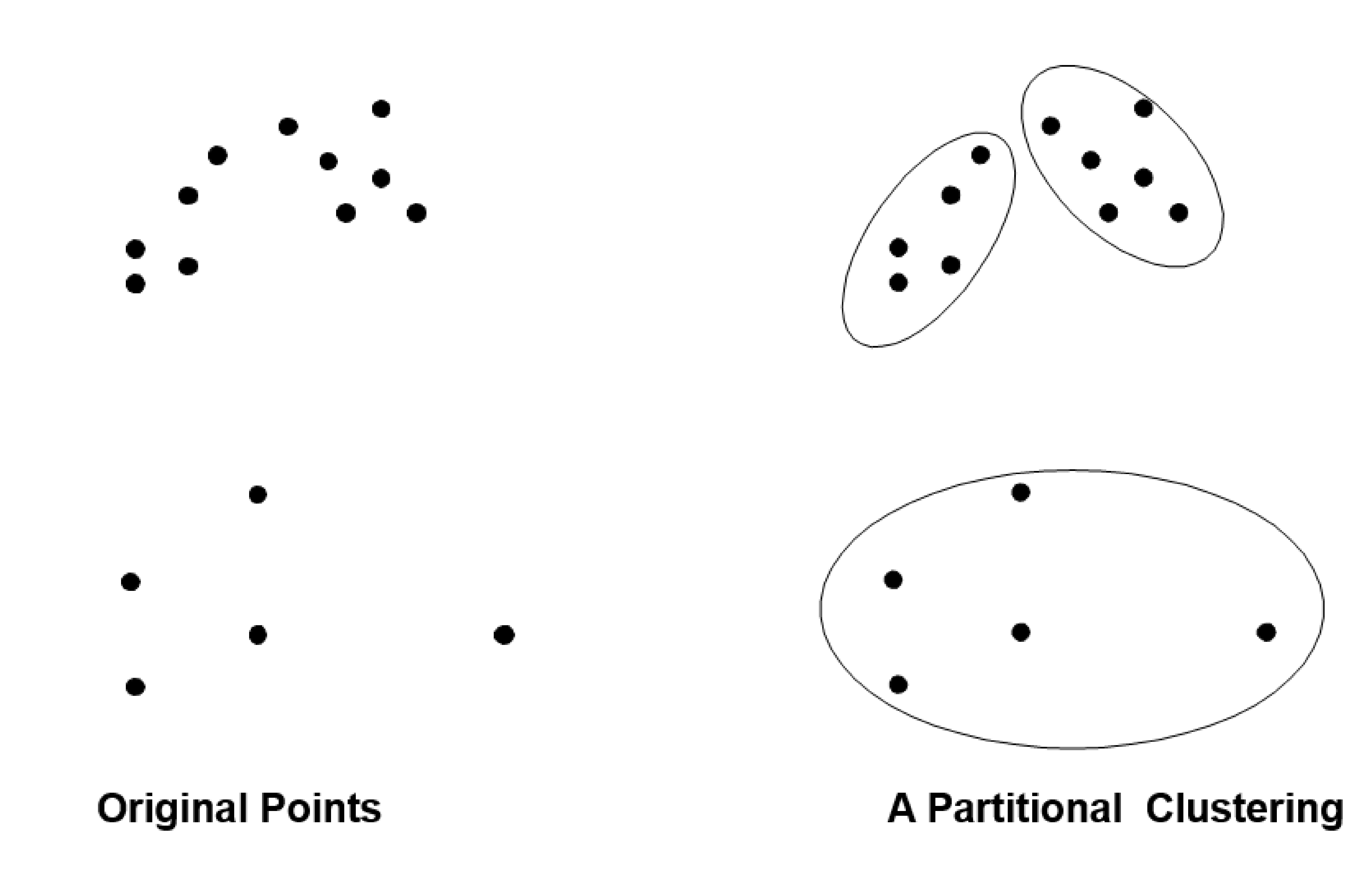
Spectral Clustering

❑ **Hierarchical Clustering**

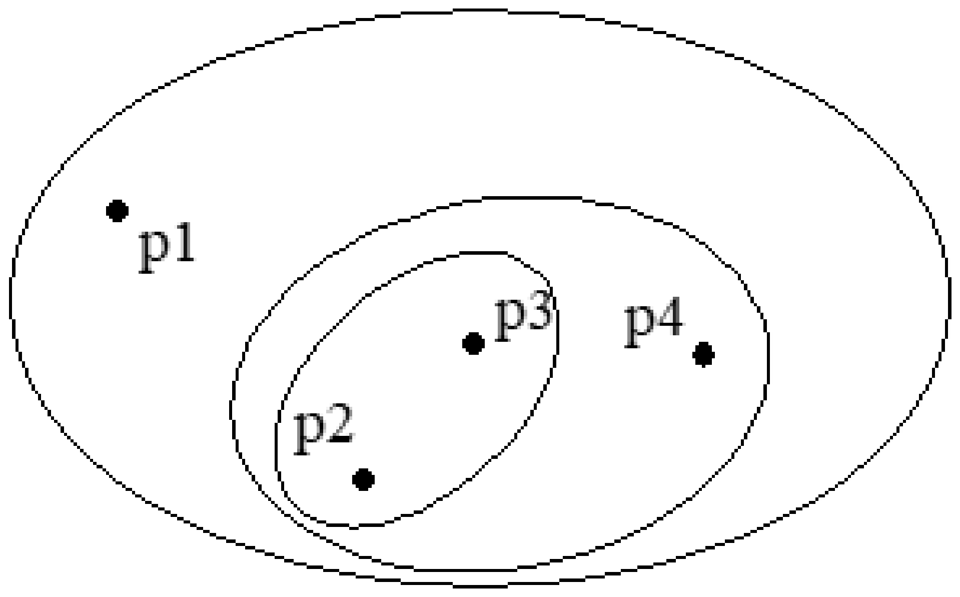
Agglomerative Clustering (Bottom Up)

Divisive Clustering (Top-Down)

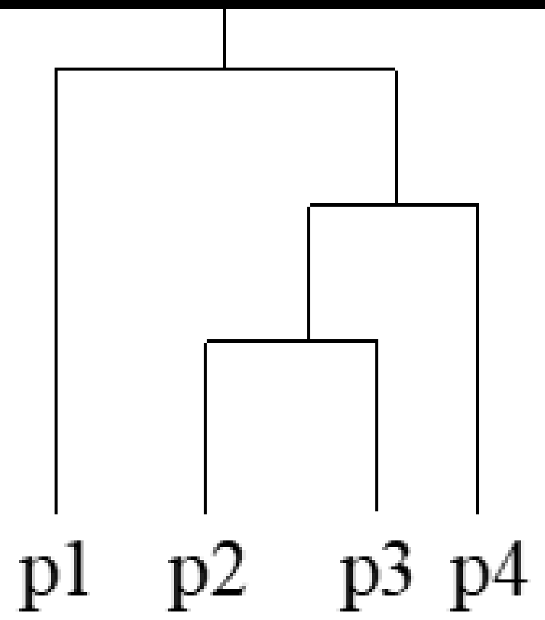
Partitional Clustering



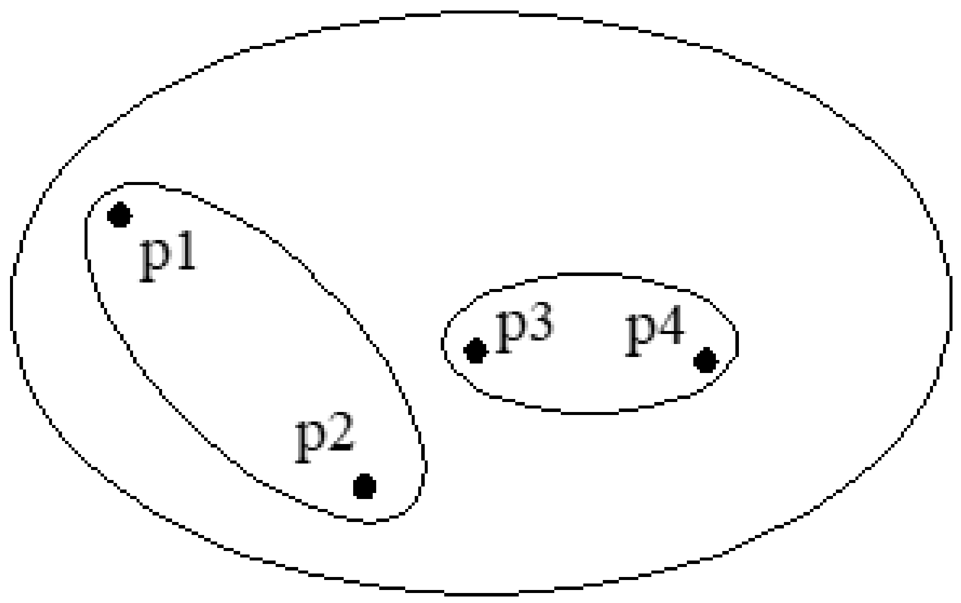
Hierarchical Clustering



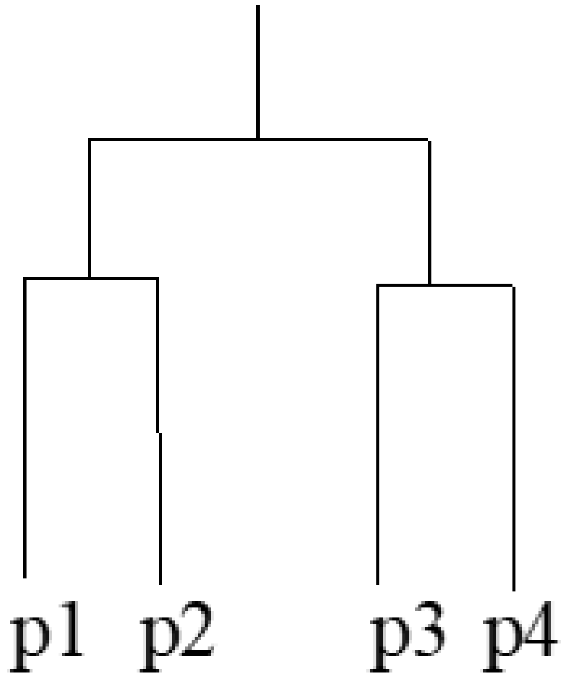
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

k-means clustering

algorithm for clustering data based on repeatedly assigning points to clusters and updating those clusters' centers

K-Means Clustering

K-means is a partitional clustering algorithm

Let the set of data points (or instances) *D* be

$$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\},$$

where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.

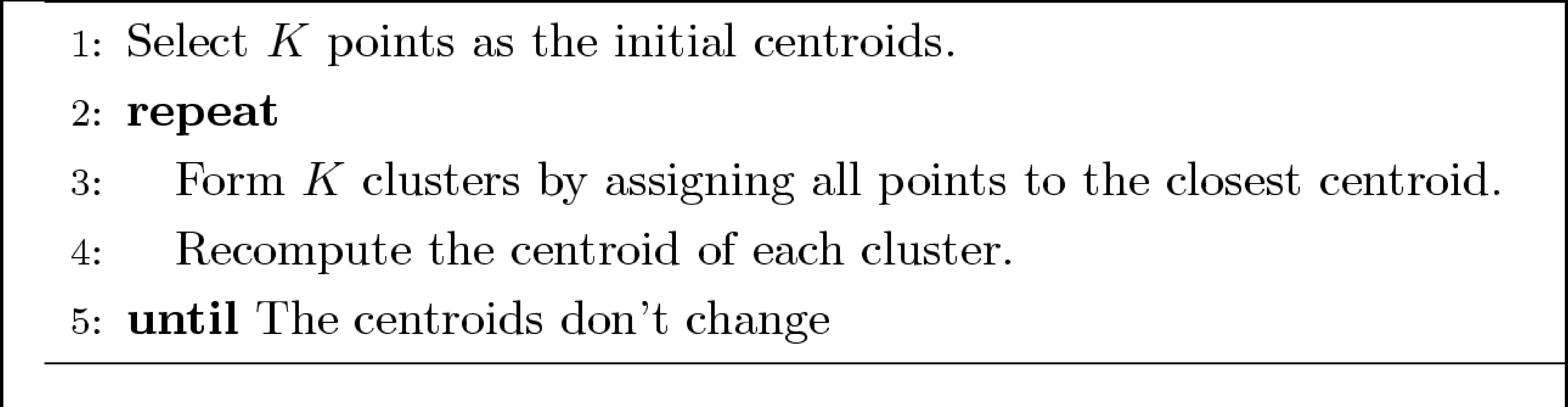
The *k*-means algorithm partitions the given data into *k* clusters.

Each cluster has a cluster **center**, called **centroid**.

k is specified by the user

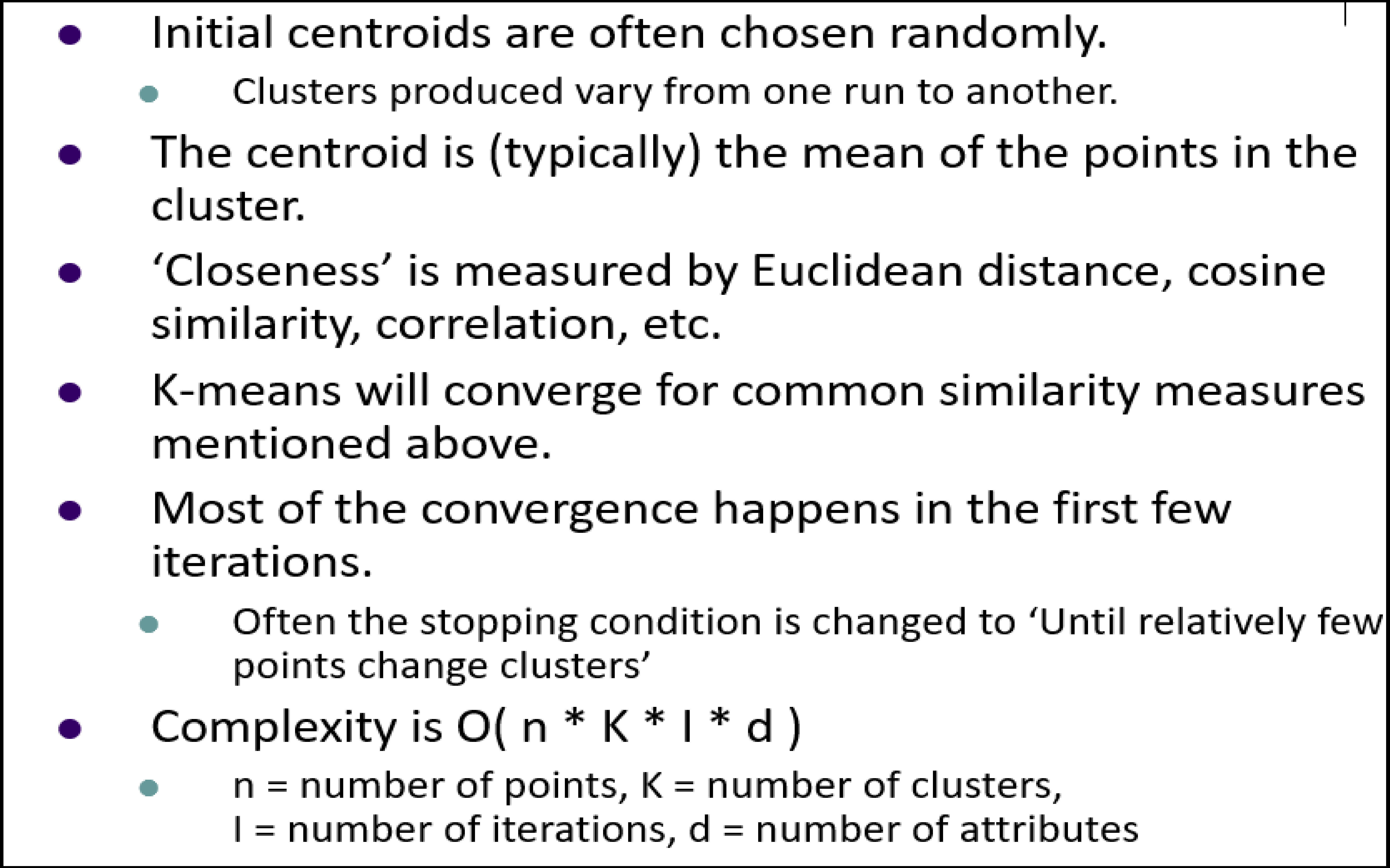
K-means Clustering

Basic algorithm



```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

K-means Clustering – Details



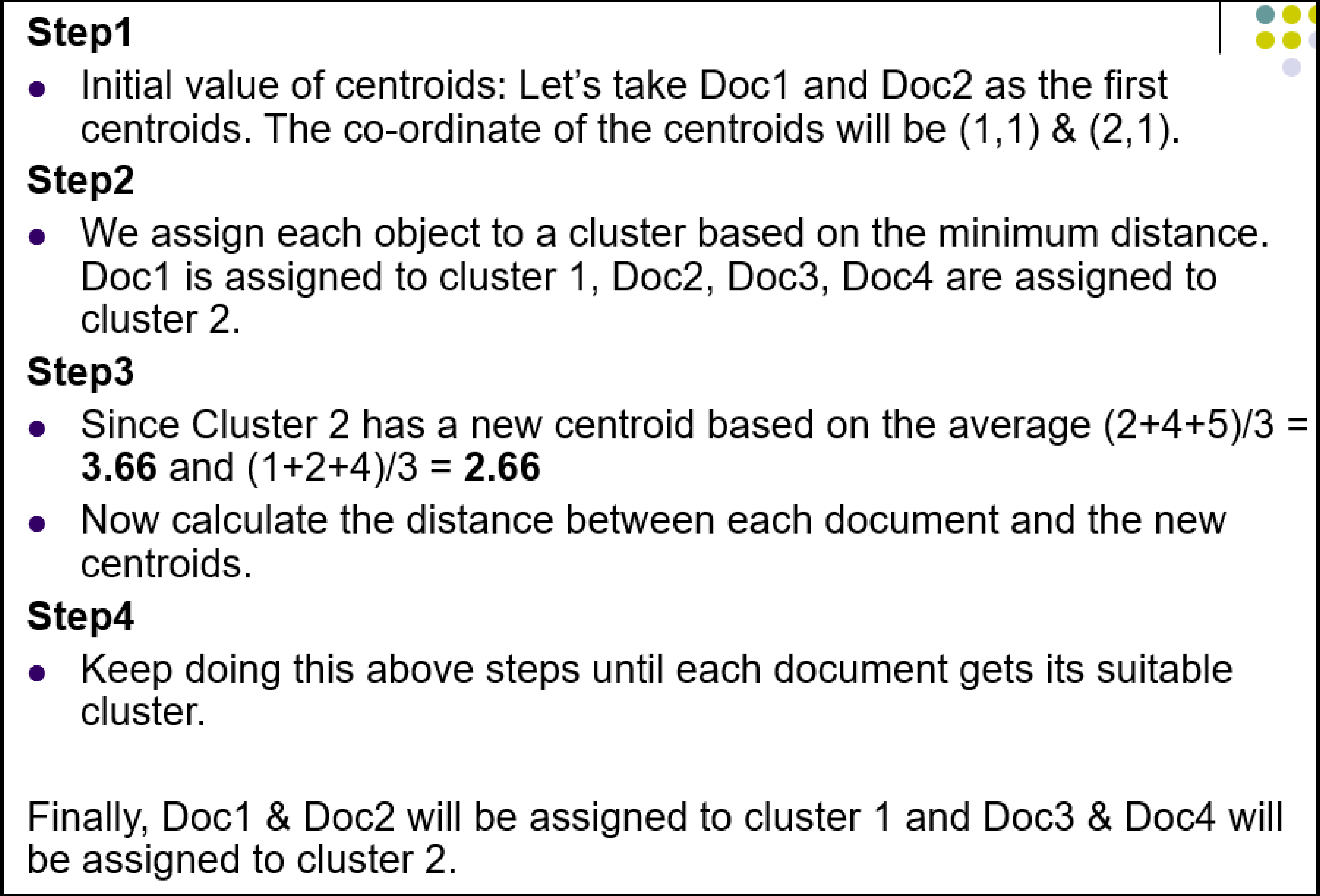
- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to ‘Until relatively few points change clusters’
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

Example:

- We have 4 documents as our training data points, each data point has 2 attributes. Each attributes represents coordinate of the object. Let's attributes are: Words & Counts. We have to determine which document belong to cluster 1 and which document belong to the other cluster.

Object	Attribute (X)	Attribute (Y)
Doc1	1	1
Doc2	2	1
Doc3	4	3
Doc4	5	4

Solution:



Step1

- Initial value of centroids: Let’s take Doc1 and Doc2 as the first centroids. The co-ordinate of the centroids will be (1,1) & (2,1).

Step2

- We assign each object to a cluster based on the minimum distance. Doc1 is assigned to cluster 1, Doc2, Doc3, Doc4 are assigned to cluster 2.

Step3

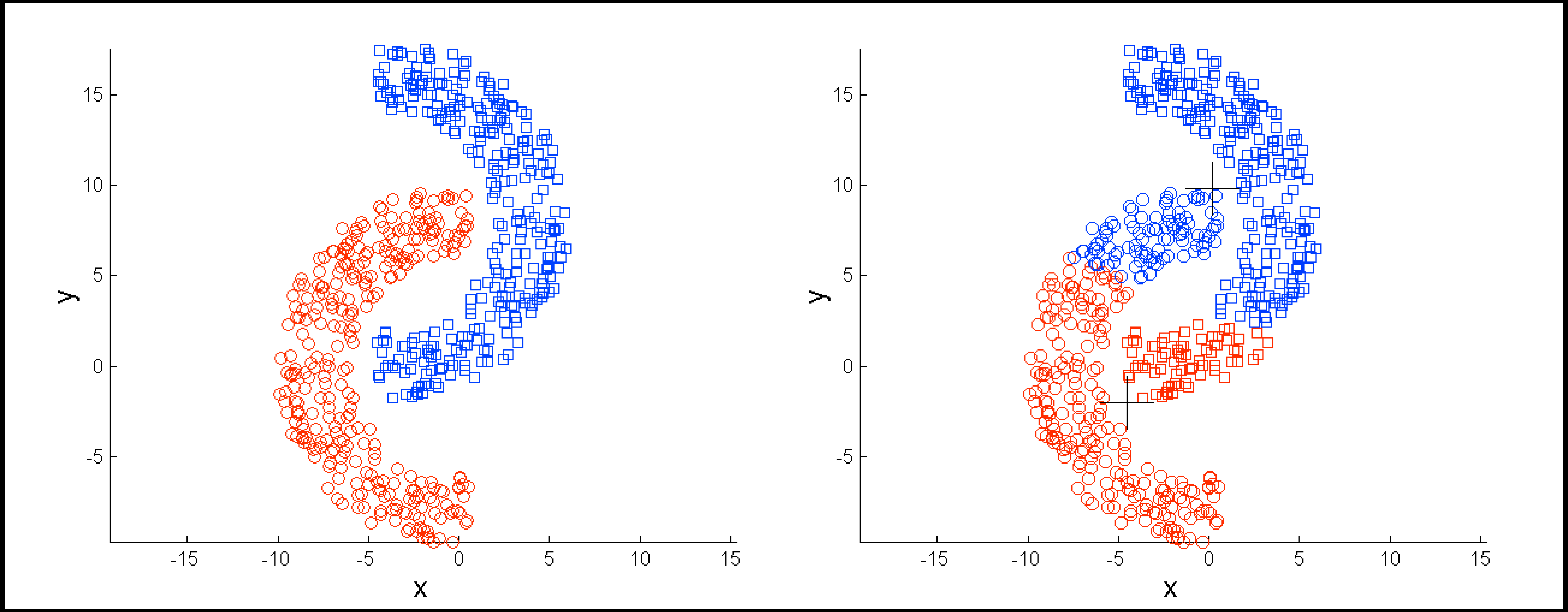
- Since Cluster 2 has a new centroid based on the average (2+4+5)/3 = **3.66** and (1+2+4)/3 = **2.66**
- Now calculate the distance between each document and the new centroids.

Step4

- Keep doing this above steps until each document gets its suitable cluster.

Finally, Doc1 & Doc2 will be assigned to cluster 1 and Doc3 & Doc4 will be assigned to cluster 2.

Limitations of K-means: Non-globular Shapes



Original Points

K-means (2 Clusters)

K-Medoid Clustering

- K-Medoid (Also called as Partitioning around Medoid) and can be defined as the point in the cluster whose dissimilarities with all the other point in the cluster is minimum.

Steps of K-Medoid

- Select 2 Medoid Randomly
- Calculate the distance b/w data points & both medoid
- Calculate the total cost for the cluster using these medoid.
- Again choose some other medoids & repeat Step 1 & 2. if you don’t get a better total cost you have to stop.

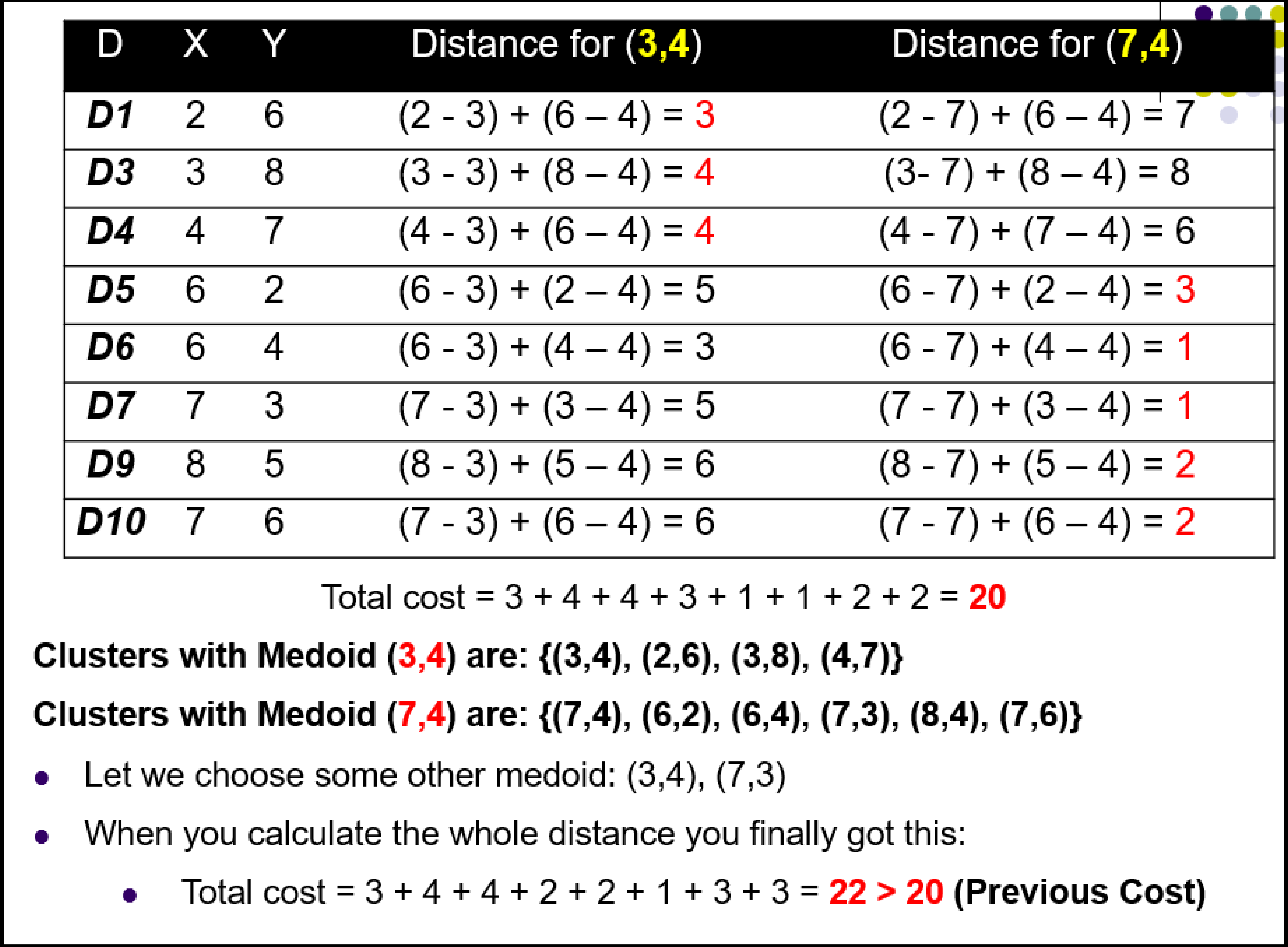
Example:

- ❖ We have these data sets **{(2,6), (3,4), (3,8), (4,7), (7,4), (6,2), (6,4), (7,3), (8,4), (7,6)}**
  - ✓ Let we divide these data sets into 2 clusters.

  First, Choose randomly two medoids from the data set. Let’s take **(3,4)** and **(7,4)** as medoids.

  In ***K-Means Algorithms***, we’ve used **Euclidean Distance** so as to calculate the distance between data points, So, in ***K-Medoid Algorithm*** we’ll use **Manhattan Distance** to calculate the distance between the data points.

Solution:



D	X	Y	Distance for (3,4)	Distance for (7,4)
D1	2	6	(2 - 3) + (6 – 4) = 3	(2 - 7) + (6 – 4) = 7
D3	3	8	(3 - 3) + (8 – 4) = 4	(3- 7) + (8 – 4) = 8
D4	4	7	(4 - 3) + (6 – 4) = 4	(4 - 7) + (7 – 4) = 6
D5	6	2	(6 - 3) + (2 – 4) = 5	(6 - 7) + (2 – 4) = 3
D6	6	4	(6 - 3) + (4 – 4) = 3	(6 - 7) + (4 – 4) = 1
D7	7	3	(7 - 3) + (3 – 4) = 5	(7 - 7) + (3 – 4) = 1
D9	8	5	(8 - 3) + (5 – 4) = 6	(8 - 7) + (5 – 4) = 2
D10	7	6	(7 - 3) + (6 – 4) = 6	(7 - 7) + (6 – 4) = 2

Total cost = 3 + 4 + 4 + 3 + 1 + 1 + 2 + 2 = **20**

Clusters with Medoid (3,4) are: {(3,4), (2,6), (3,8), (4,7)}

Clusters with Medoid (7,4) are: {(7,4), (6,2), (6,4), (7,3), (8,4), (7,6)}

- Let we choose some other medoid: (3,4), (7,3)
- When you calculate the whole distance you finally got this:
  - Total cost = 3 + 4 + 4 + 2 + 2 + 1 + 3 + 3 = **22 > 20 (Previous Cost)**

Spectral Clustering

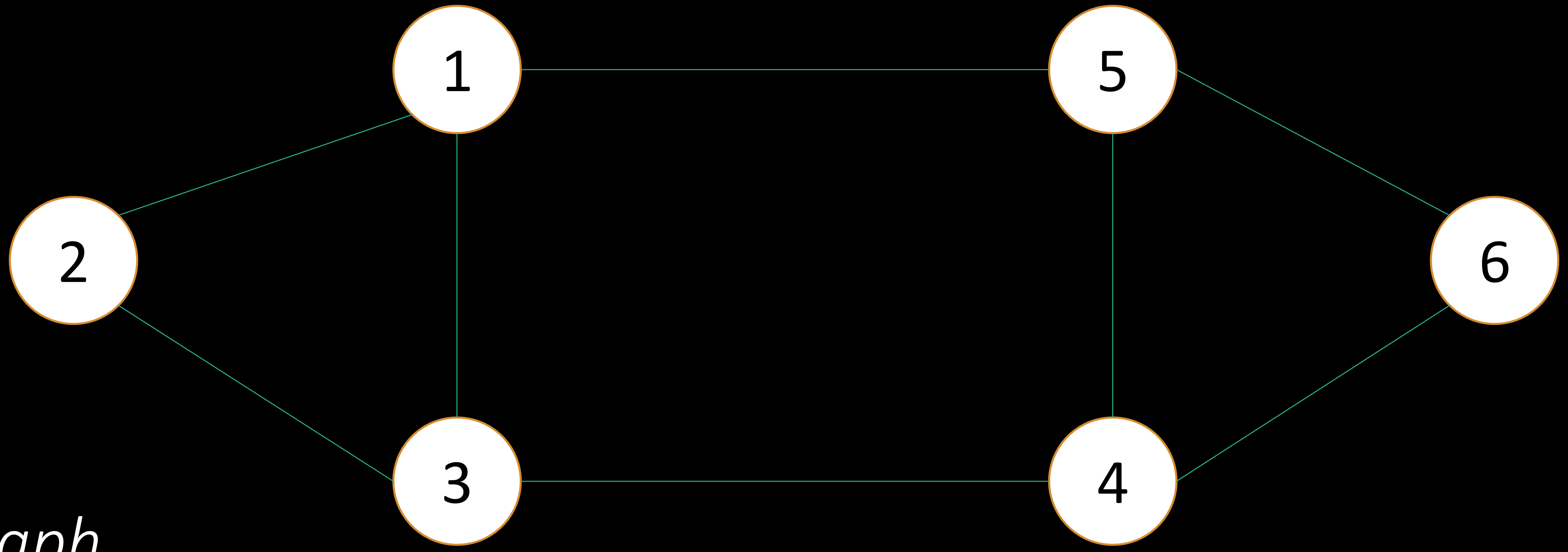
Spectral Clustering is about finding the clusters in a graph. e.g. Finding the mostly connected sub graphs by identifying the clusters.

Steps of Spectral Clustering

- Pre-Processing: Matrix Representation of a graph.
- De-composition: Compare Eigen value & Eigen vector.
- Grouping: Partitioning or Clustering.

Example:

- Lets say we've an undirected graph G (V,E)



Step 1: Find Matrix Representation for the graph

Adjacency Matrix:

	1	2	3	4	5	6
1	0	1	1	0	1	0
2	1	0	1	0	0	0
3	1	1	0	1	0	0
4	0	0	1	0	1	1
5	1	0	0	1	0	1
6	0	0	0	1	1	0

Solution:

Degree Matrix:

	1	2	3	4	5	6
1	3	0	0	0	0	0
2	0	2	0	0	0	0
3	0	0	3	0	0	0
4	0	0	0	3	0	0
5	0	0	0	0	3	0
6	0	0	0	0	0	2

0 → Pair of nodes that are not connected

-1 → Pair of nodes that are connected

Laplacian Matrix = Degree Matrix – Adjacency Matrix

	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	-1	0	0	-1	3	-1
6	0	0	0	-1	-1	2

Cont..

Step 2: Compute Eigen Value and Eigen Vector

Calculating Eigen needs more time, so try it by your self as homework.

Finally, when you calculated Eigen Value & Vector, you'll get this below table:

1	**-3**
2	**-6**
3	**-3**
4	**3**
5	**3**
6	**6**

Step 3: Group the into two cluster A & B

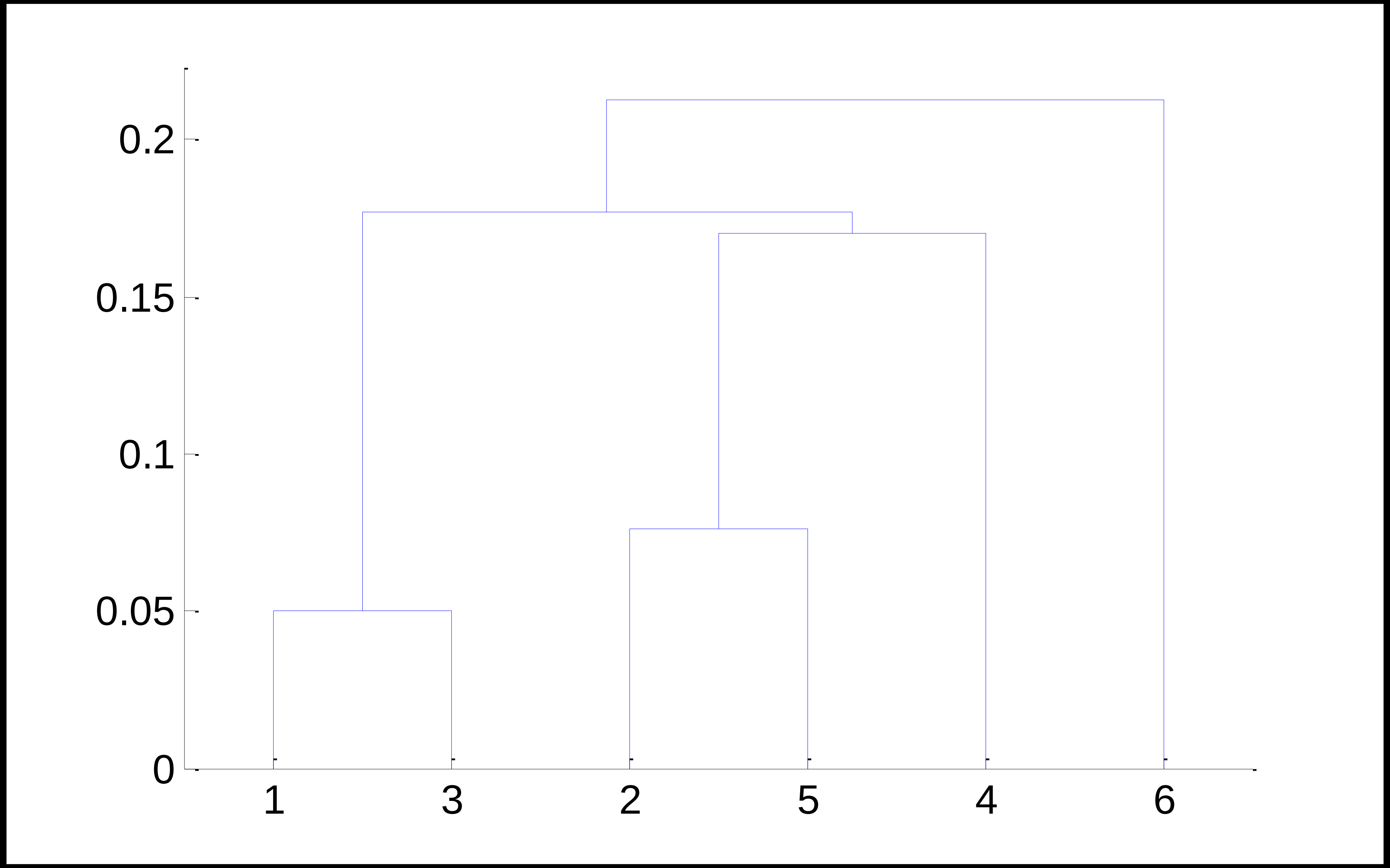
So, if you got positive number from the Eigen Value, let's put on cluster A

if you got negative number from the Eigen Value, let's put on cluster B

The final two clusters will be: A = {1,2,3} and B = {4,5,6}

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits



Cont..

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)

Traditional hierarchical algorithms use a similarity or distance matrix

- Merge or split one cluster at a time

Agglomerative Algorithm

More popular hierarchical clustering technique

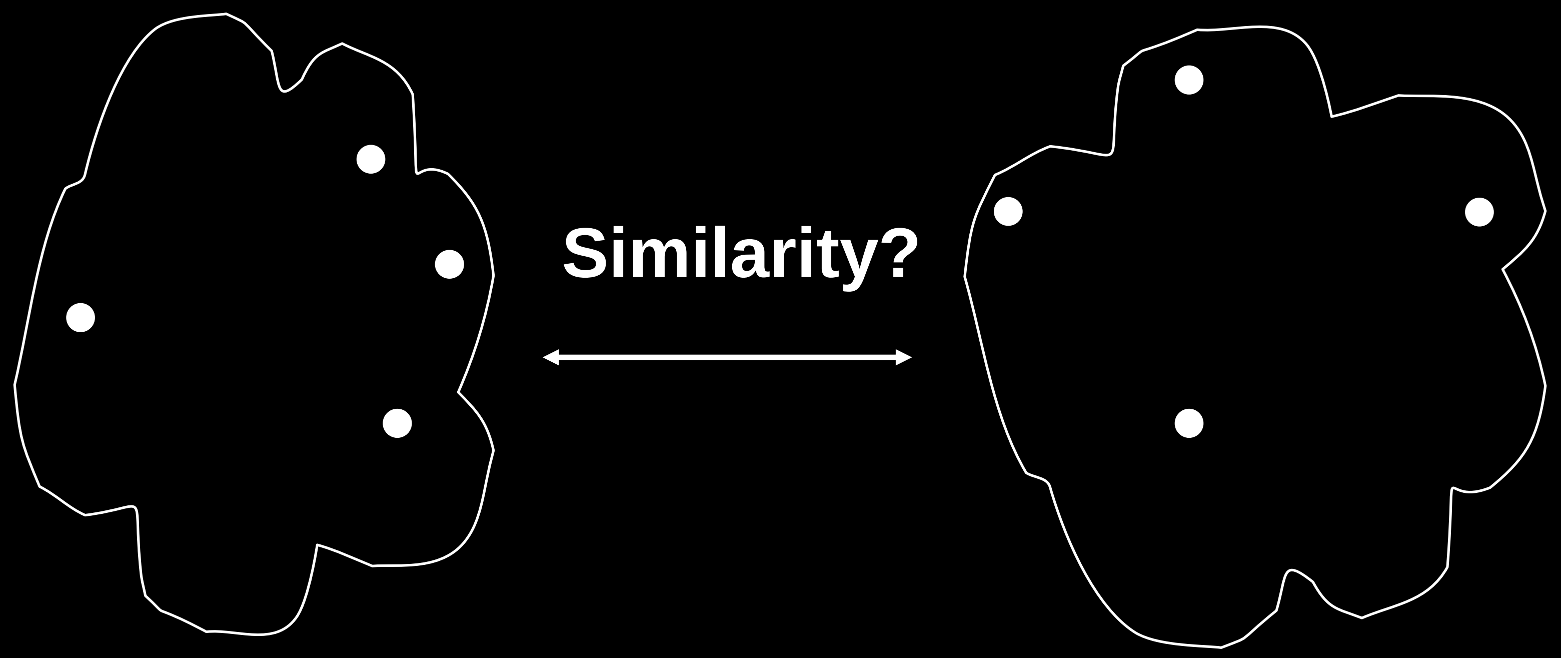
Basic algorithm is straightforward

```
1. Compute the proximity matrix
2. Let each data point be a cluster
3. Repeat
4.         Merge the two closest clusters
5.         Update the proximity matrix
6. Until only a single cluster remains
```

Key operation is the computation of the proximity of two clusters

- Different approaches to defining the distance between clusters distinguish the different algorithms

How to Define Inter-Cluster Similarity

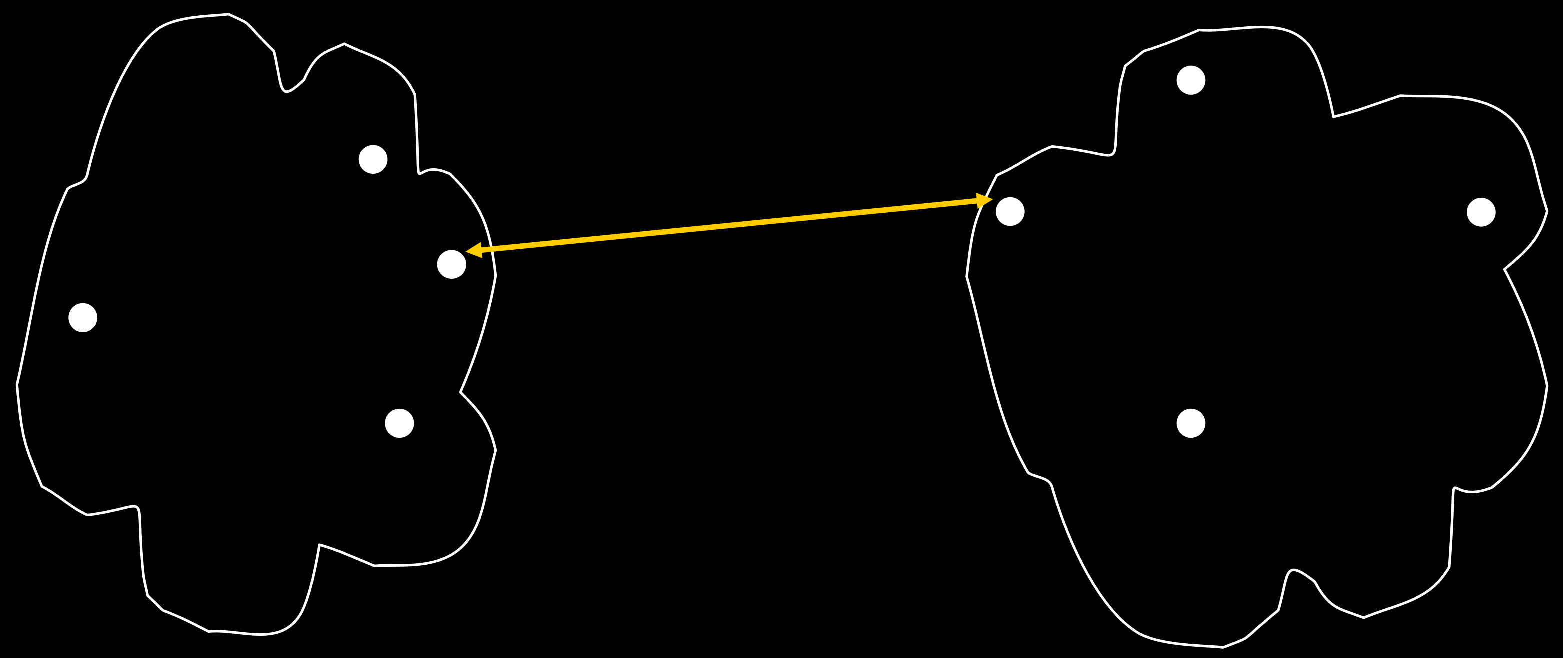


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Cont..

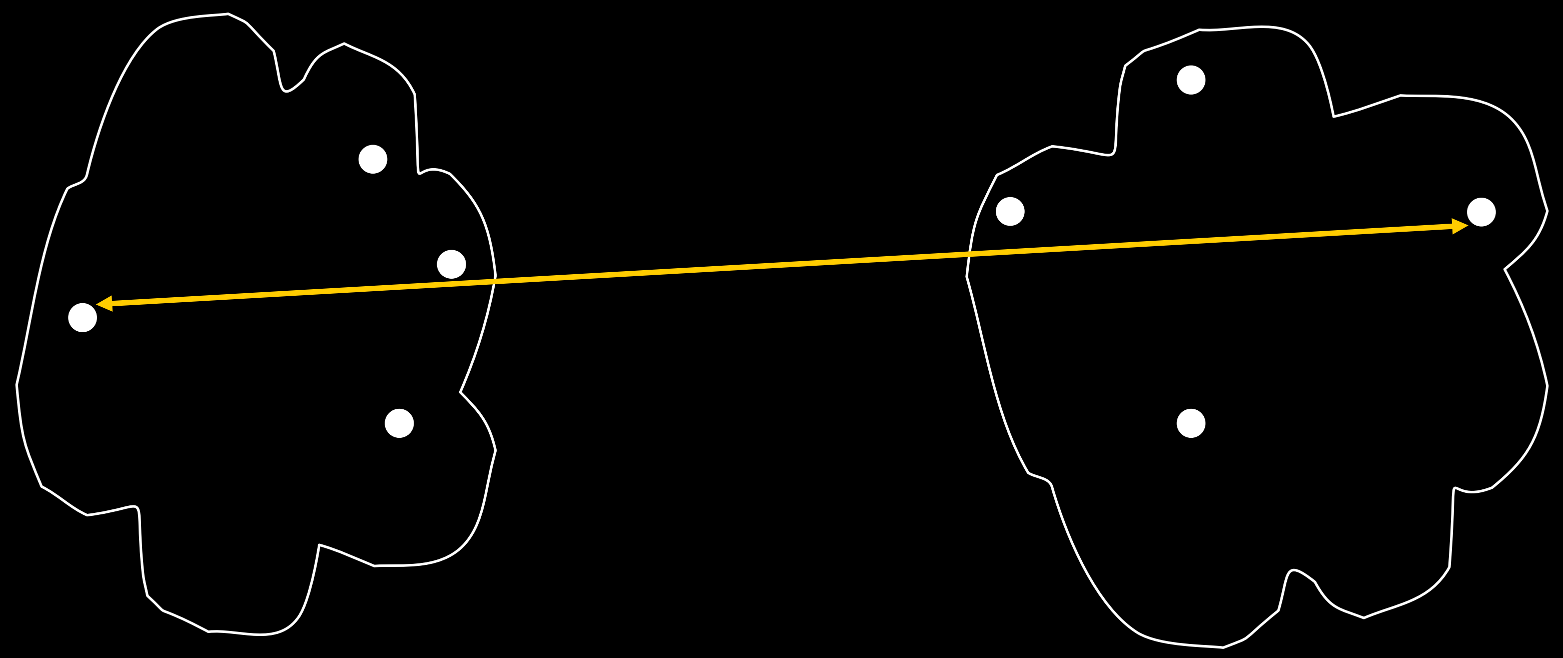


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Proximity Matrix

Cont..



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

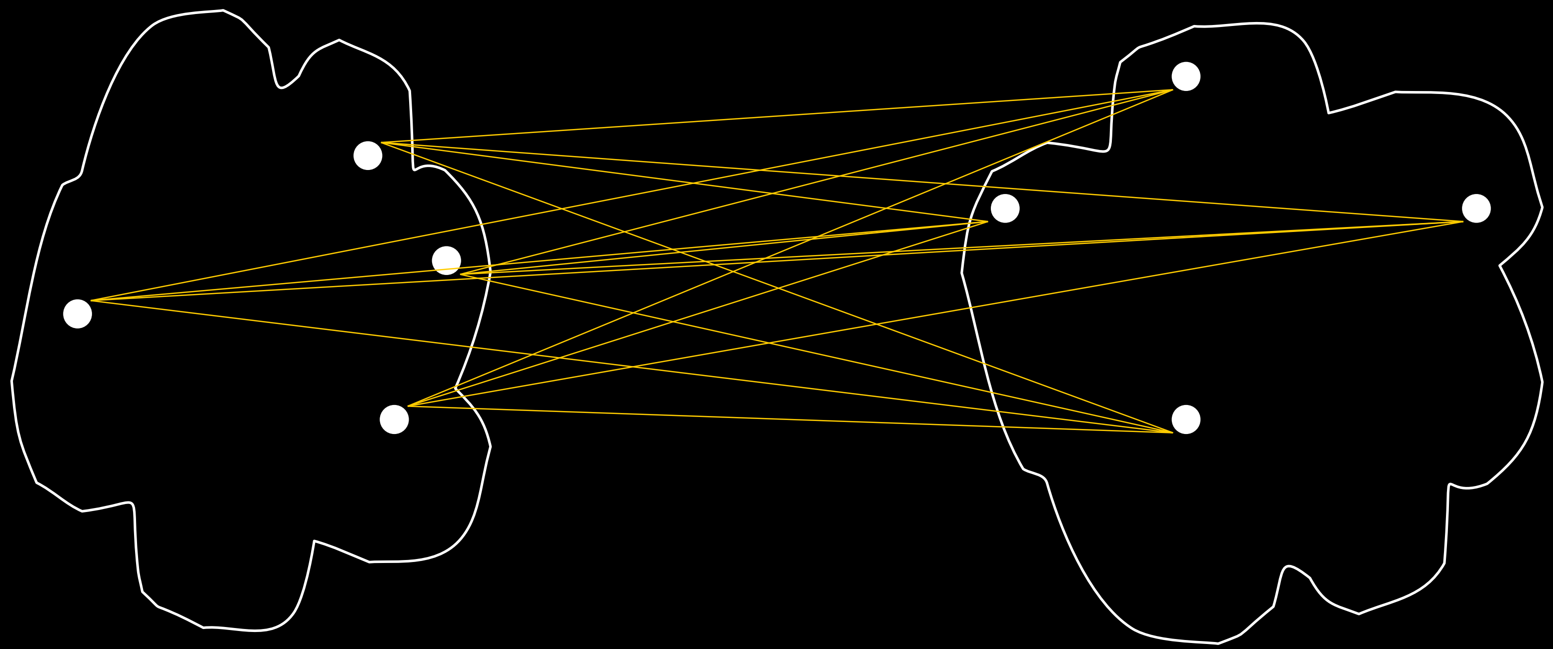
.

.

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Cont..

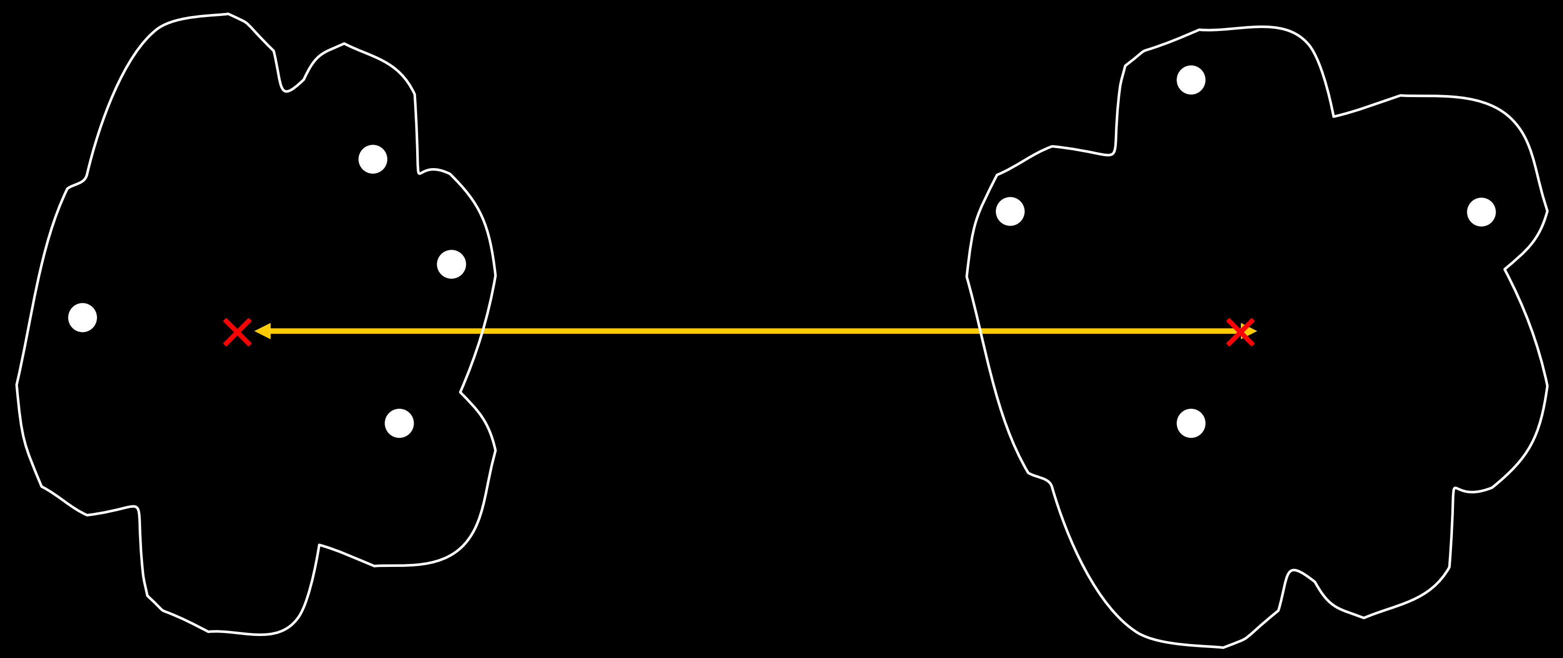


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

Cont..



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is

  - Accuracy, precision, recall

- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?

- But “clusters are in the eye of the beholder”!

Then why do we want to evaluate them?

- To avoid finding patterns in noise
- To compare clustering algorithms
- To compare two sets of clusters
- To compare two clusters

Summary

- **Unsupervised Learning** is a machine learning technique in which the users do not need to supervise the model. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with the unlabeled data.
  - ✓ It allow users to perform more complex processing tasks compared to supervised learning.

Unsupervised learning problems further grouped into clustering problems.

Practical Implementation for Clustering Algorithm!

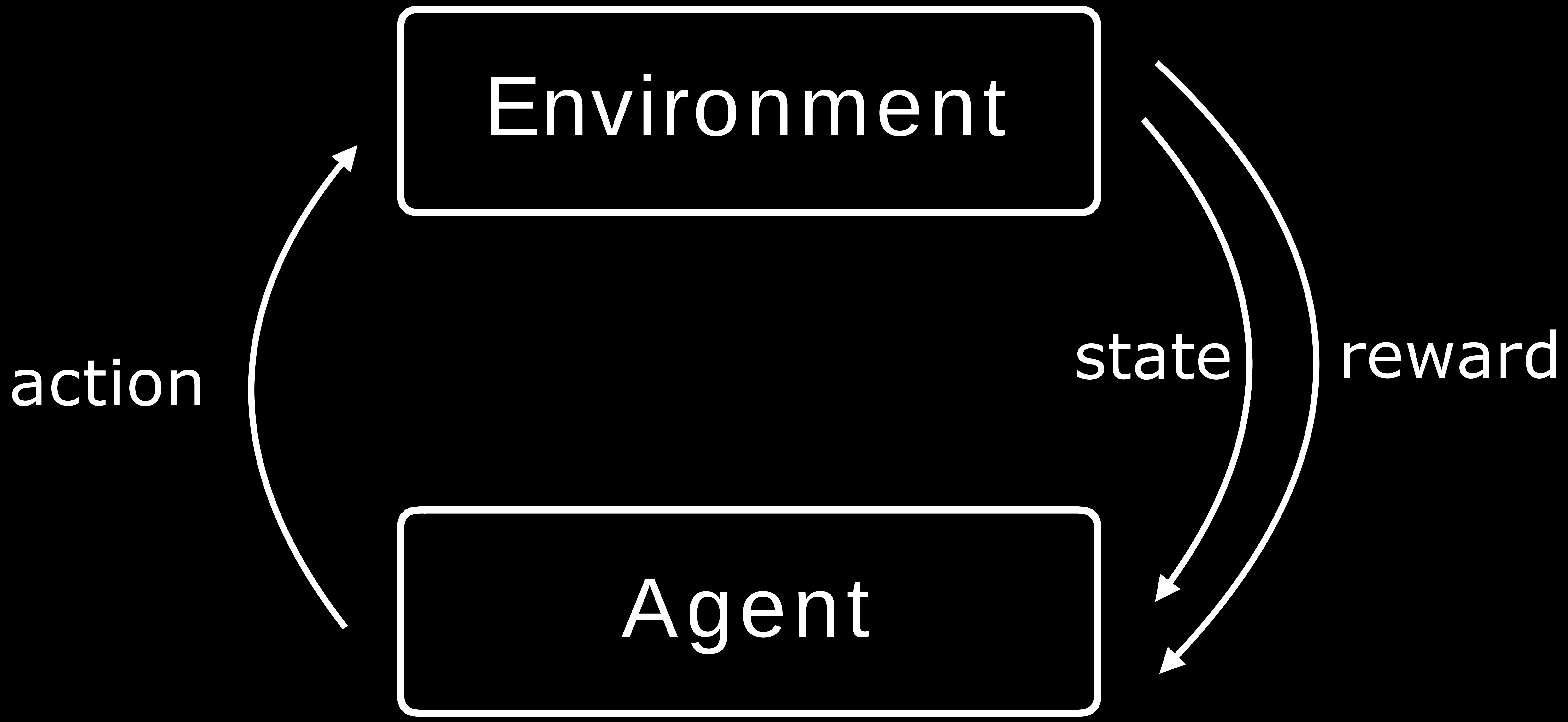
So as to do the practical session of these clustering Algorithms make sure that your computer is installed these below software and modules.

- **Python**
- **Jupyter Notebook**
- **Numpy Module**
- **Matplotlib Module**
- **Pandas Module**

Reinforcement Learning

reinforcement learning

given a set of rewards or punishments, learn what actions to take in the future



Environment

action

state

reward

Agent

Thank You!