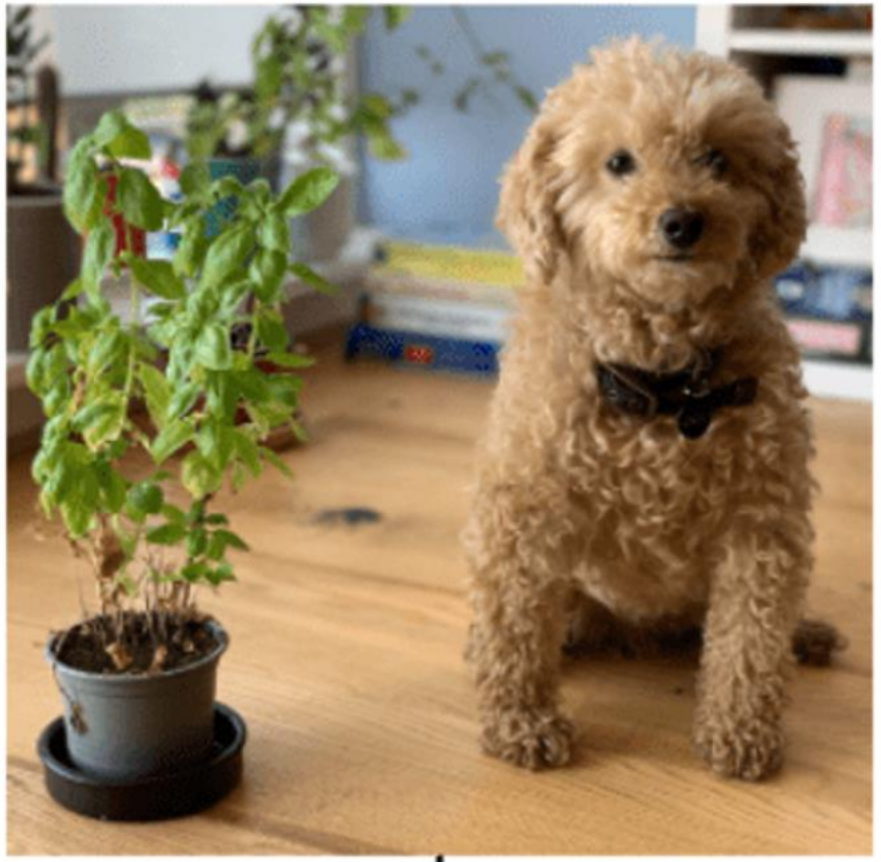
classification

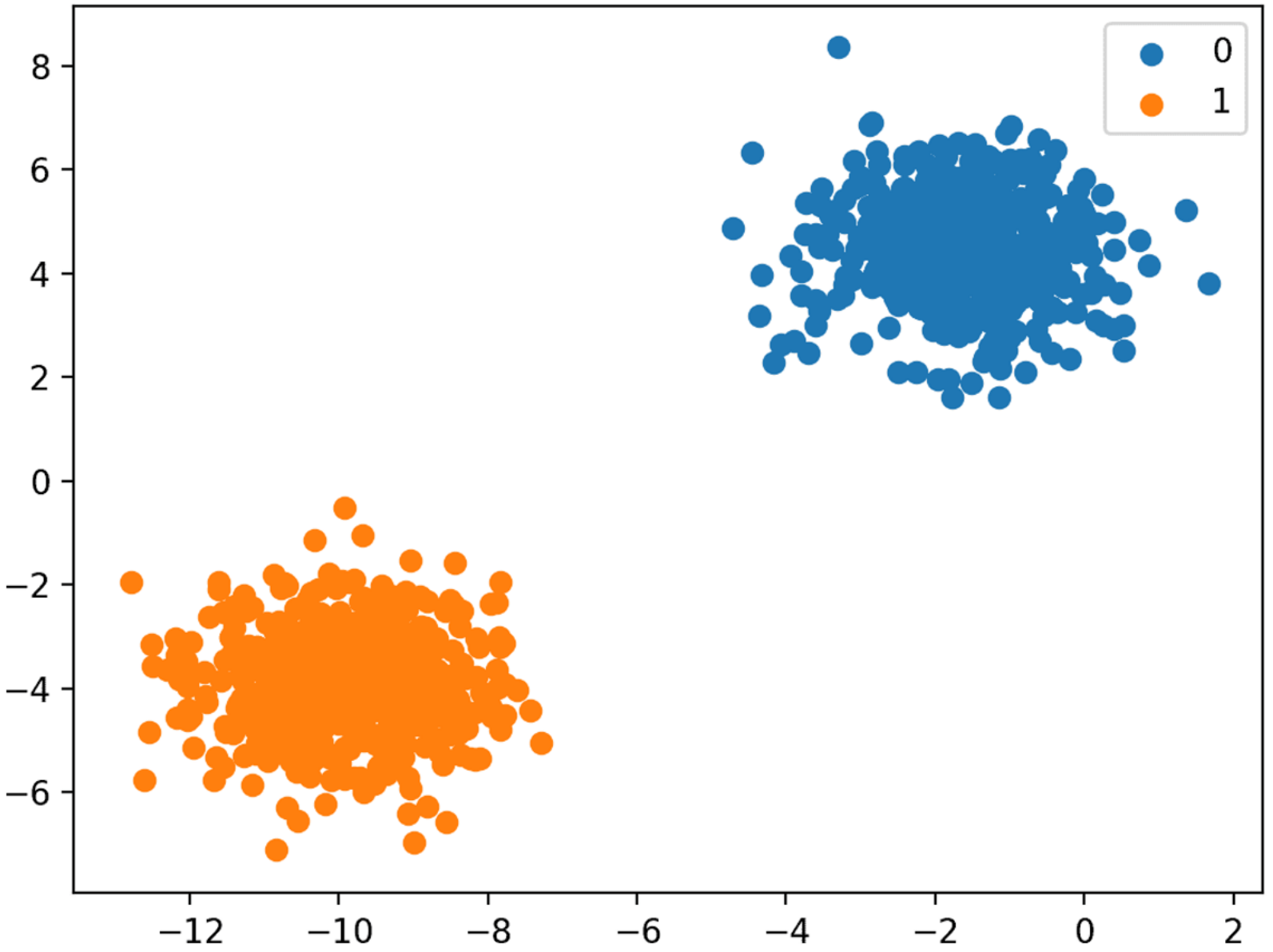
supervised learning task of learning a function mapping an input point to a discrete category

Binary Classification

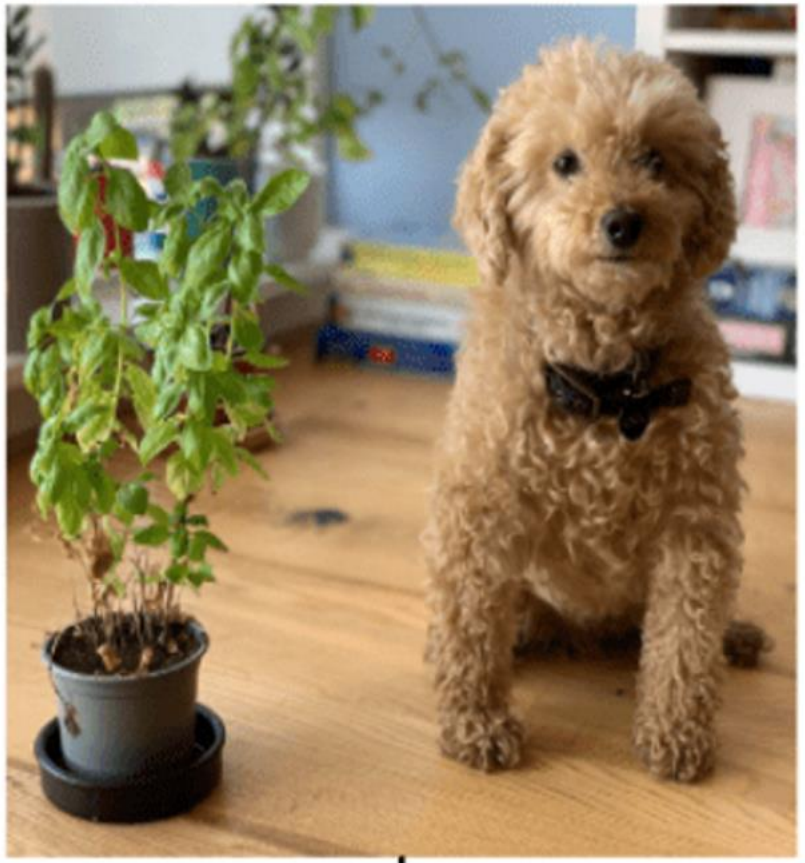


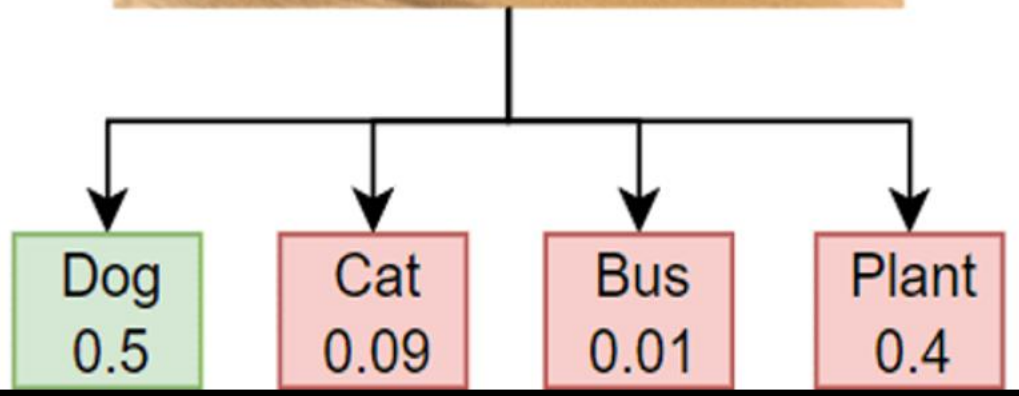
Dog
0.9

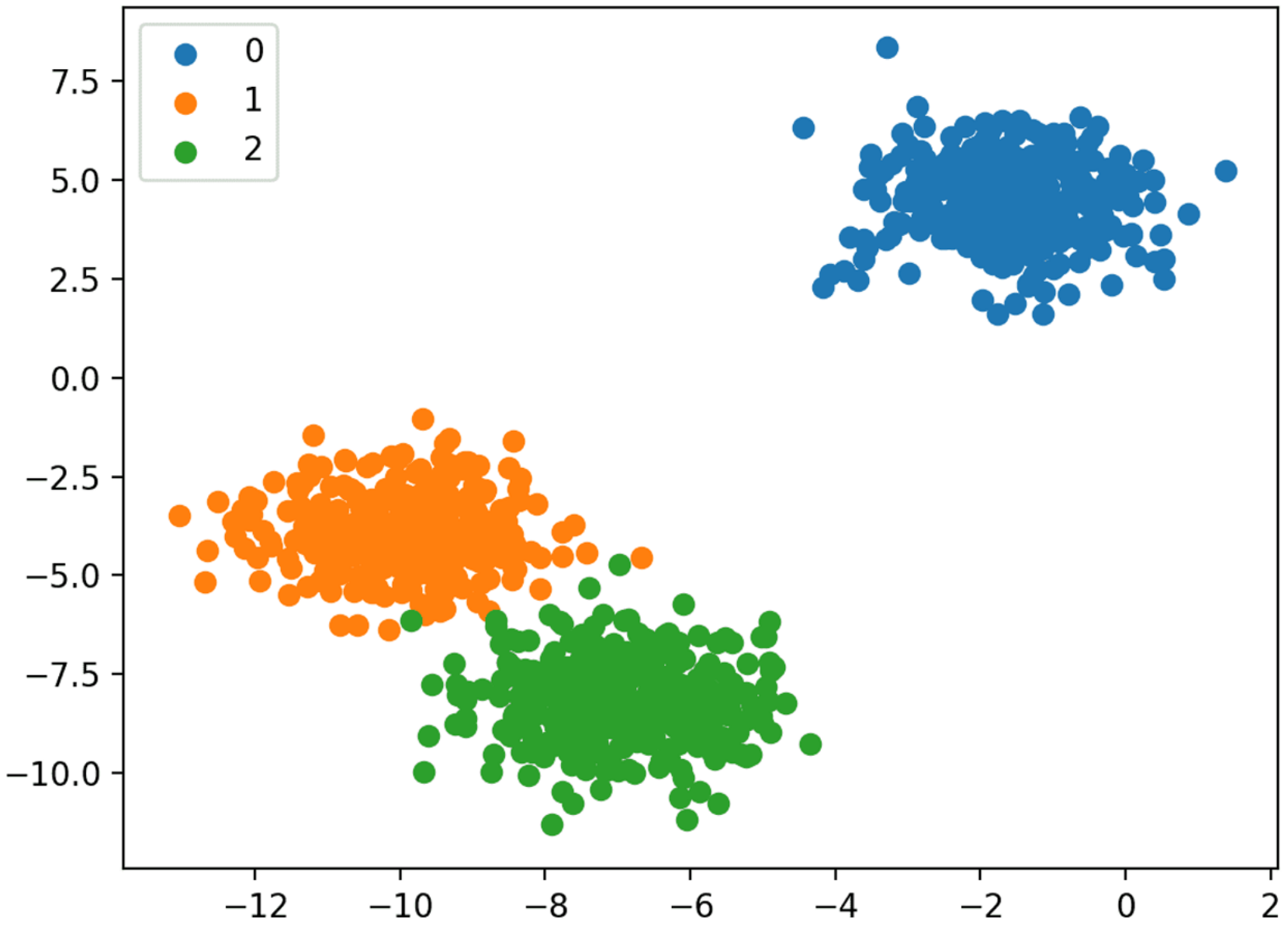
Not Dog
0.1


0
1
8
6
4
2
0
−2
−4
−6
−12
−10
−8
−6
−4
−2
0
2

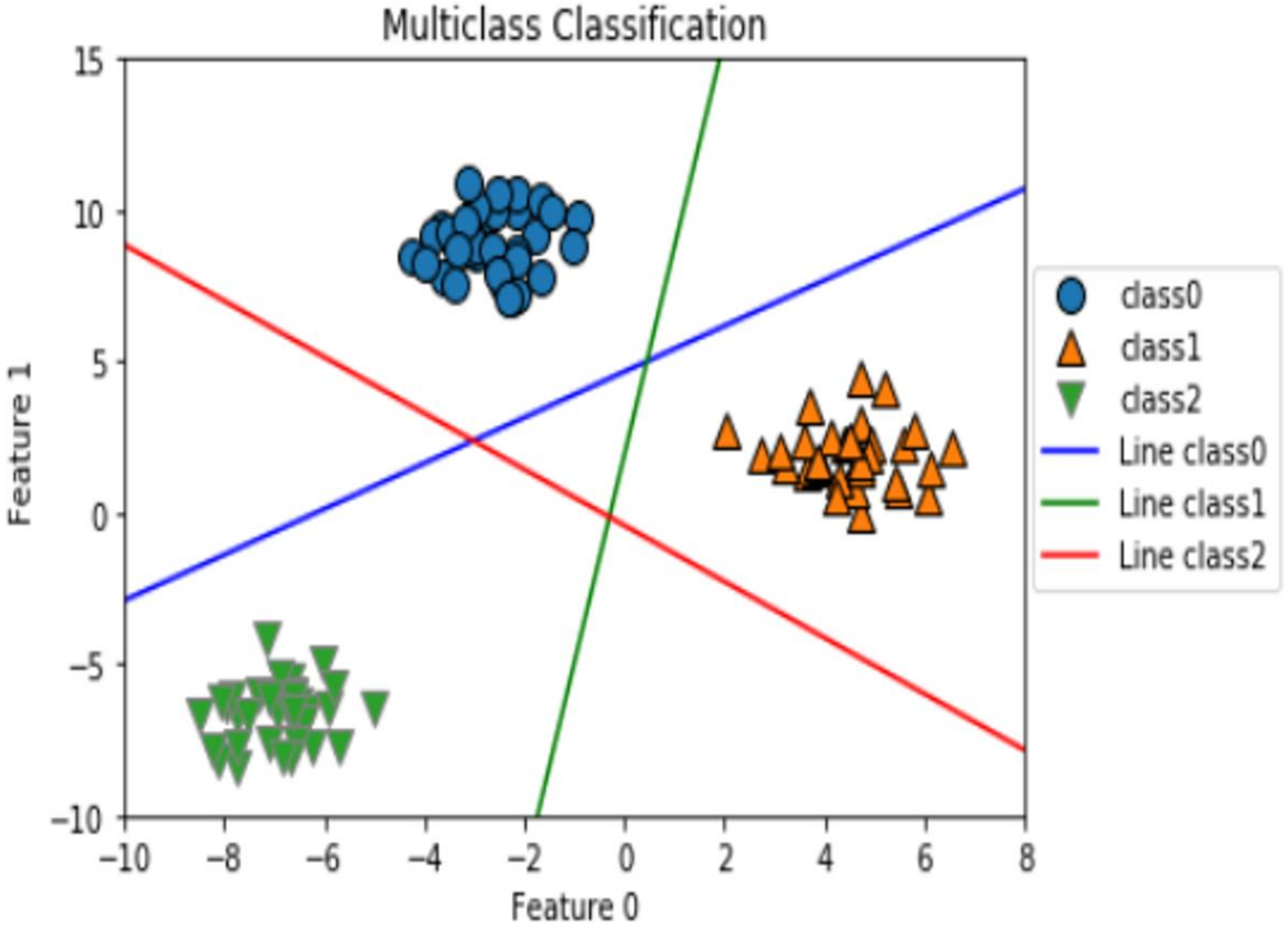
Multiclass Classification



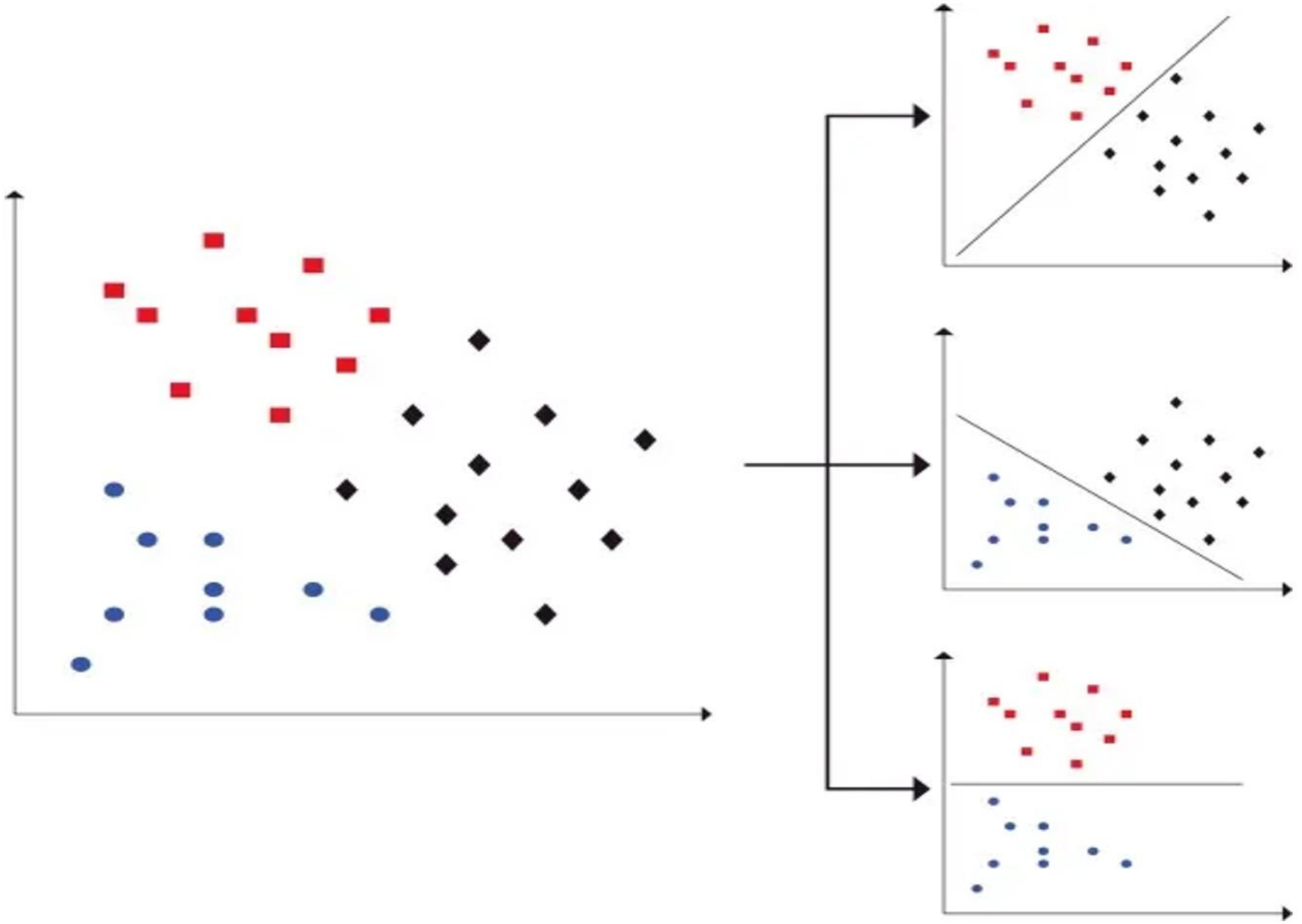



0
1
2
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−12
−10
−8
−6
−4
−2
0
2

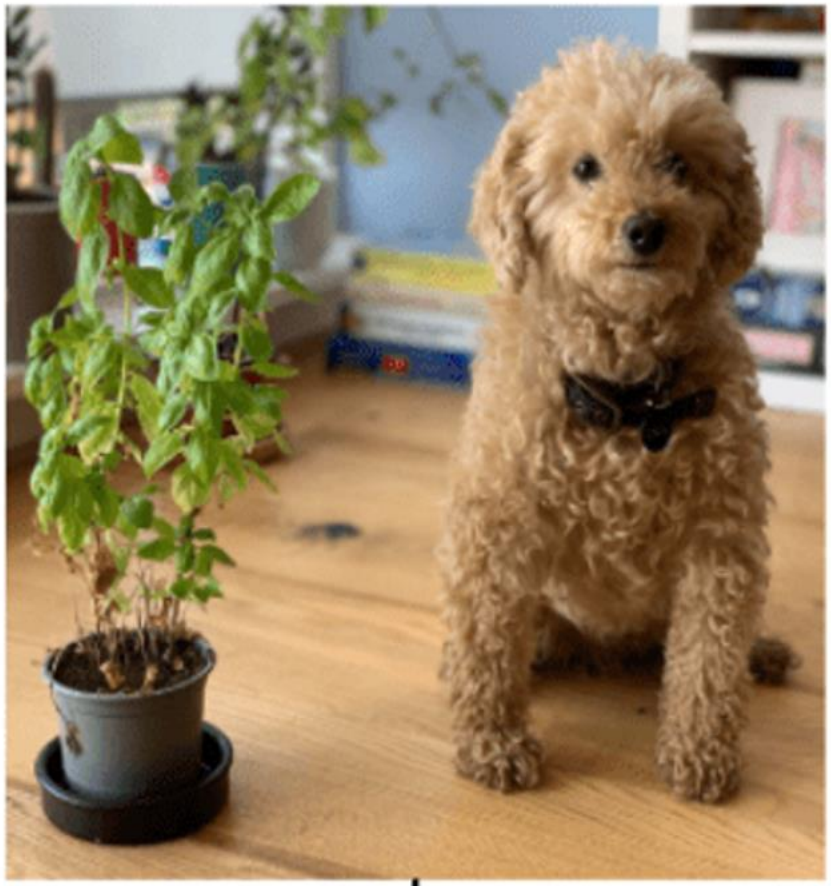
One-vs-All(Rest): Fit one binary classification model for each class vs. all other classes.



One-vs-One: Fit one binary classification model for each pair of classes.



Multilabel Classification



Dog	Cat	Bus	Plant
0.8	0.2	0.04	0.7


100
FOR MOTION PICTURE USE ONLY
PR 10008679 P
Z1
ONE HUNDRED
FOR MOTION PICTURE USE ONLY
THIS NOTE IS NOT LEGAL IT IS TO BE USED FOR MOTION PICTURES
July 4th
PR 10008679 P
100
100
FEDERAL RESERVE NOTE
JB 51393142 A
B2
ONE HUNDRED DOLLARS
THE UNITED STATES OF AMERICA
THIS NOTE IS LEGAL TENDER FOR ALL DEBTS, PUBLIC AND PRIVATE
JULY 4, 1776
JB 51393142 A
100
100
FRANKLIN

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

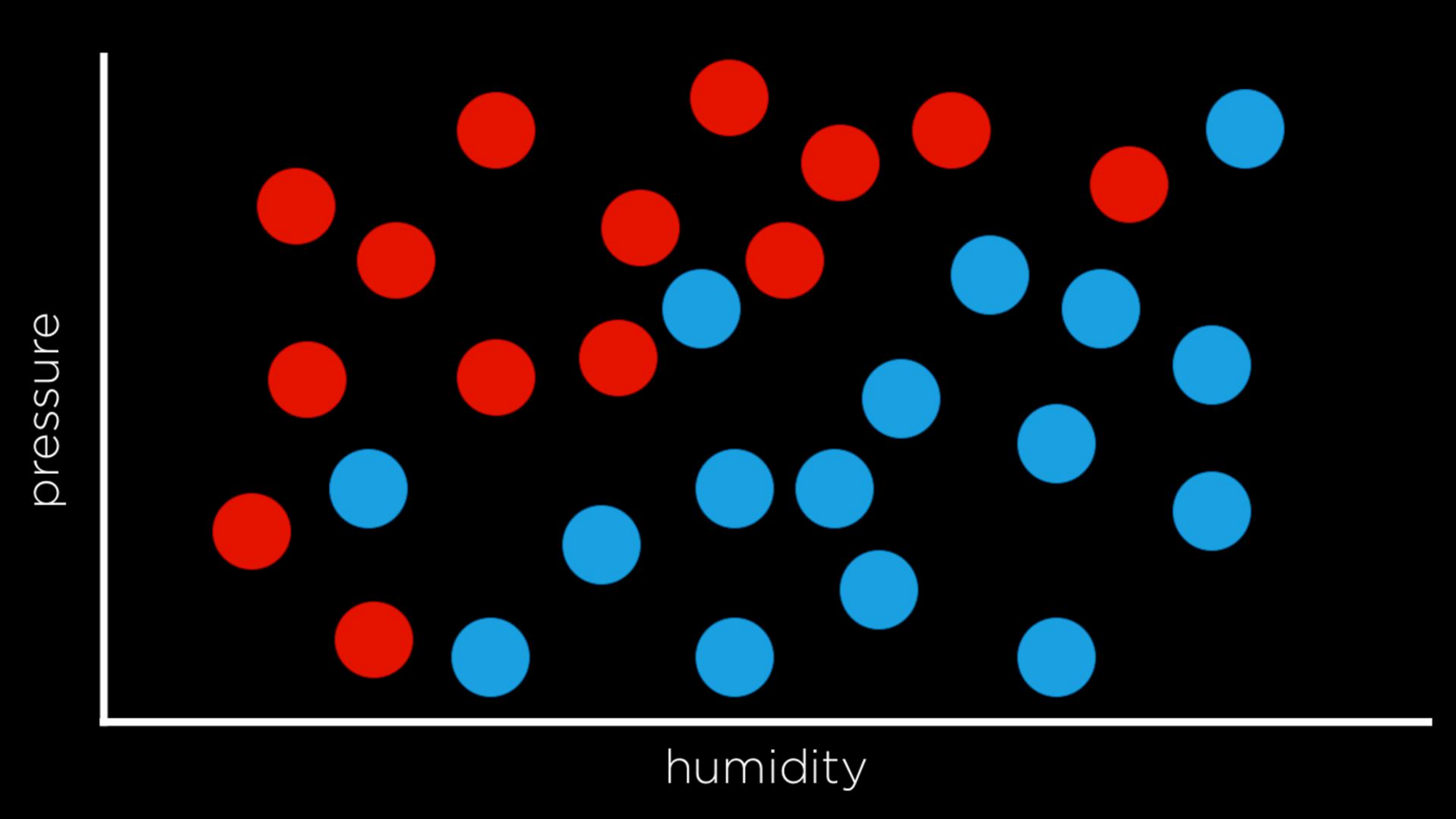
$f(humidity, pressure)$

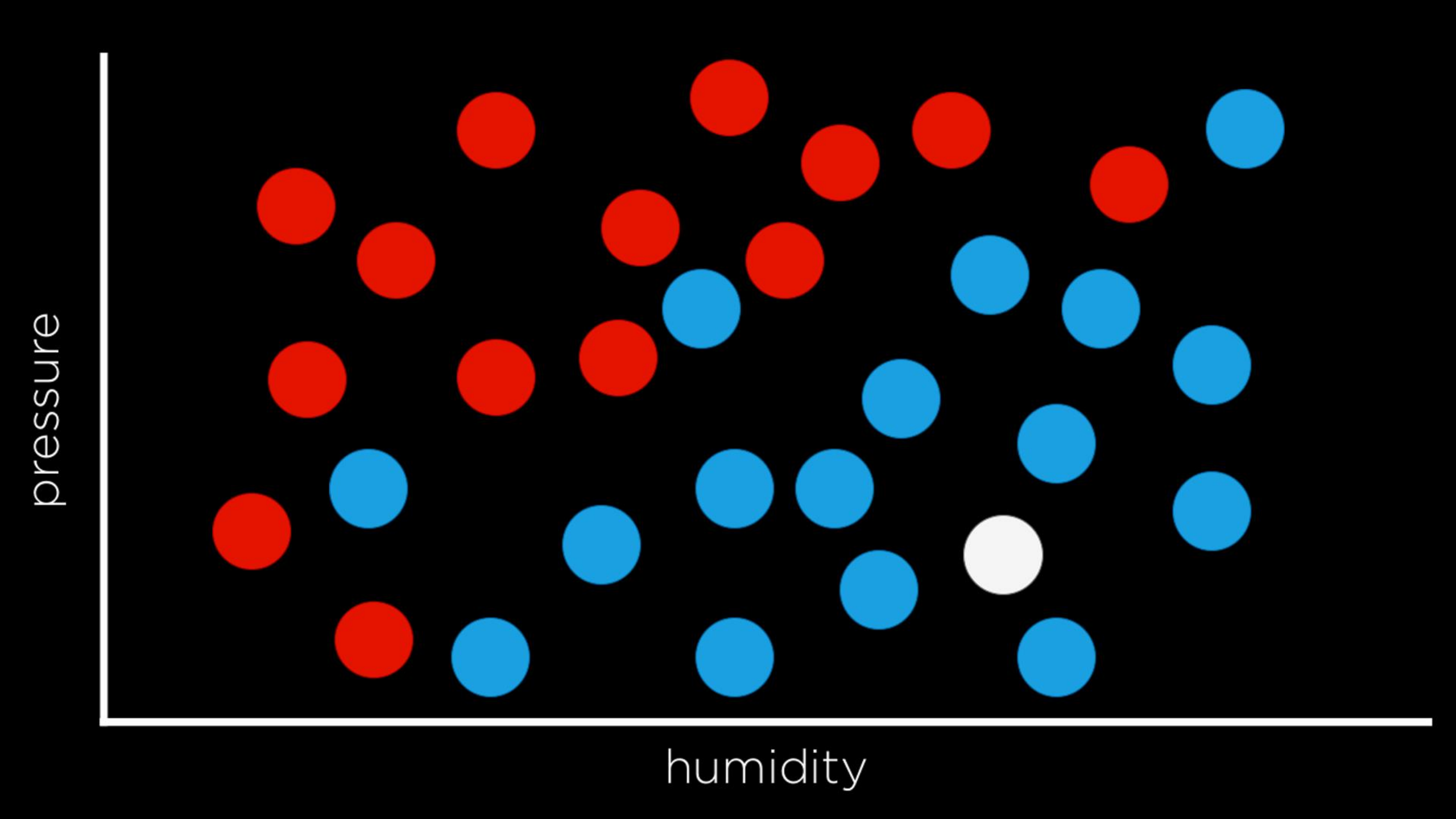
$f(93, 999.7) = \text{Rain}$

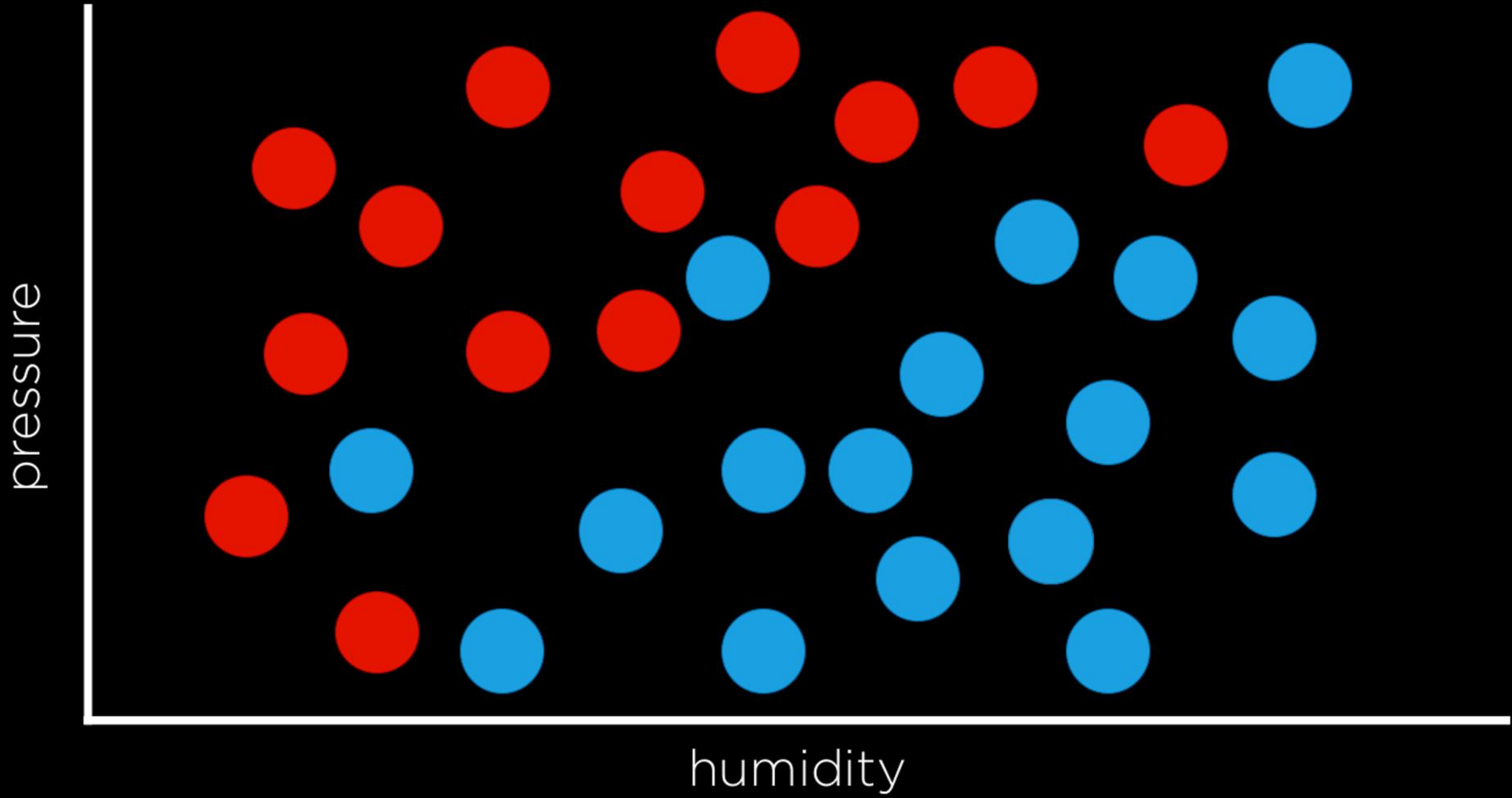
$f(49, 1015.5) = \text{No Rain}$

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

$h(humidity, pressure)$

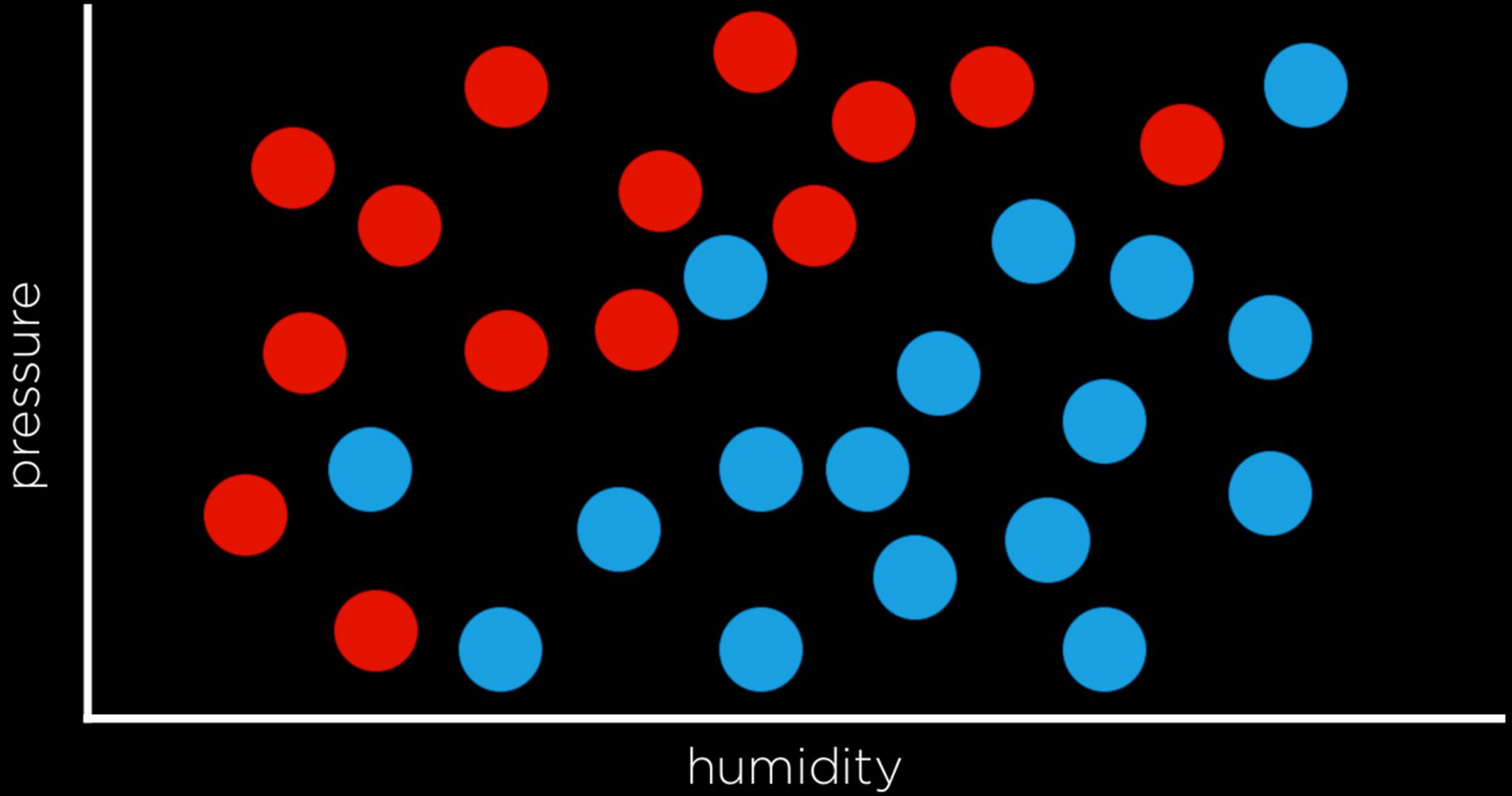

pressure
humidity

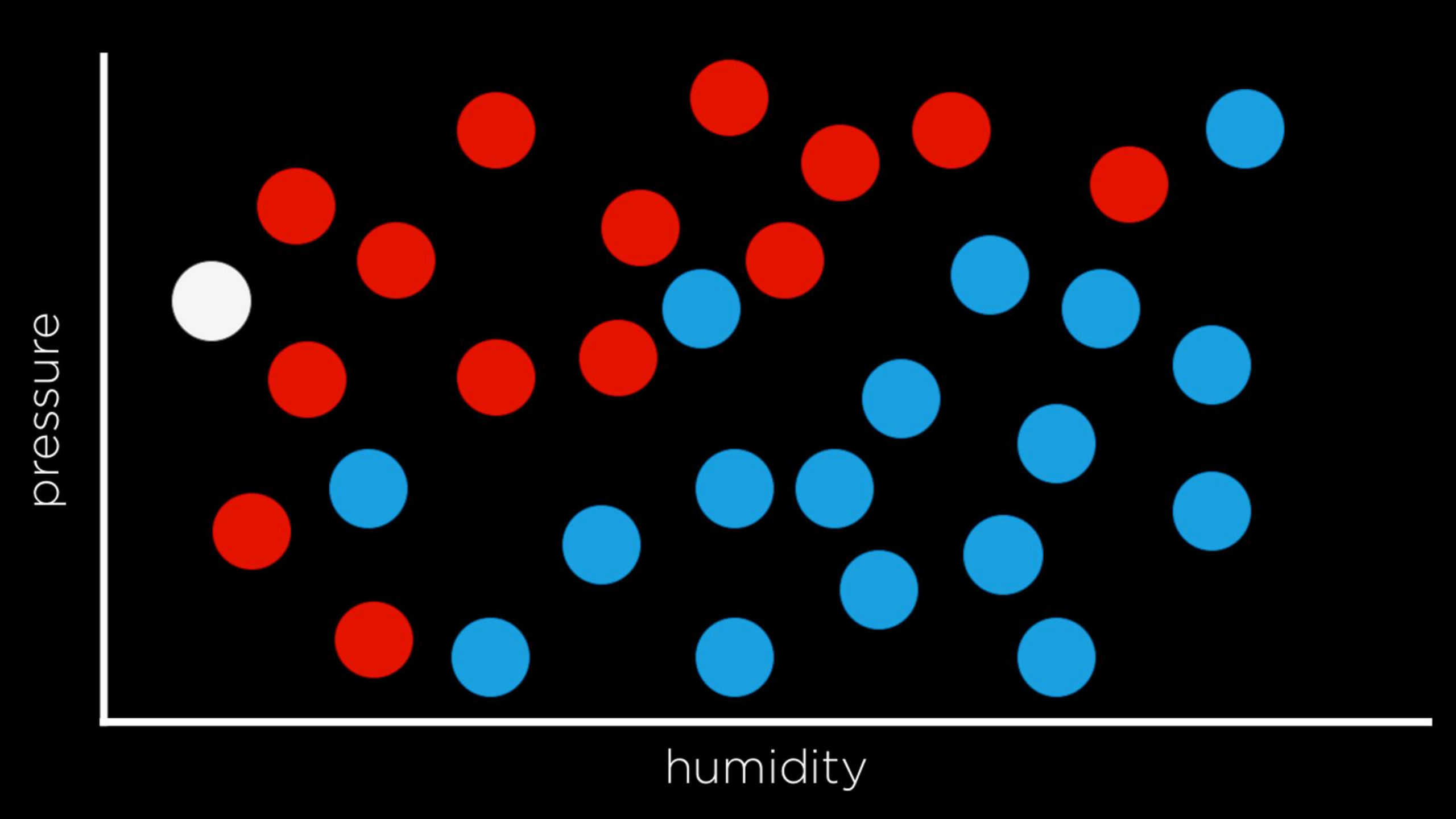

pressure
humidity

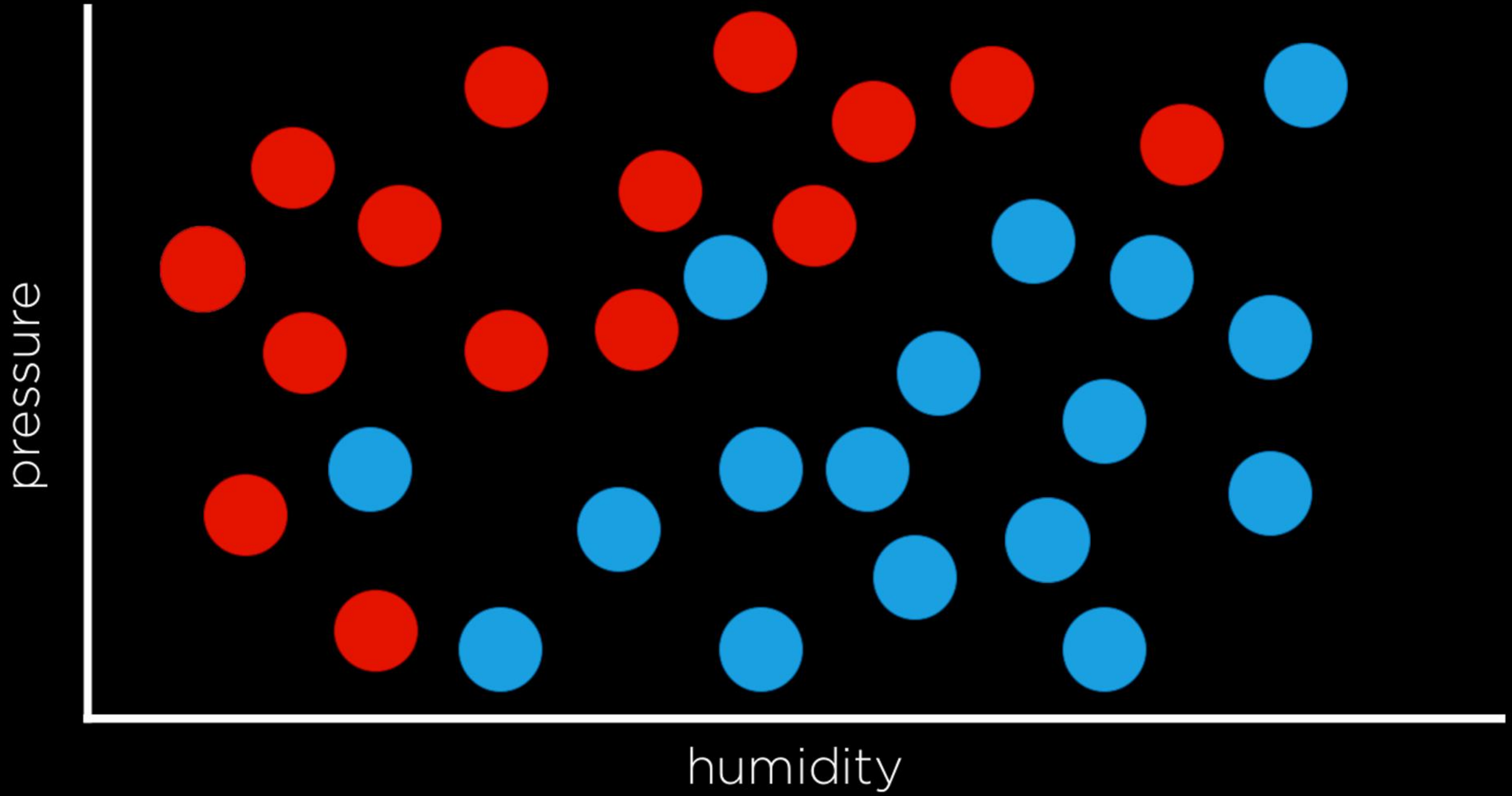

pressure
humidity

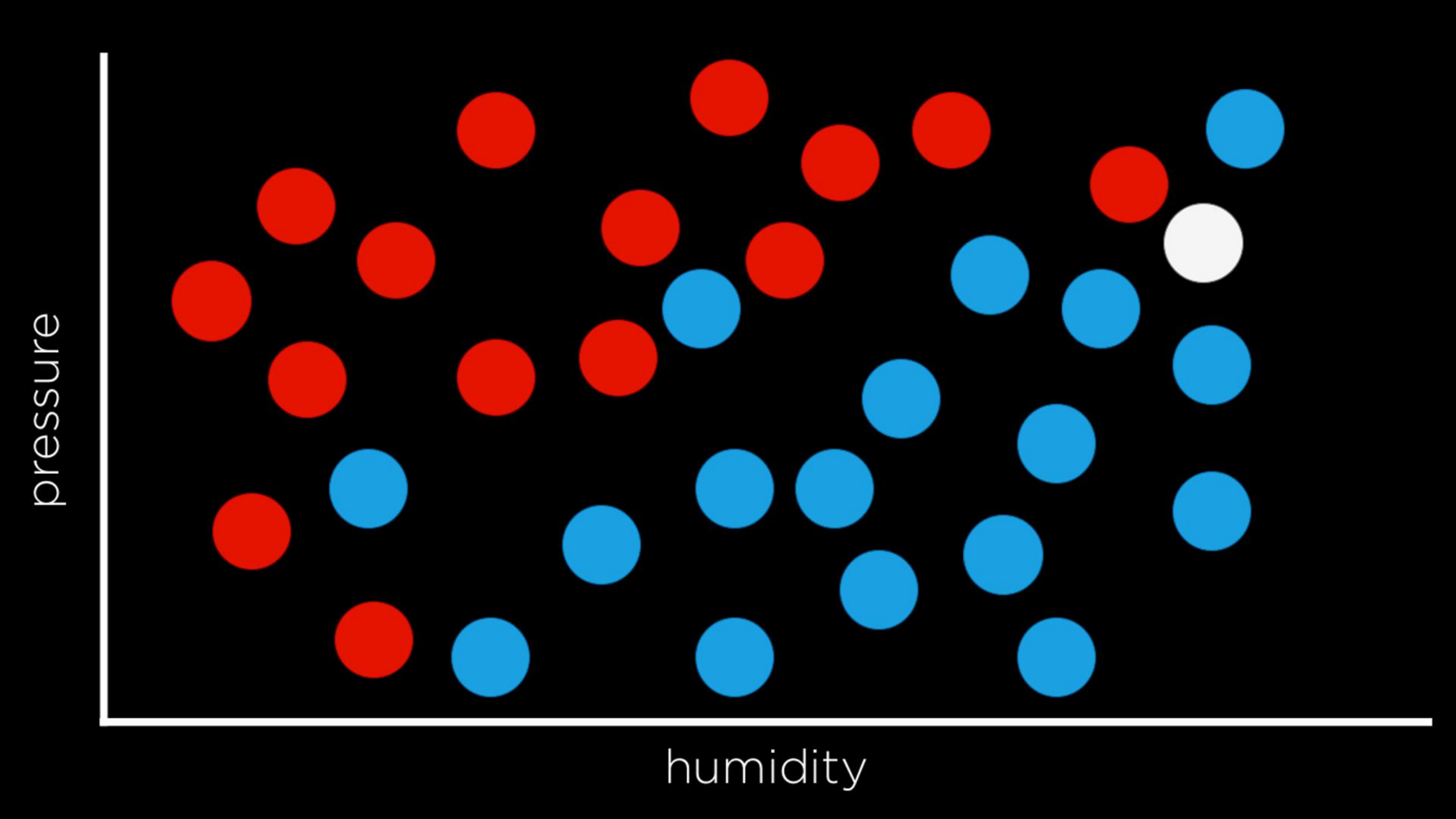
nearest-neighbor classification

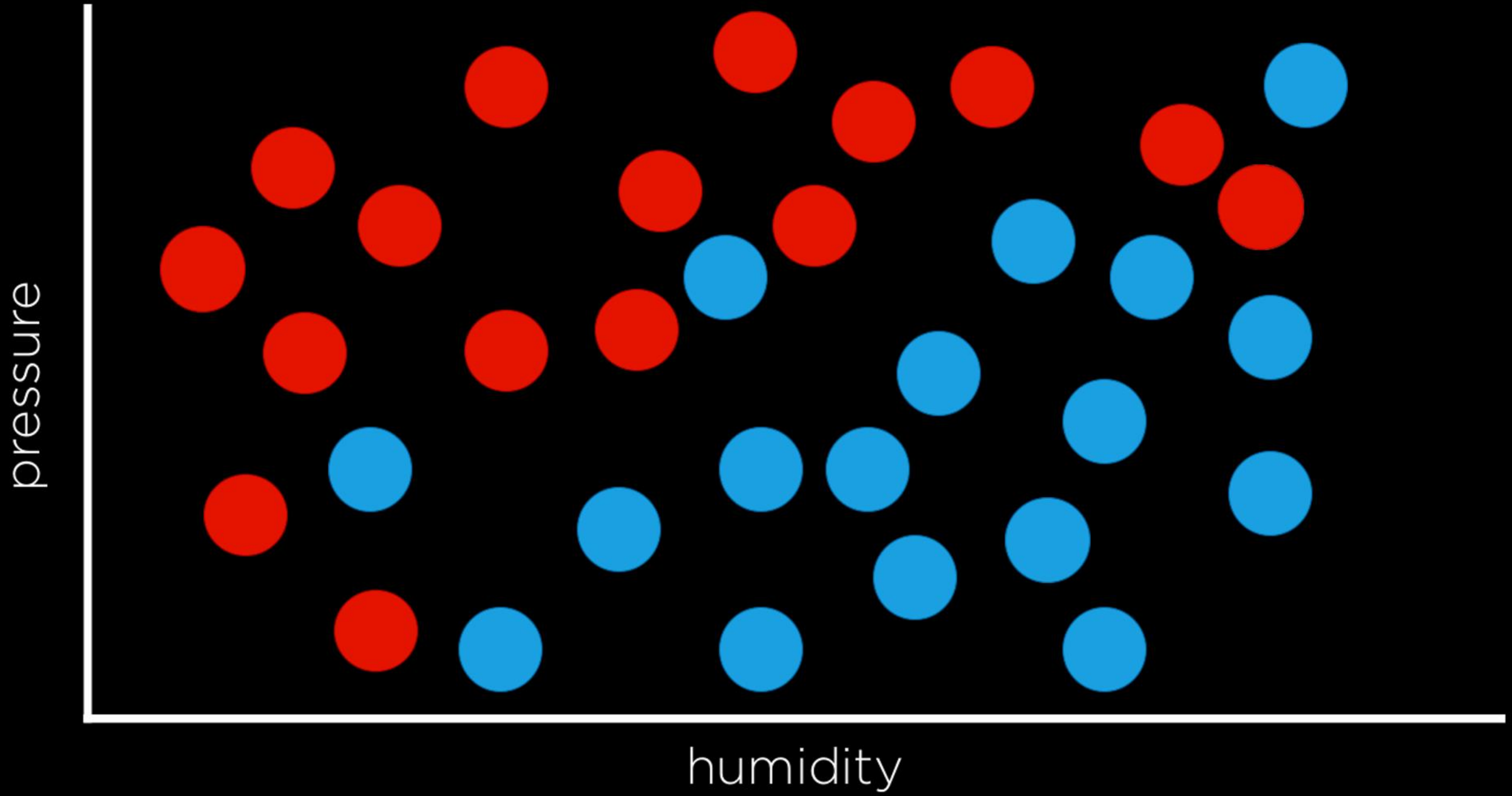
algorithm that, given an input, chooses the class of the nearest data point to that input


pressure
humidity


pressure
humidity

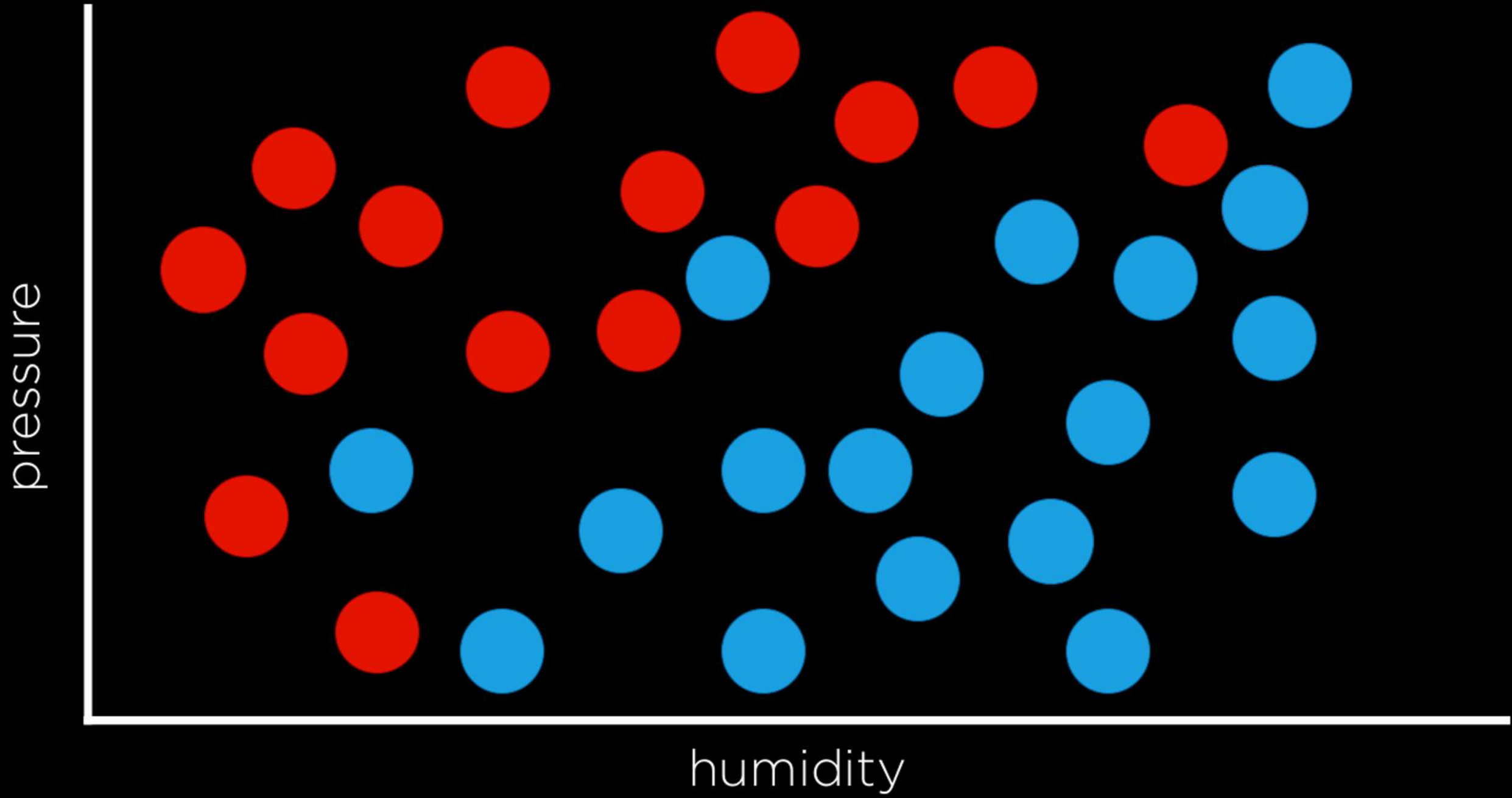

pressure
humidity


pressure
humidity

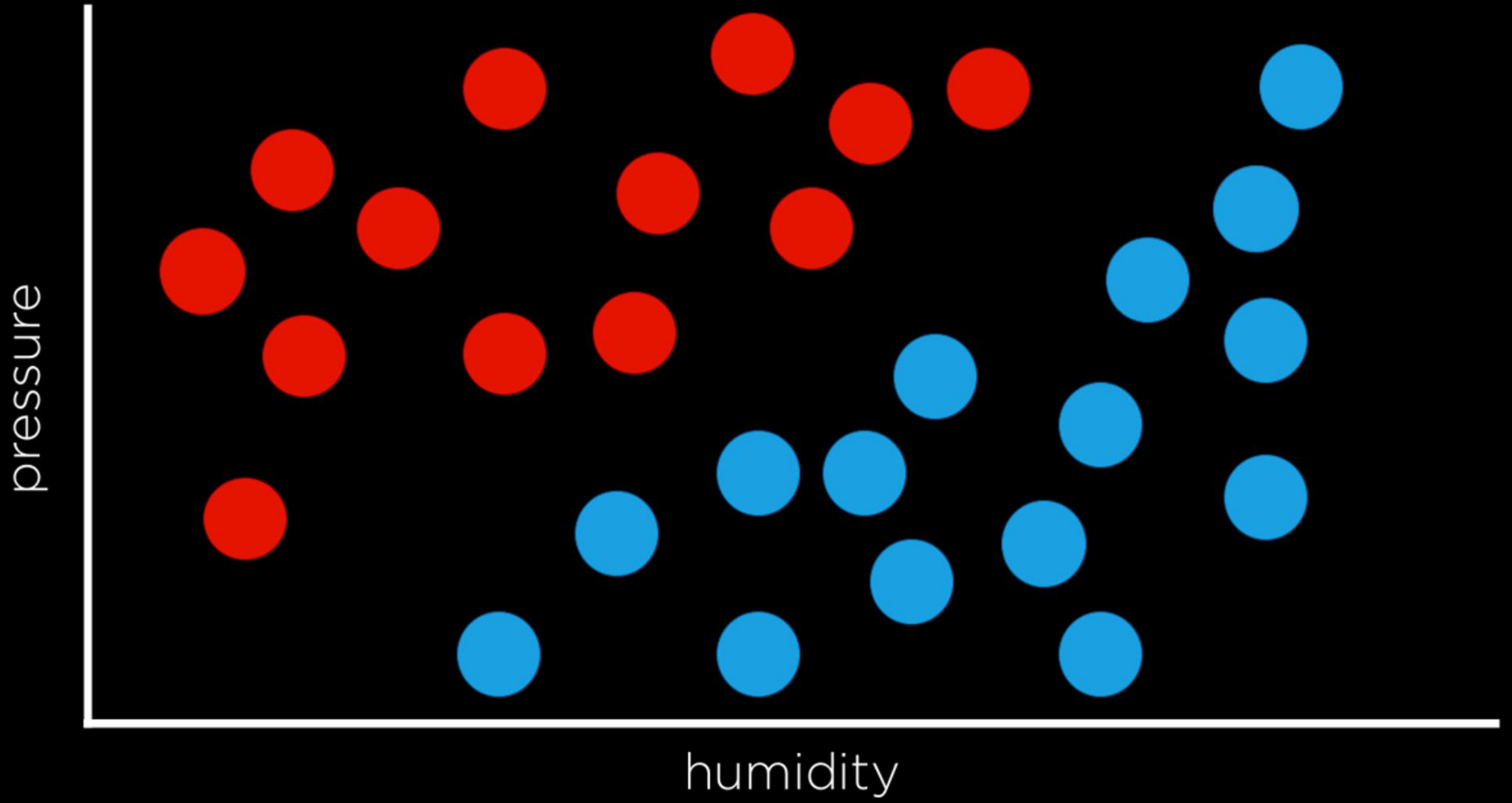

pressure
humidity

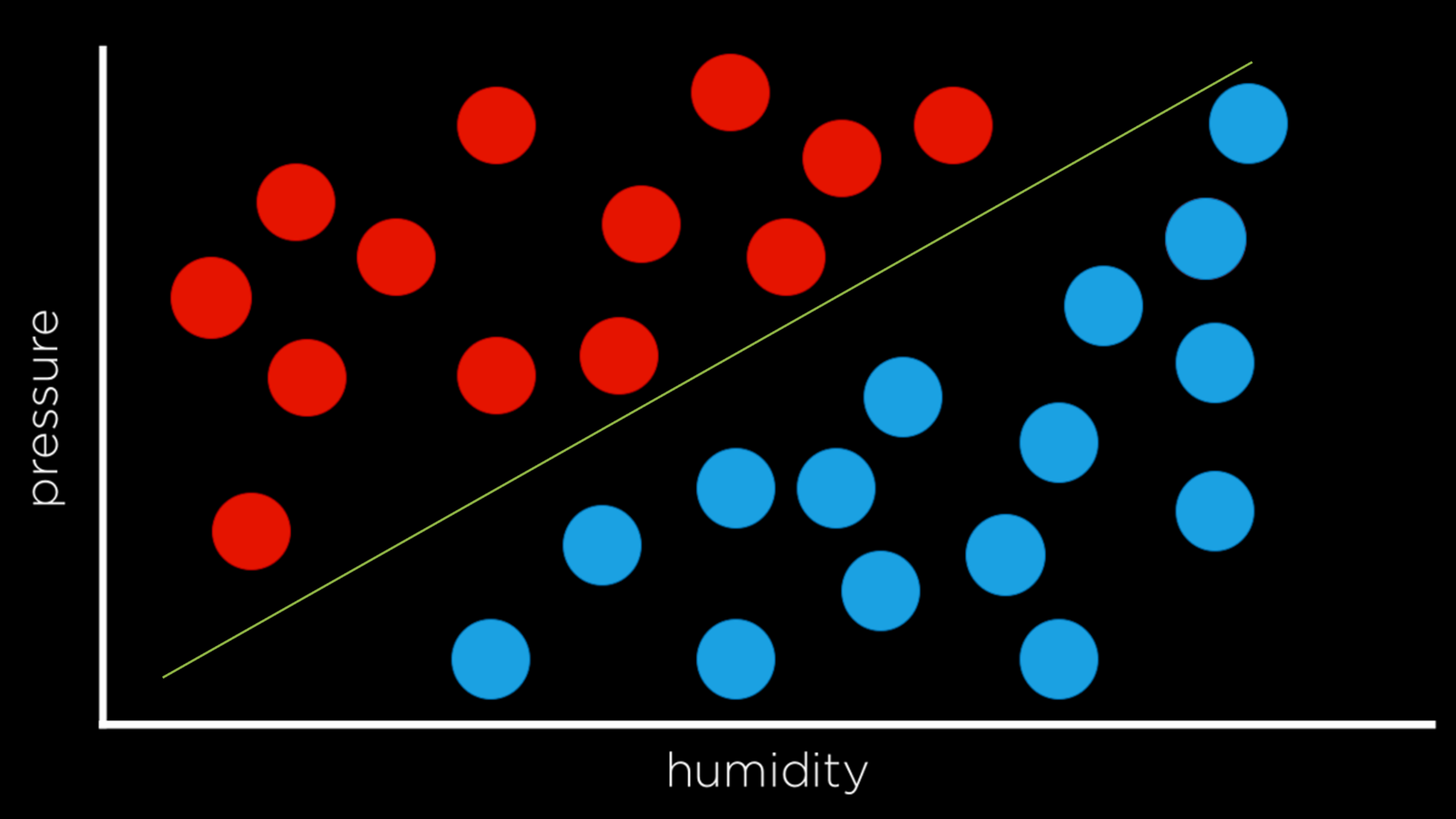
k-nearest-neighbor classification

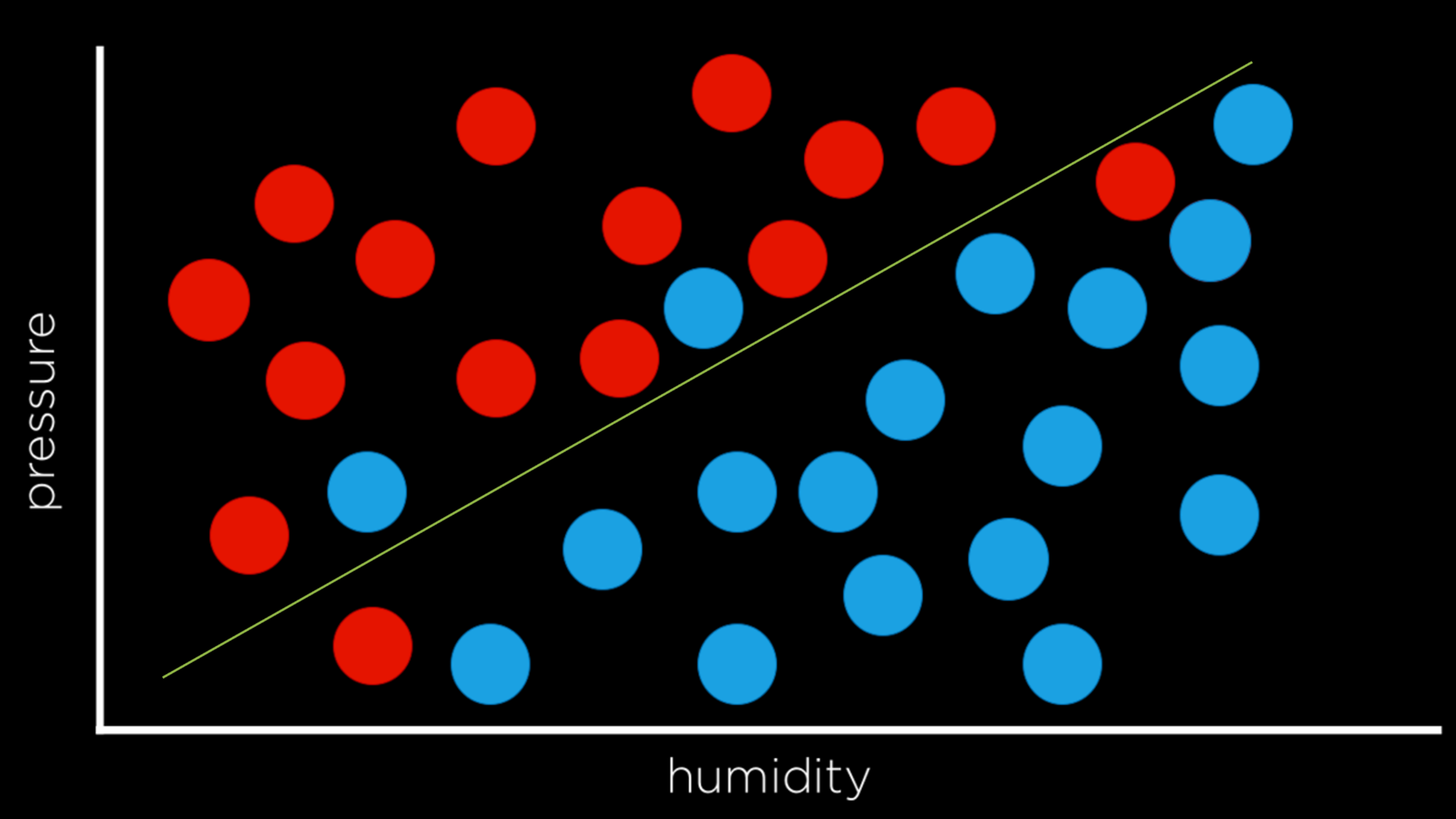
algorithm that, given an input, chooses the most common class out of the k nearest data points to that input


pressure
humidity

KNN has some drawbacks and challenges, such as computational expense, slow speed, memory and storage issues for large datasets, sensitivity to the choice of k and the distance metric, and susceptibility to the curse of dimensionality.


pressure
humidity


pressure
humidity


pressure
humidity

x_1 = Humidity

x_2 = Pressure

$$h(x_1, x_2) = \begin{cases} \text{Rain} & \text{if } w_0 + w_1 x_1 + w_2 x_2 \geq 0 \\ \text{No Rain} & \text{otherwise} \end{cases}$$

Weight Vector $\mathbf{w}$: (w_0, w_1, w_2)

Input Vector $\mathbf{x}$: $(1, x_1, x_2)$

$\mathbf{w} \cdot \mathbf{x}$: $w_0 + w_1 x_1 + w_2 x_2$

$$h(x_1, x_2) = \begin{cases} 1 \text{ if } w_0 + w_1 x_1 + w_2 x_2 \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

Weight Vector $\mathbf{w}$: (w_0, w_1, w_2)

Input Vector $\mathbf{x}$: $(1, x_1, x_2)$

$\mathbf{w} \cdot \mathbf{x}$: $w_0 + w_1 x_1 + w_2 x_2$

$$h_{\mathbf{w}}(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{w} \cdot \mathbf{x} \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

perceptron learning rule

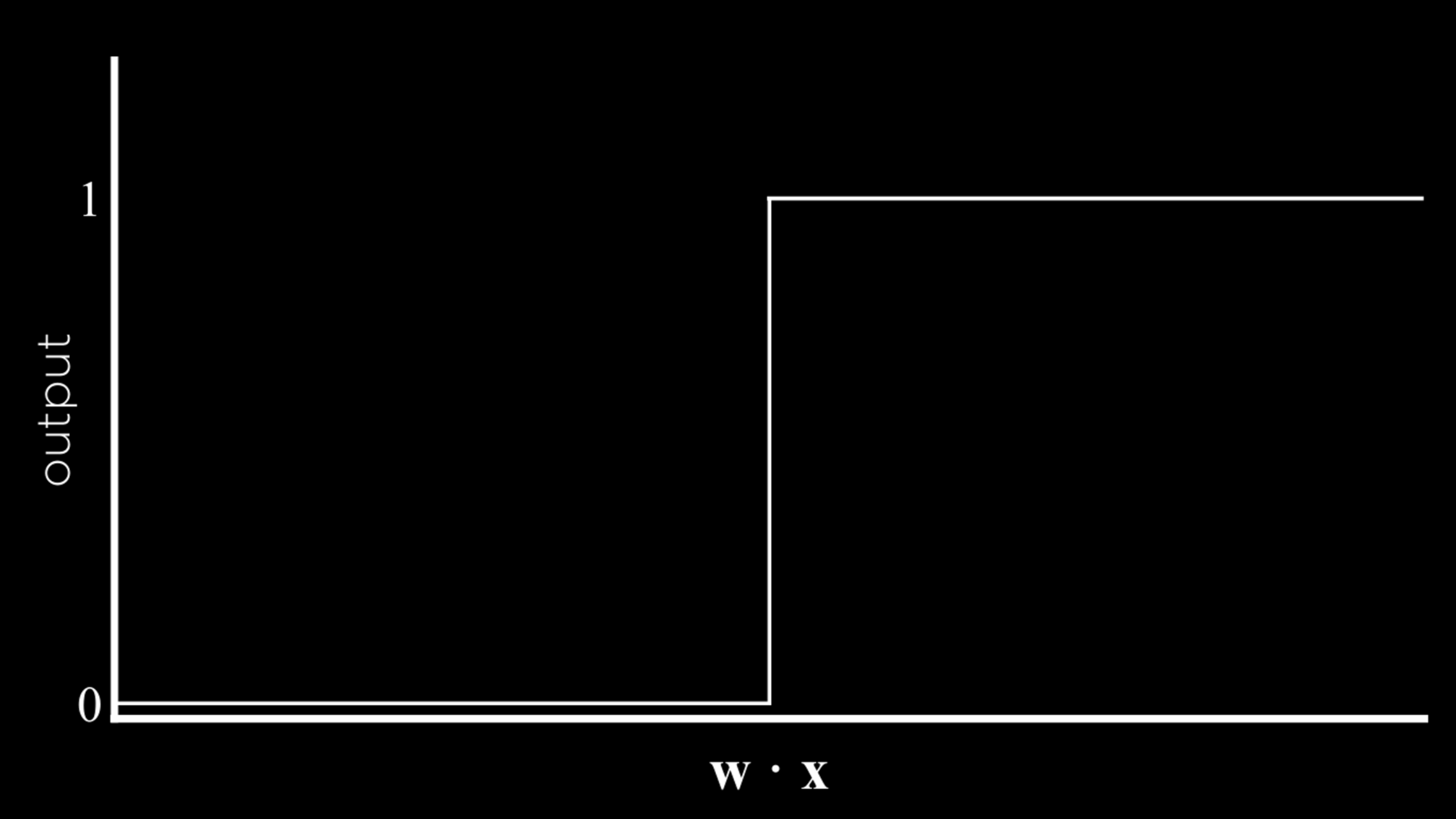
Given data point $(\mathbf{x}, \mathbf{y})$, update each weight according to:

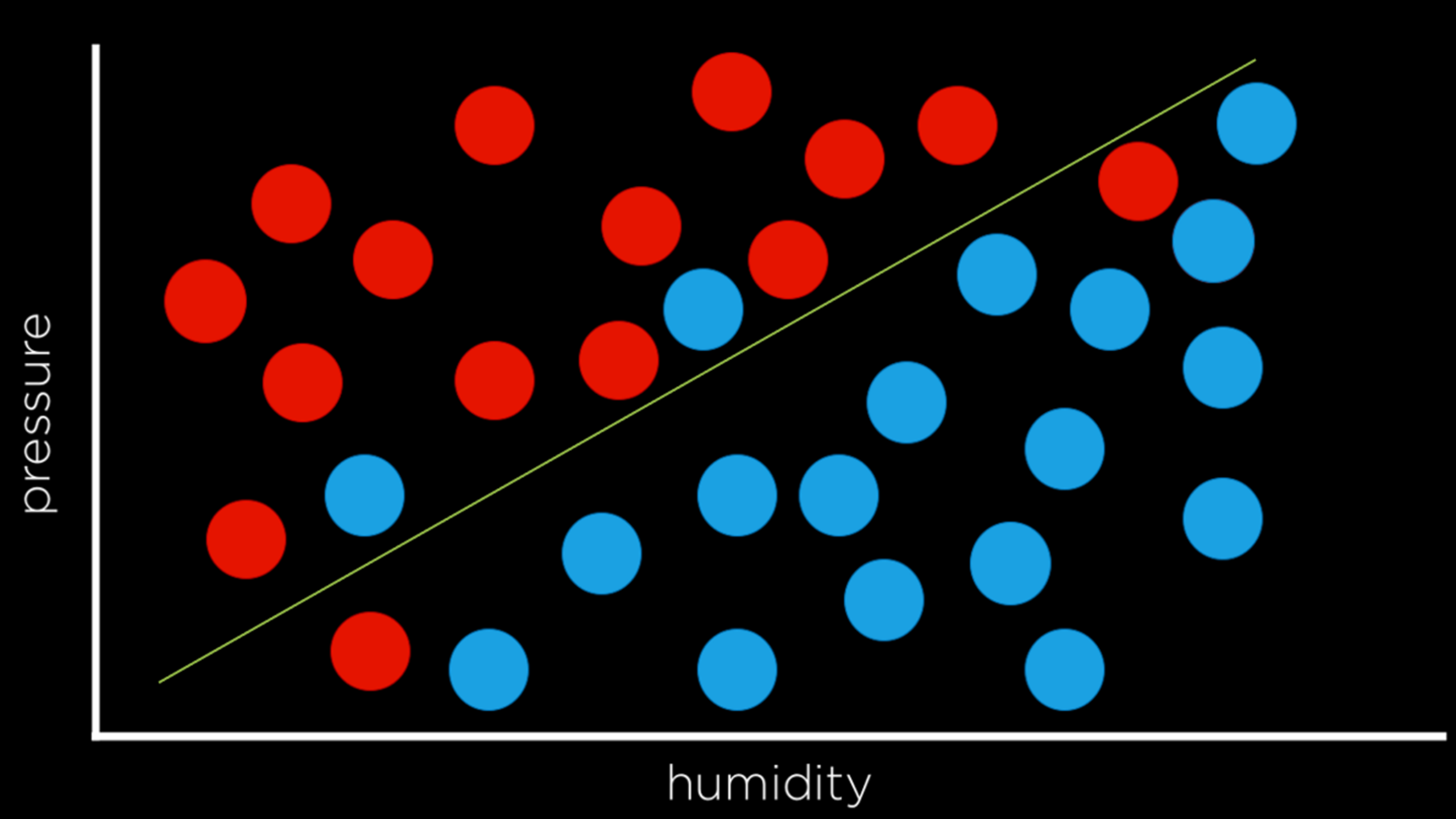
$$w_i = w_i + \alpha(\mathbf{y} - h_{\mathbf{w}}(\mathbf{x})) \times x_i$$

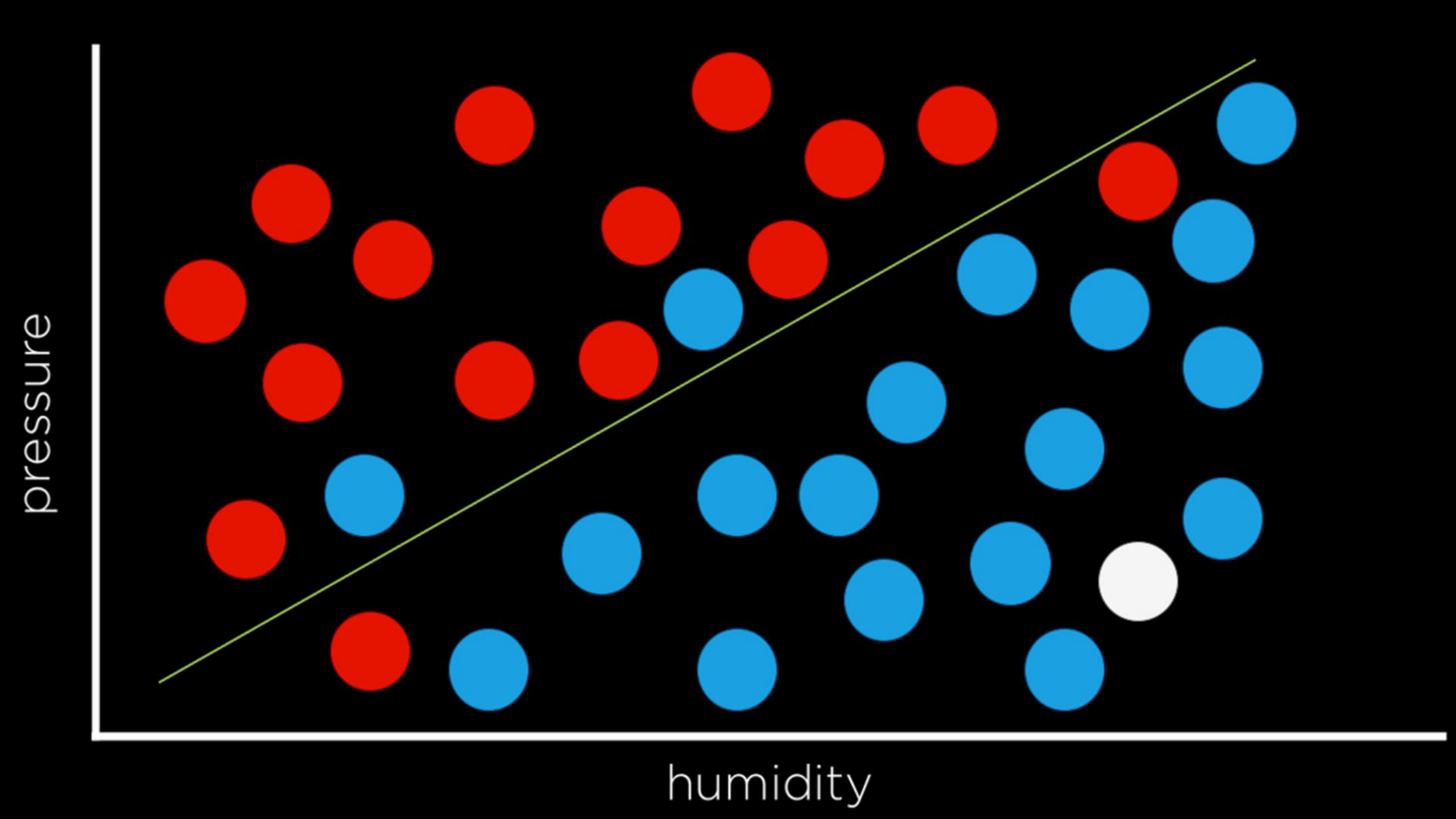
perceptron learning rule

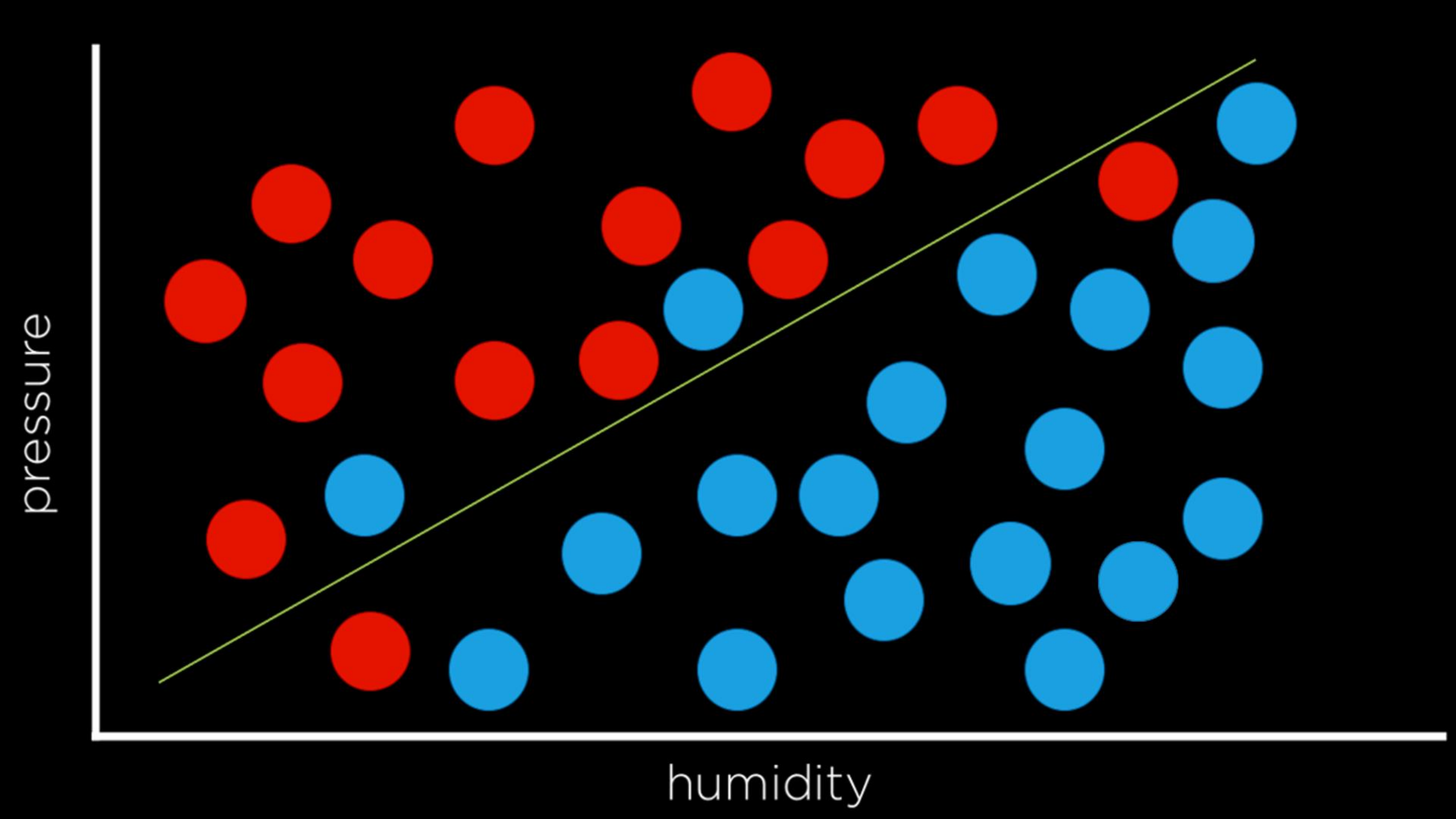
Given data point $(\mathbf{x}, \mathbf{y})$, update each weight according to:

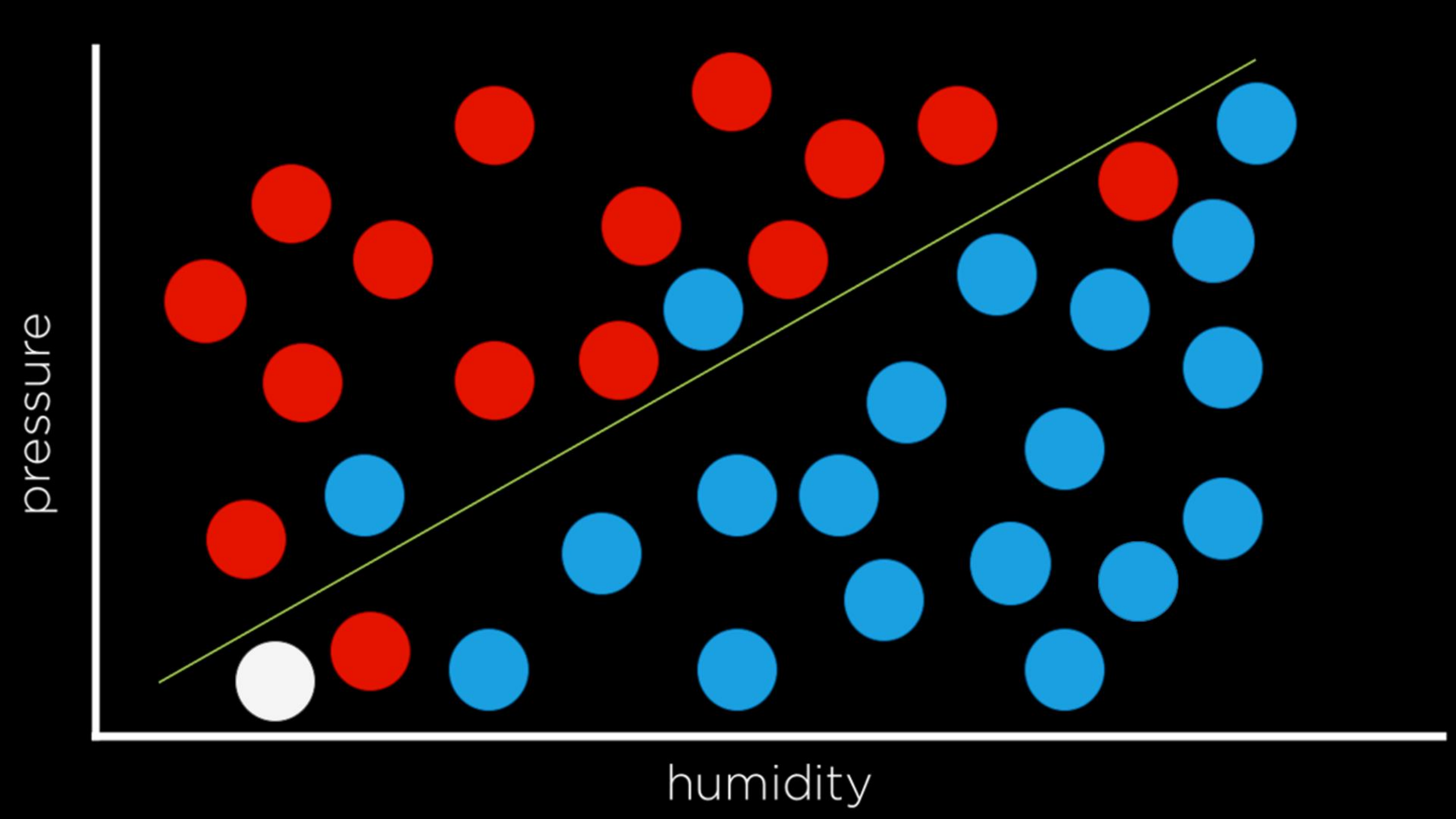
$$w_i = w_i + \alpha(\text{actual value - estimate}) \times x_i$$

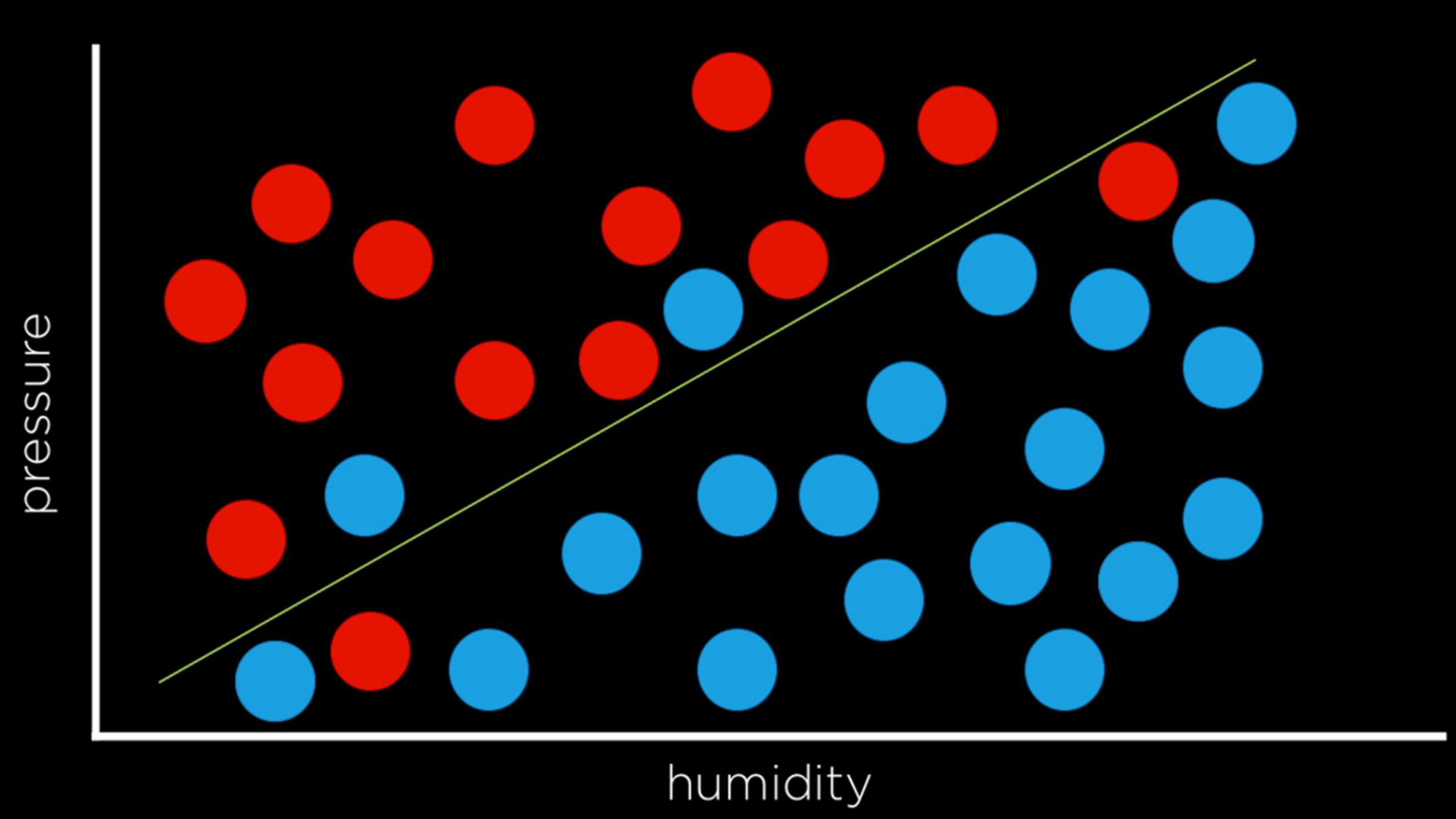

1
output
0
w · x

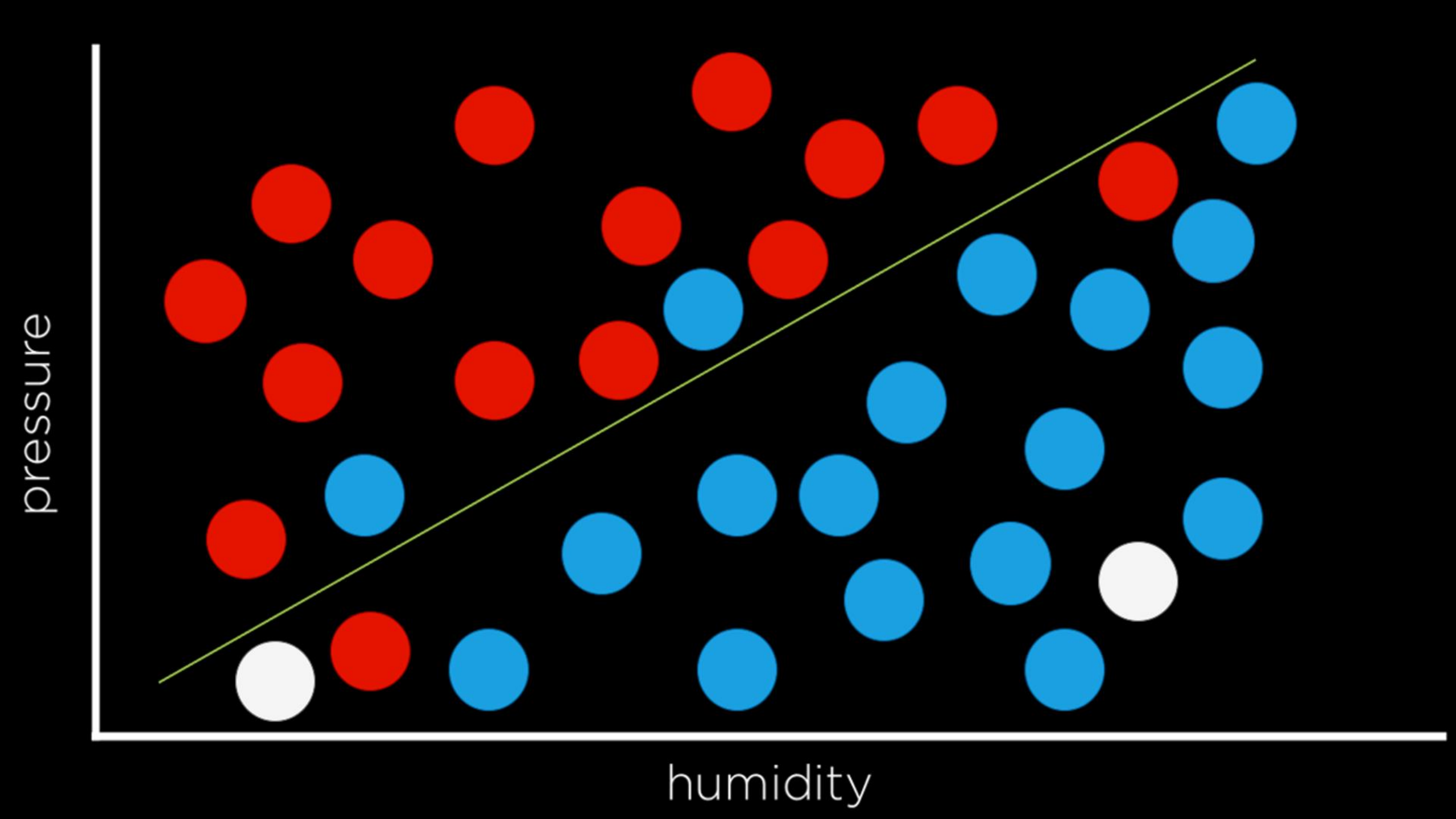

pressure
humidity

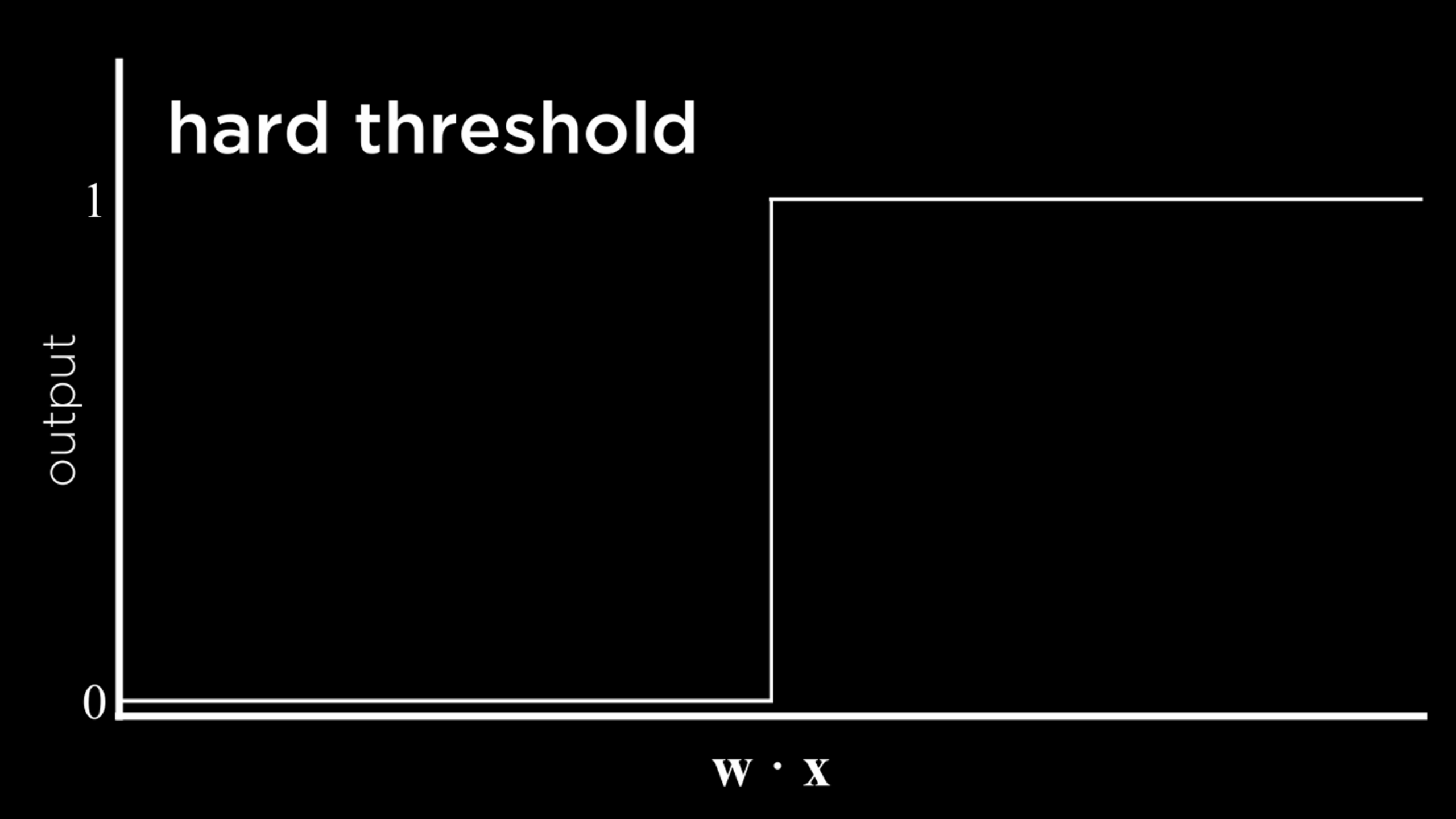

pressure
humidity

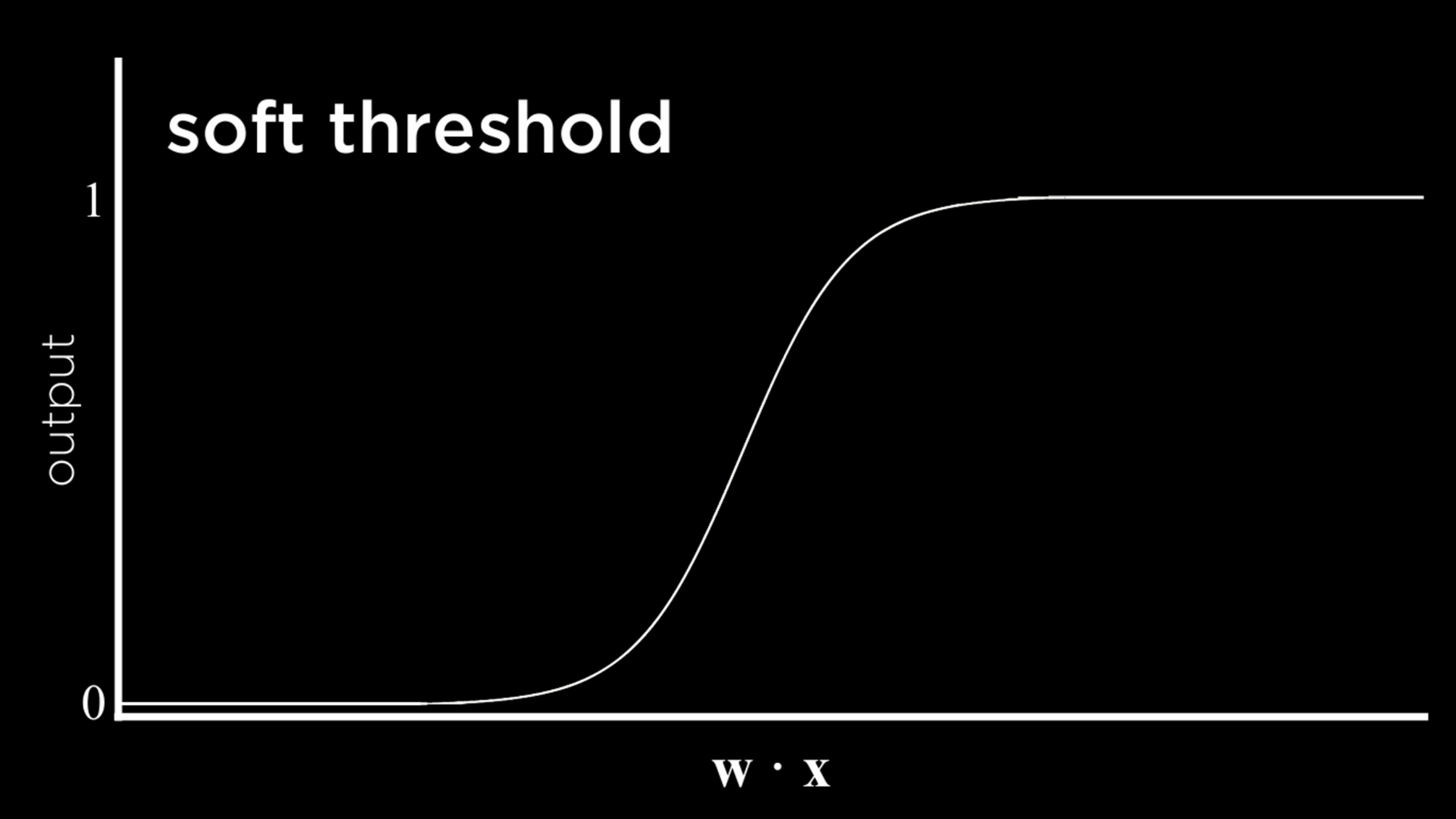

pressure
humidity

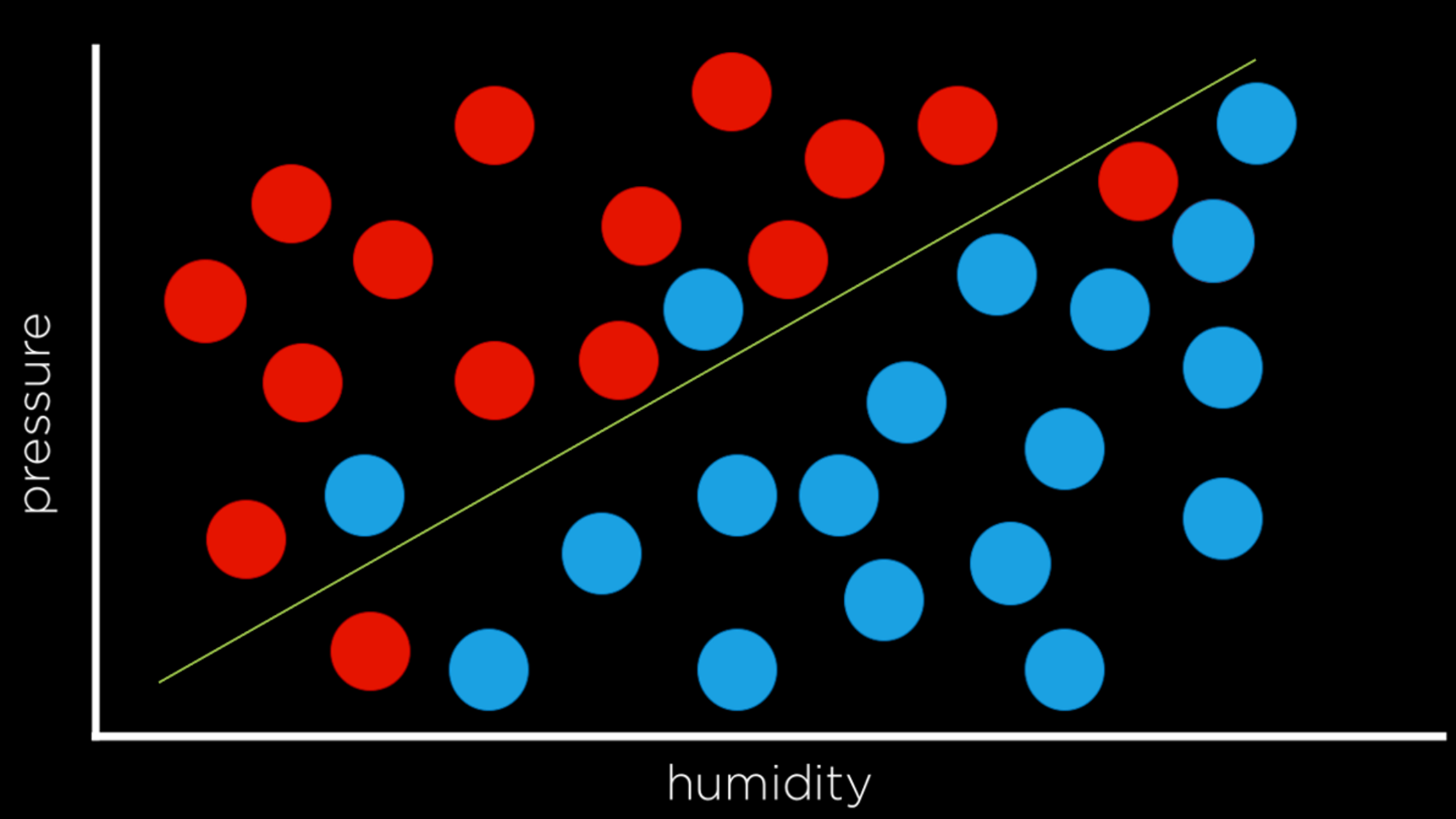

pressure
humidity

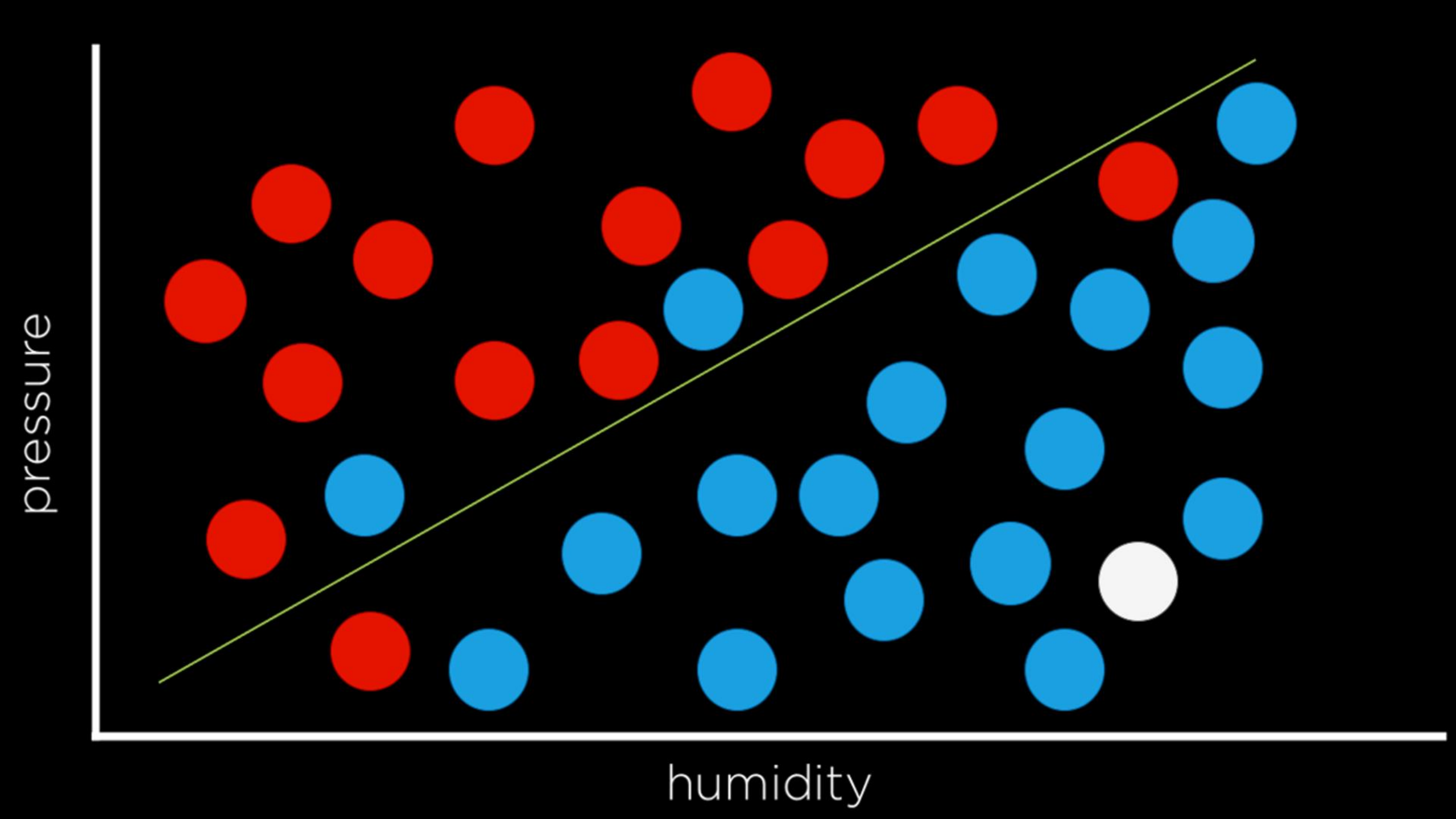

pressure
humidity

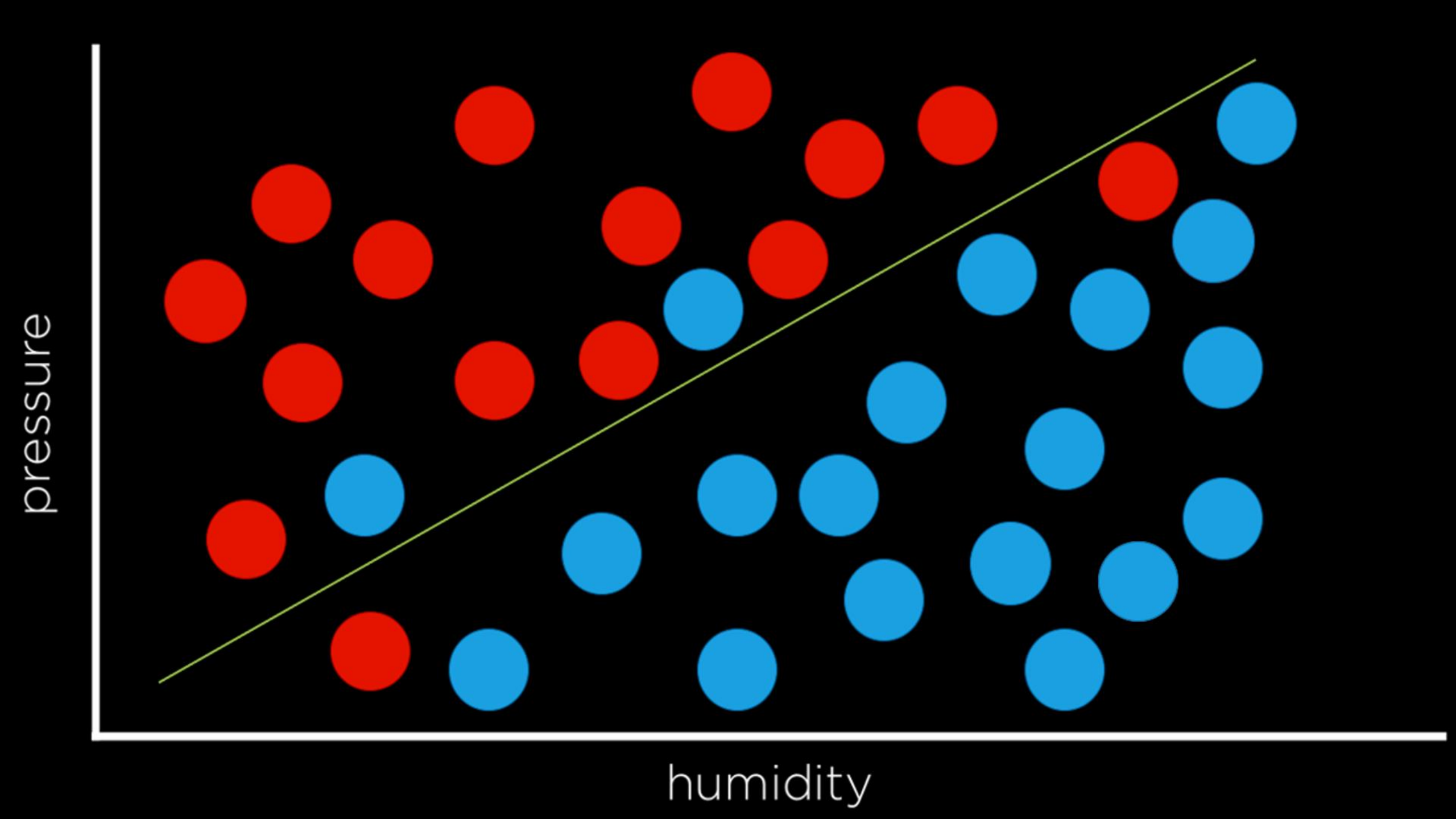

pressure
humidity

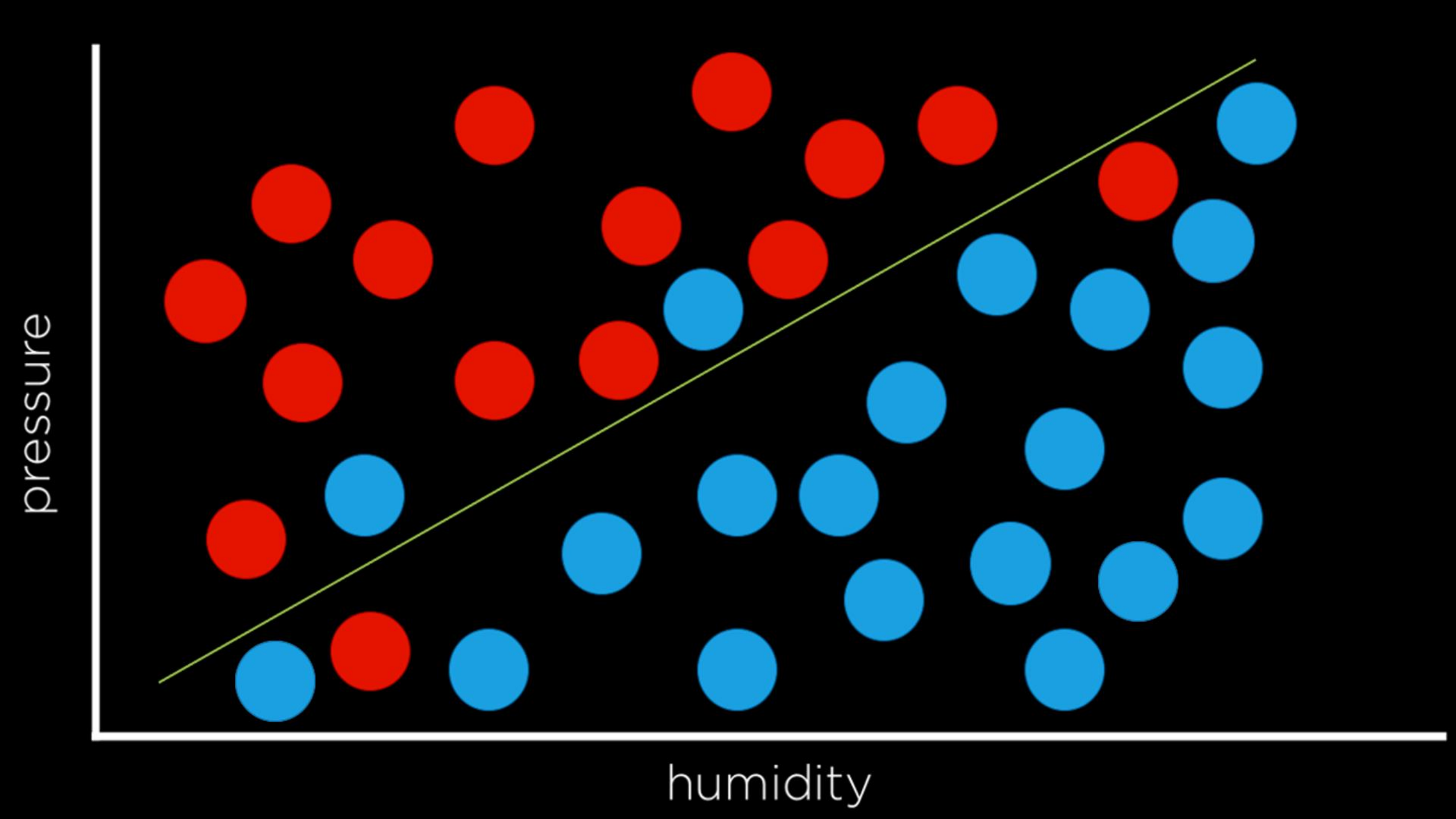

hard threshold
1
0
output
w · x


soft threshold
1
output
0
w · x


pressure
humidity


pressure
humidity

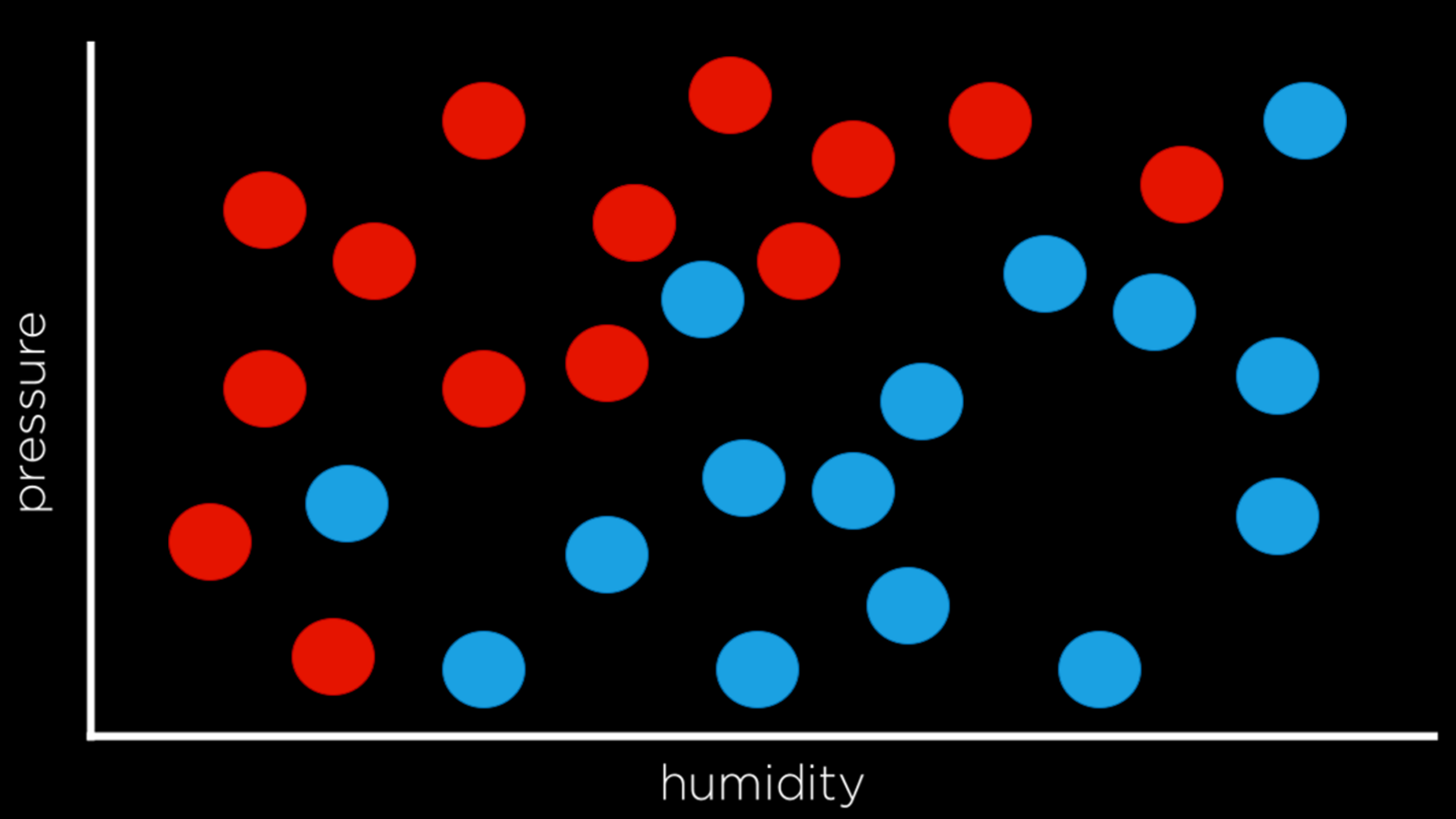

pressure
humidity

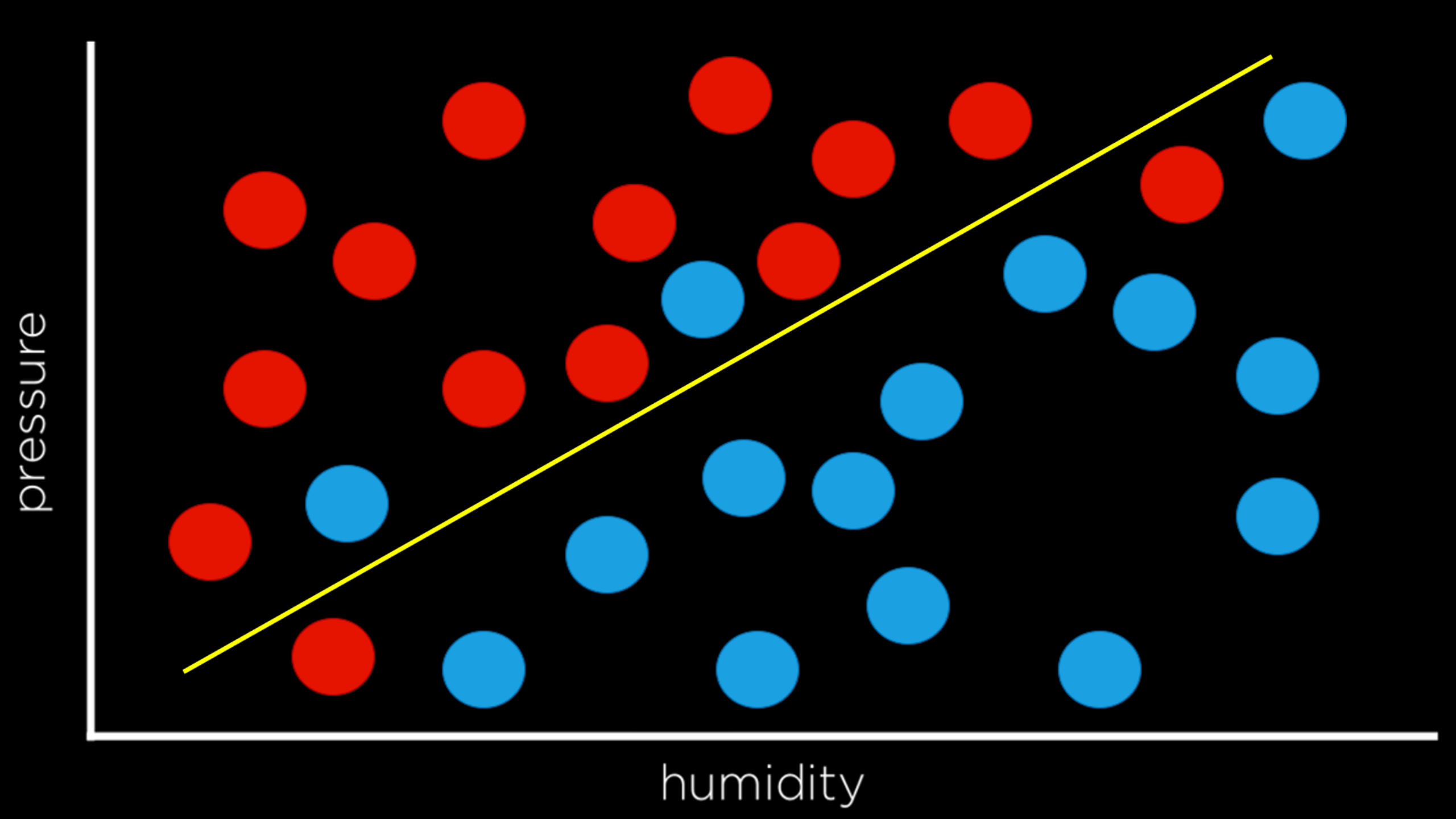

pressure
humidity

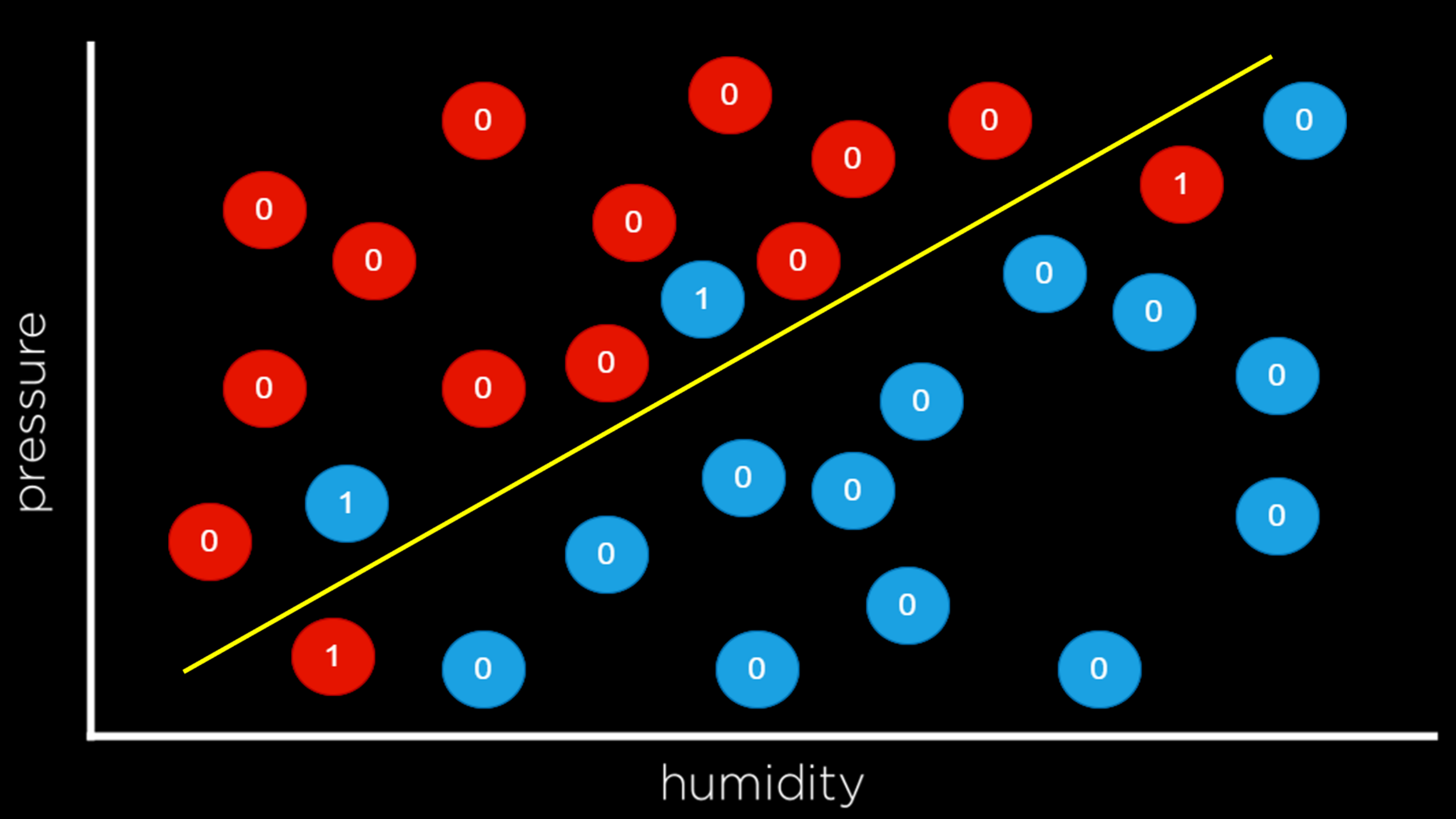
Evaluating Hypotheses

loss function

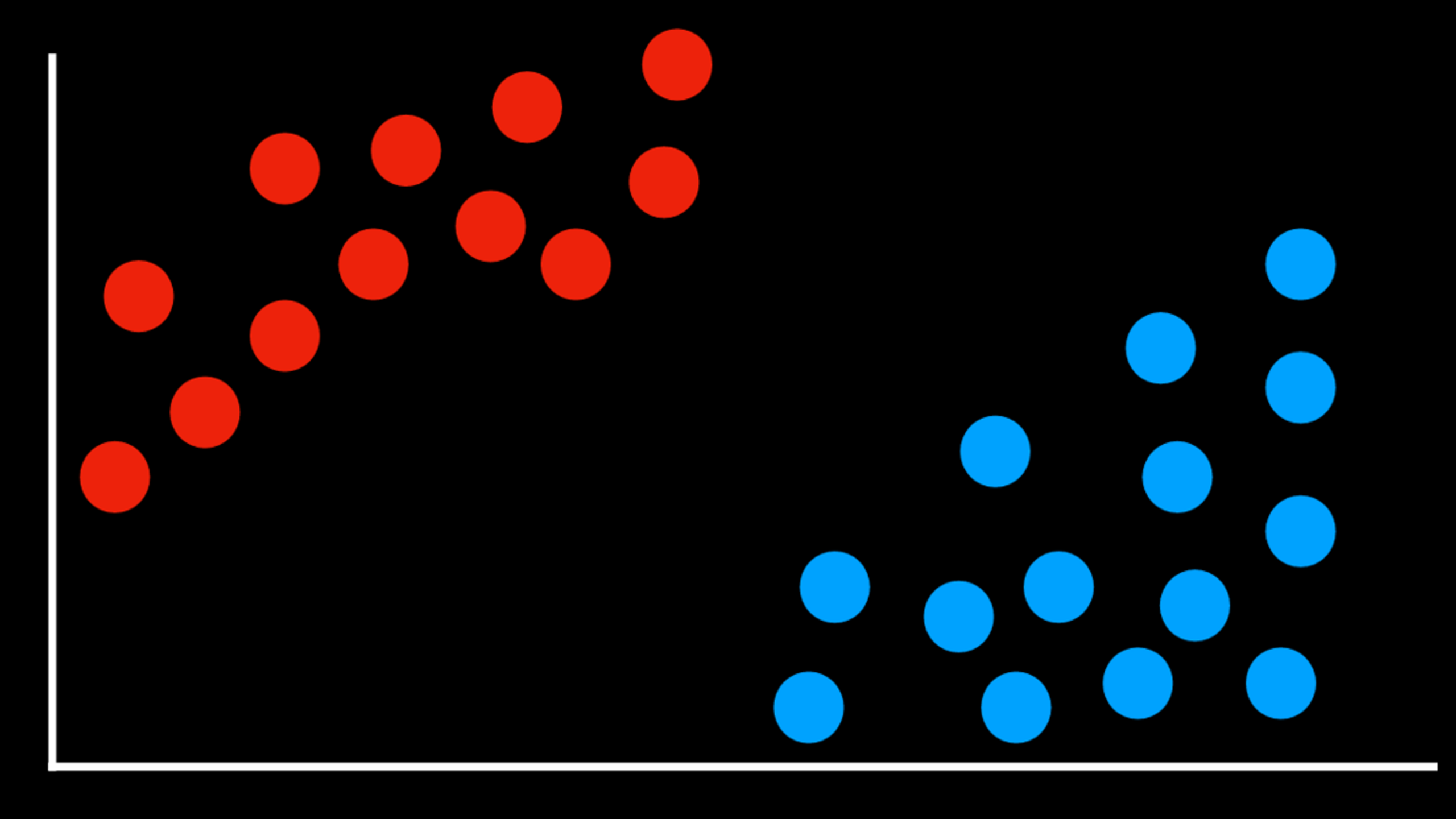
function that expresses how poorly our hypothesis performs

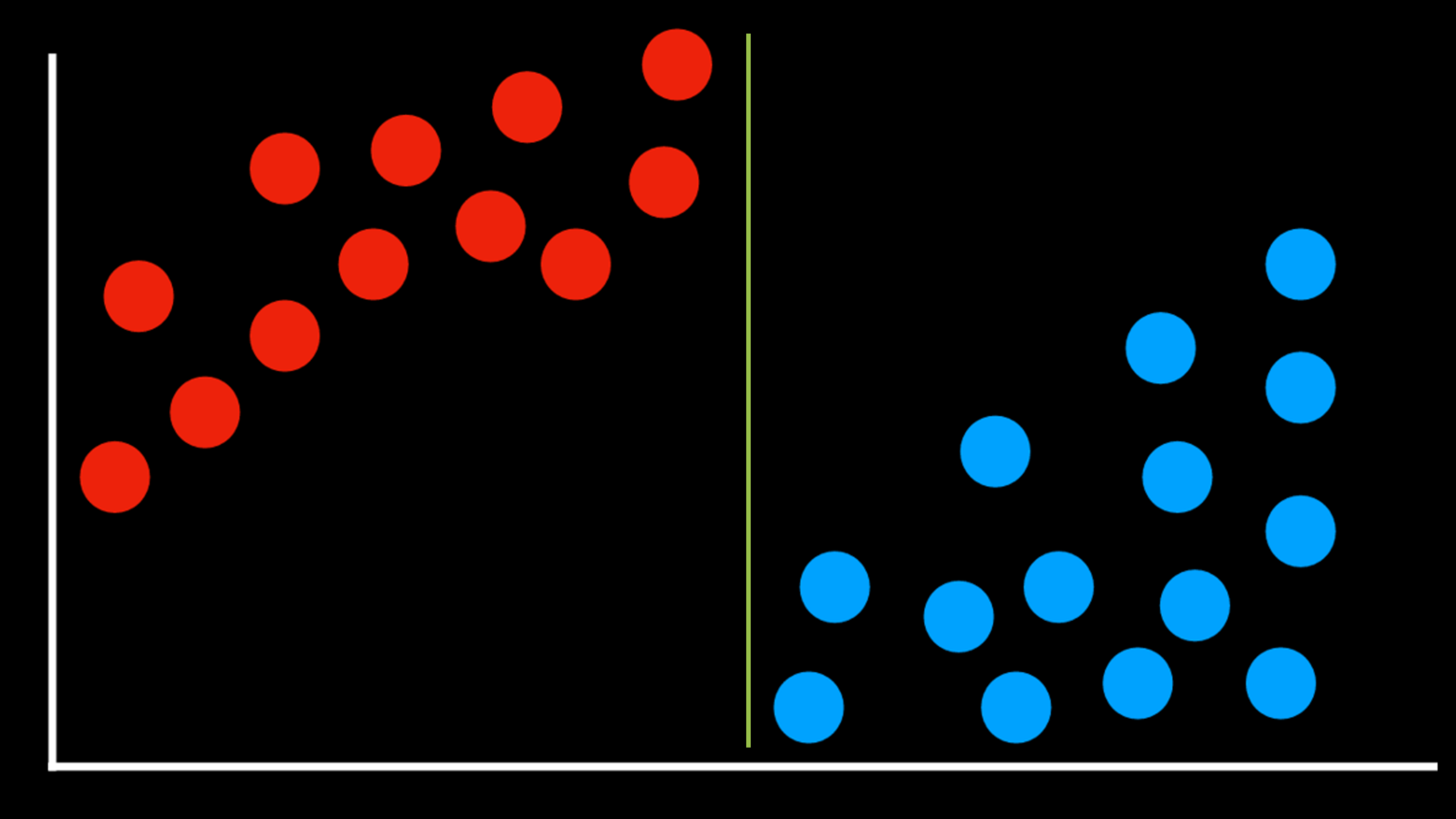

pressure
humidity

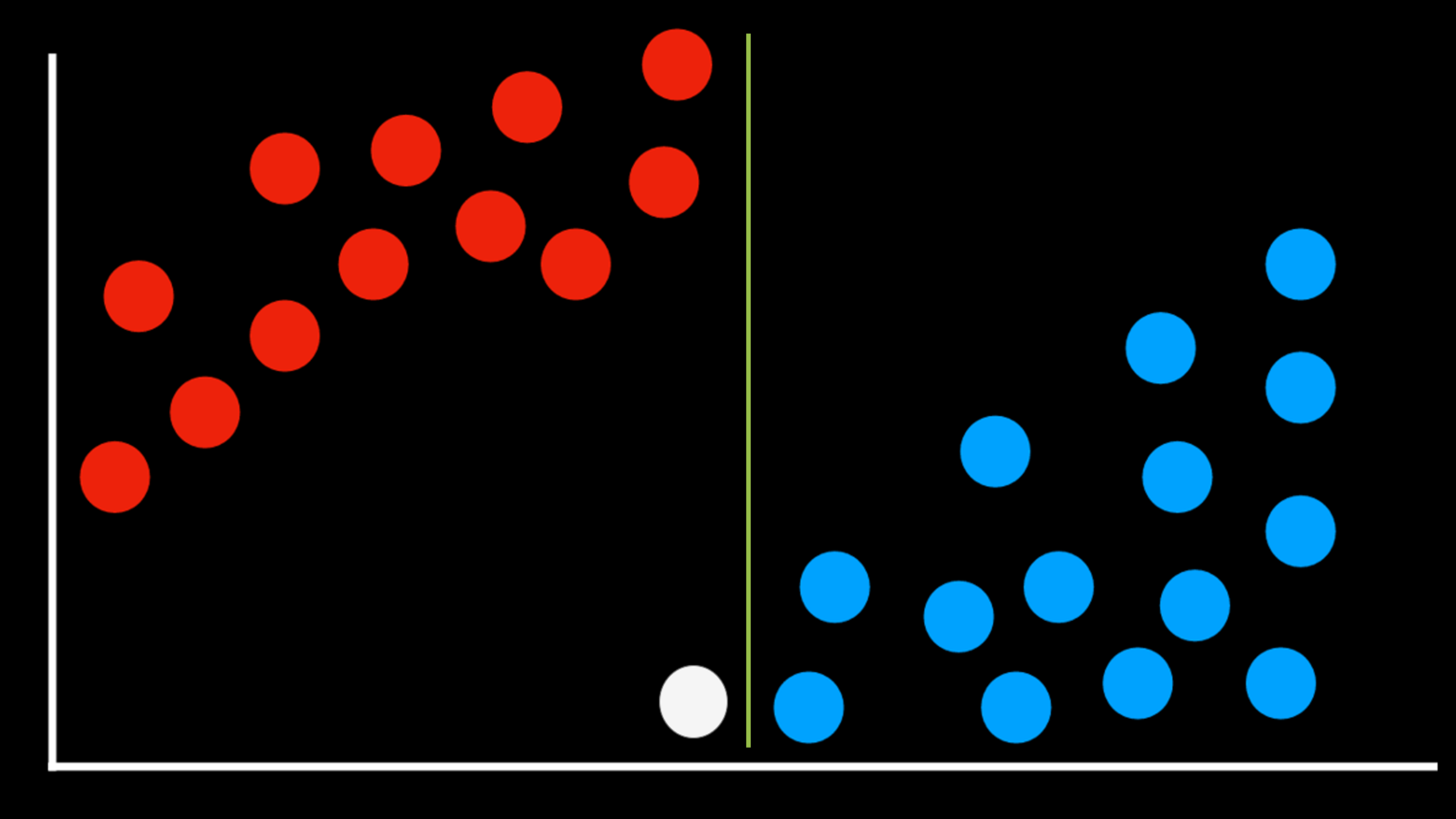

pressure
humidity

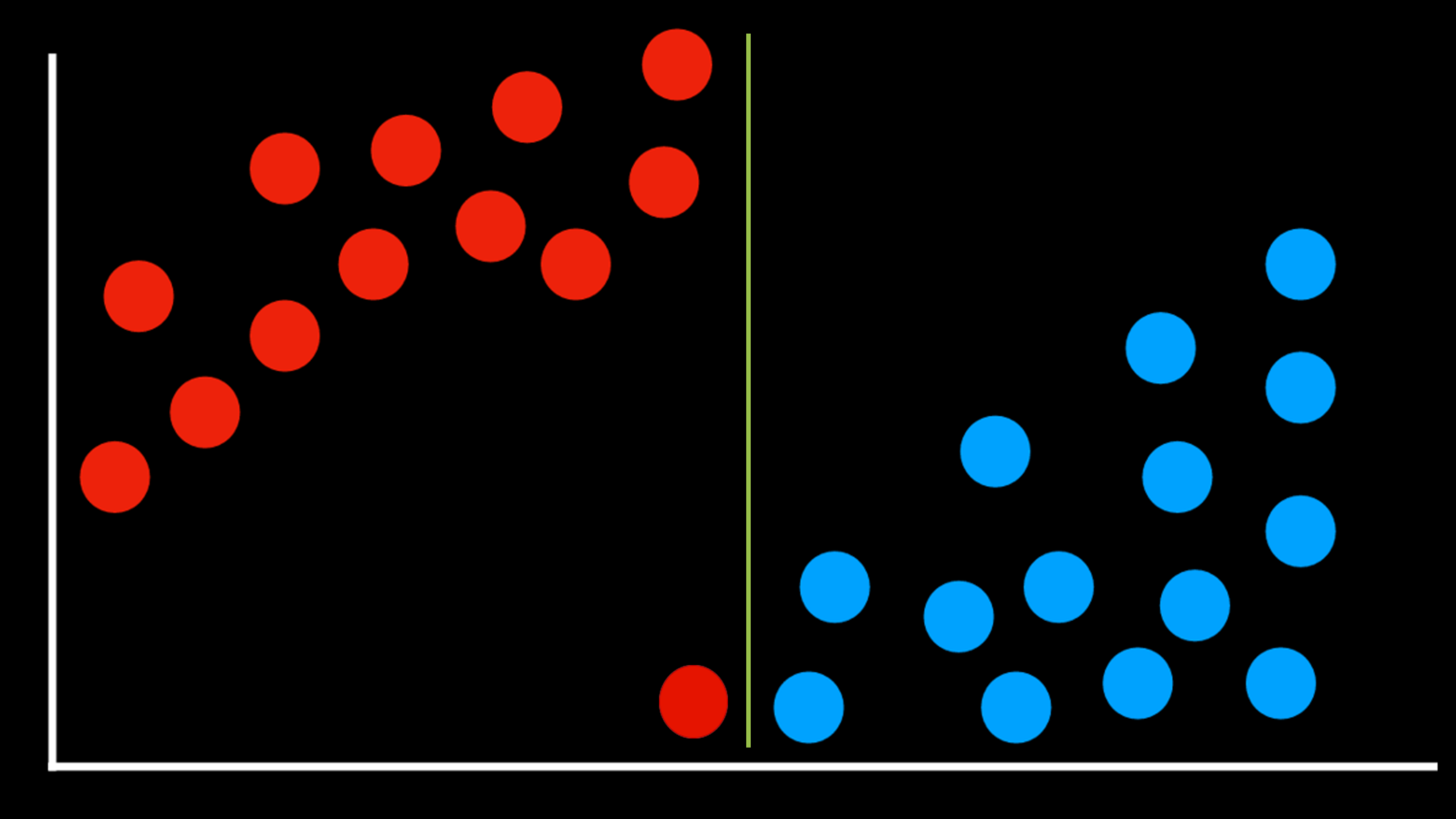

pressure
humidity
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
1
1
1
1

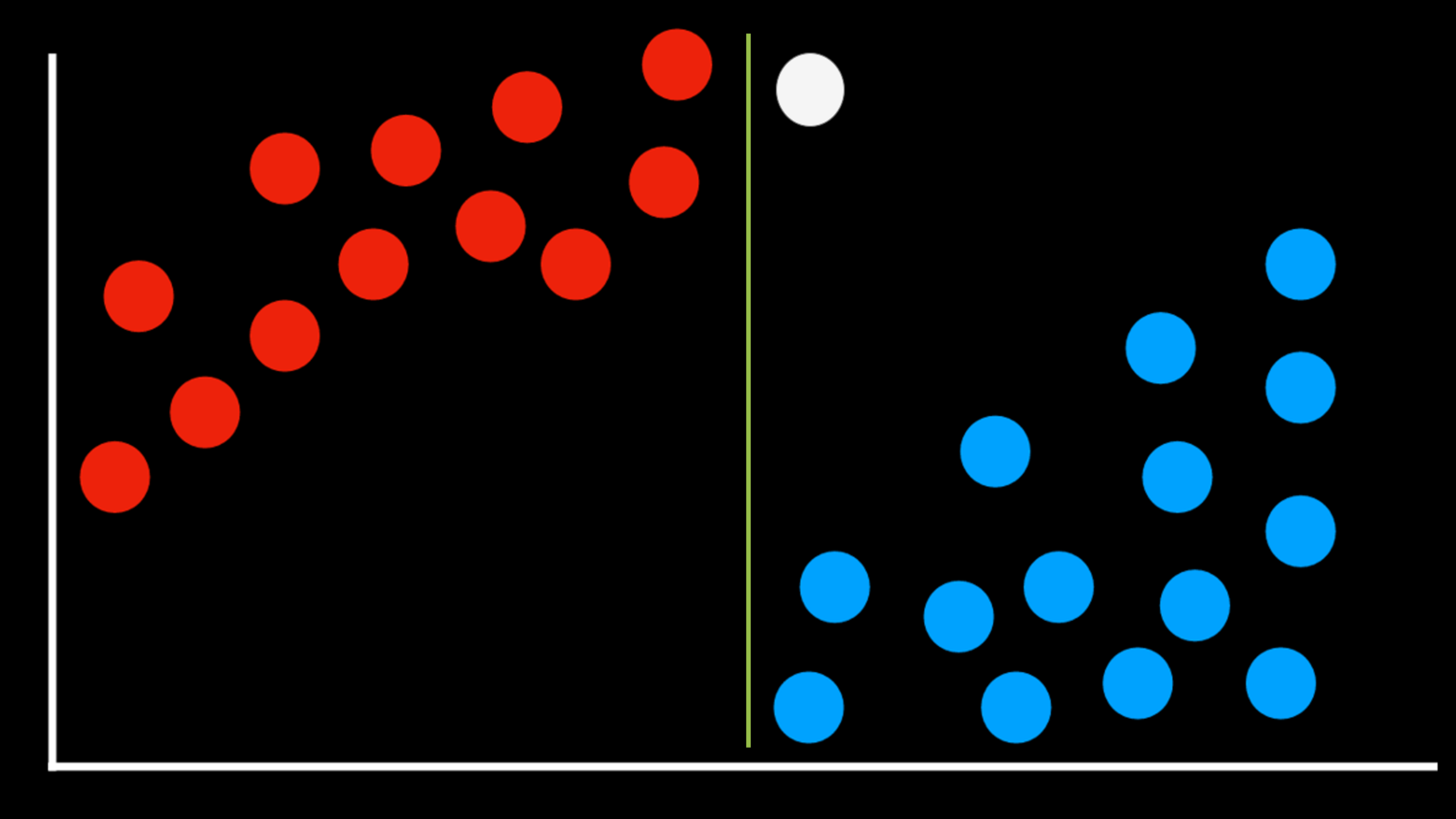
Support Vector Machines

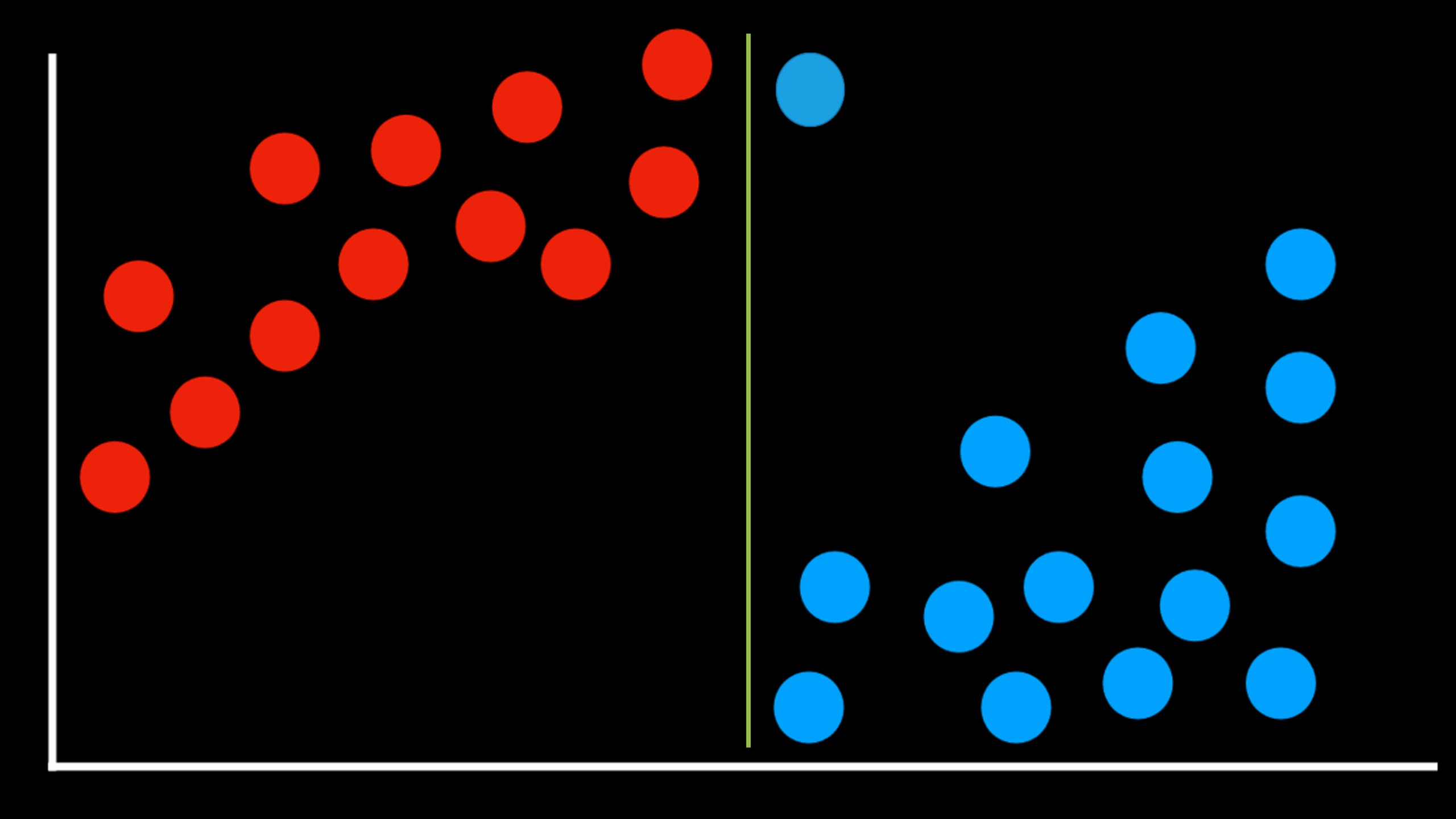


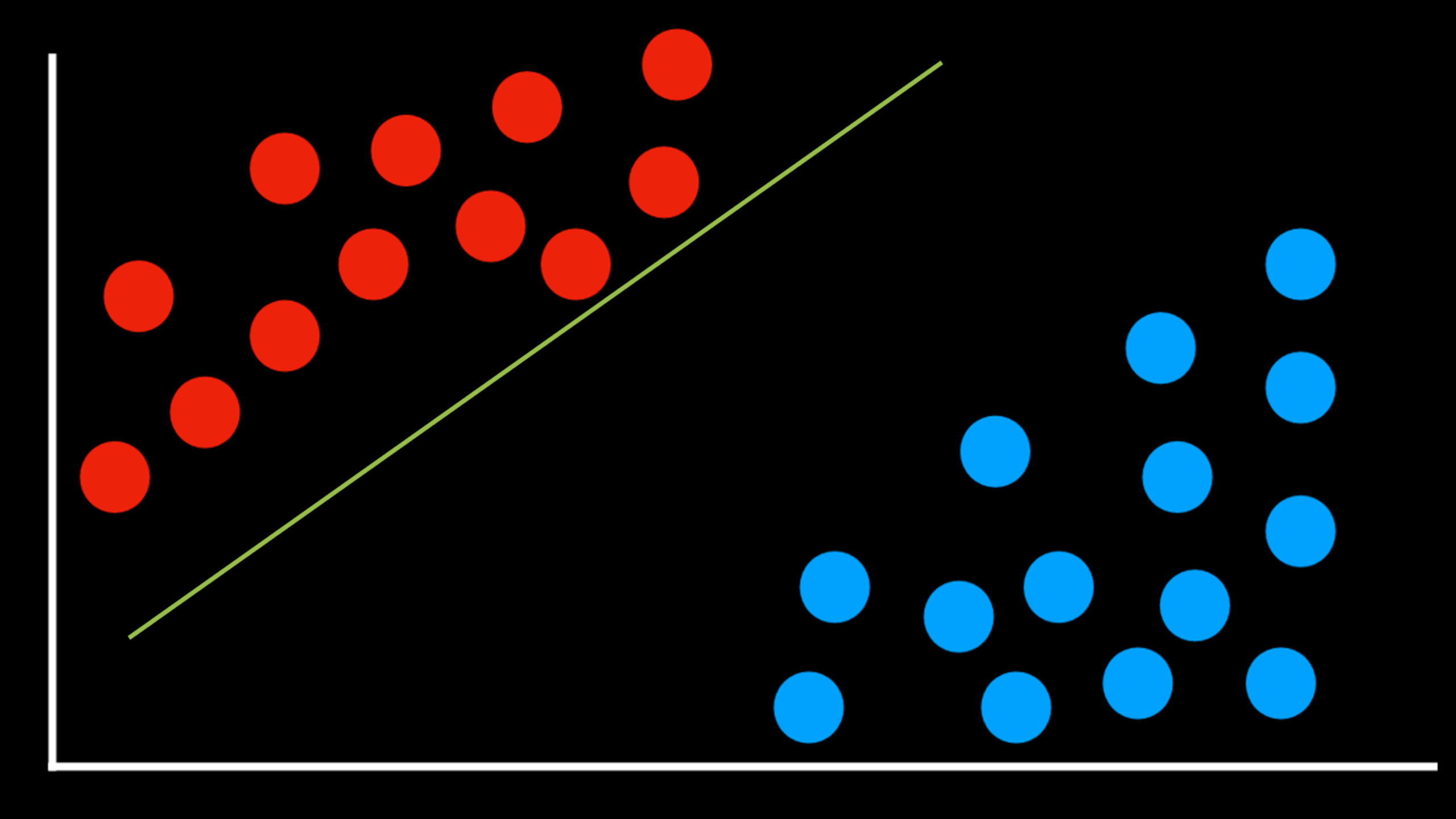


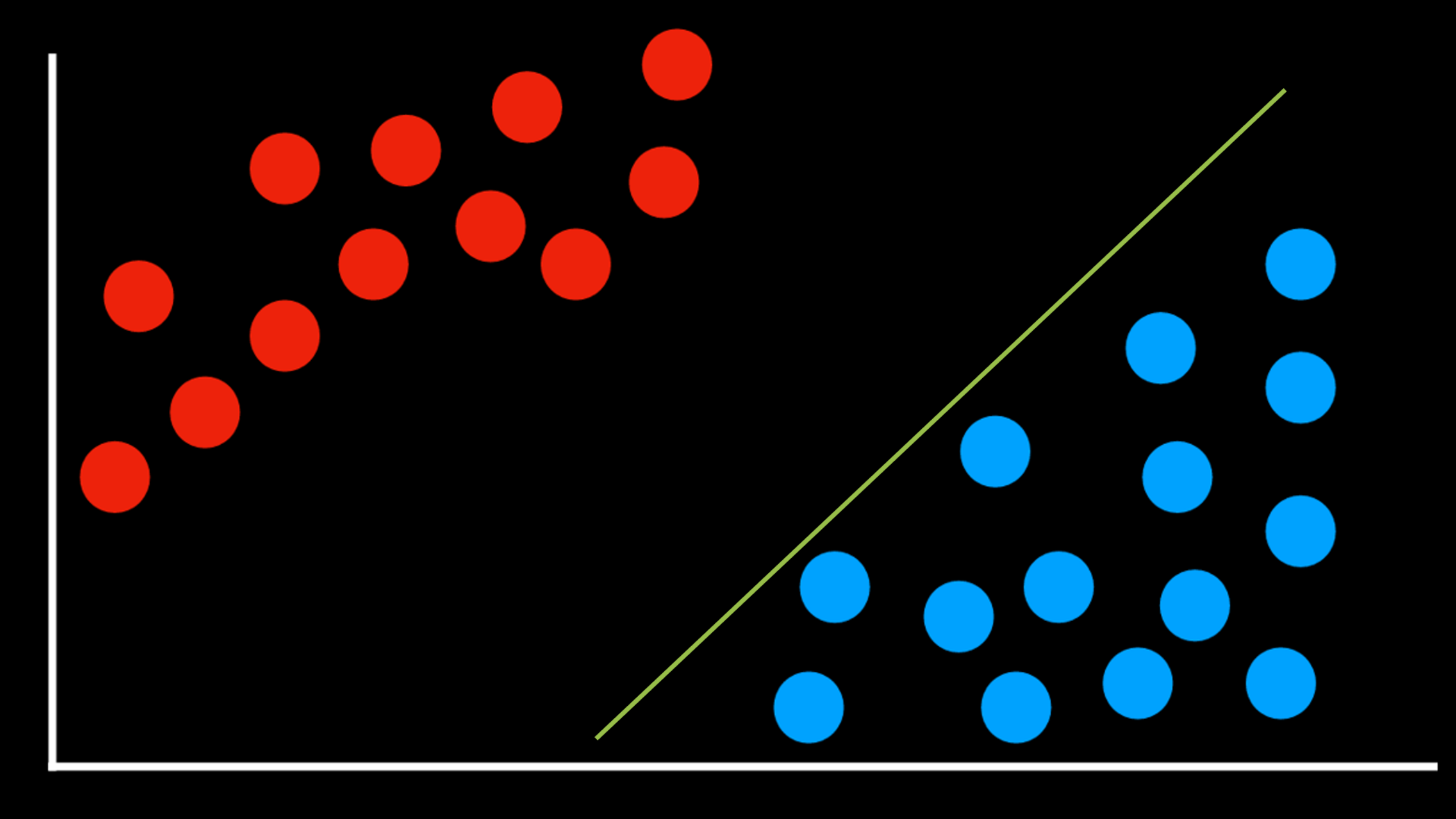


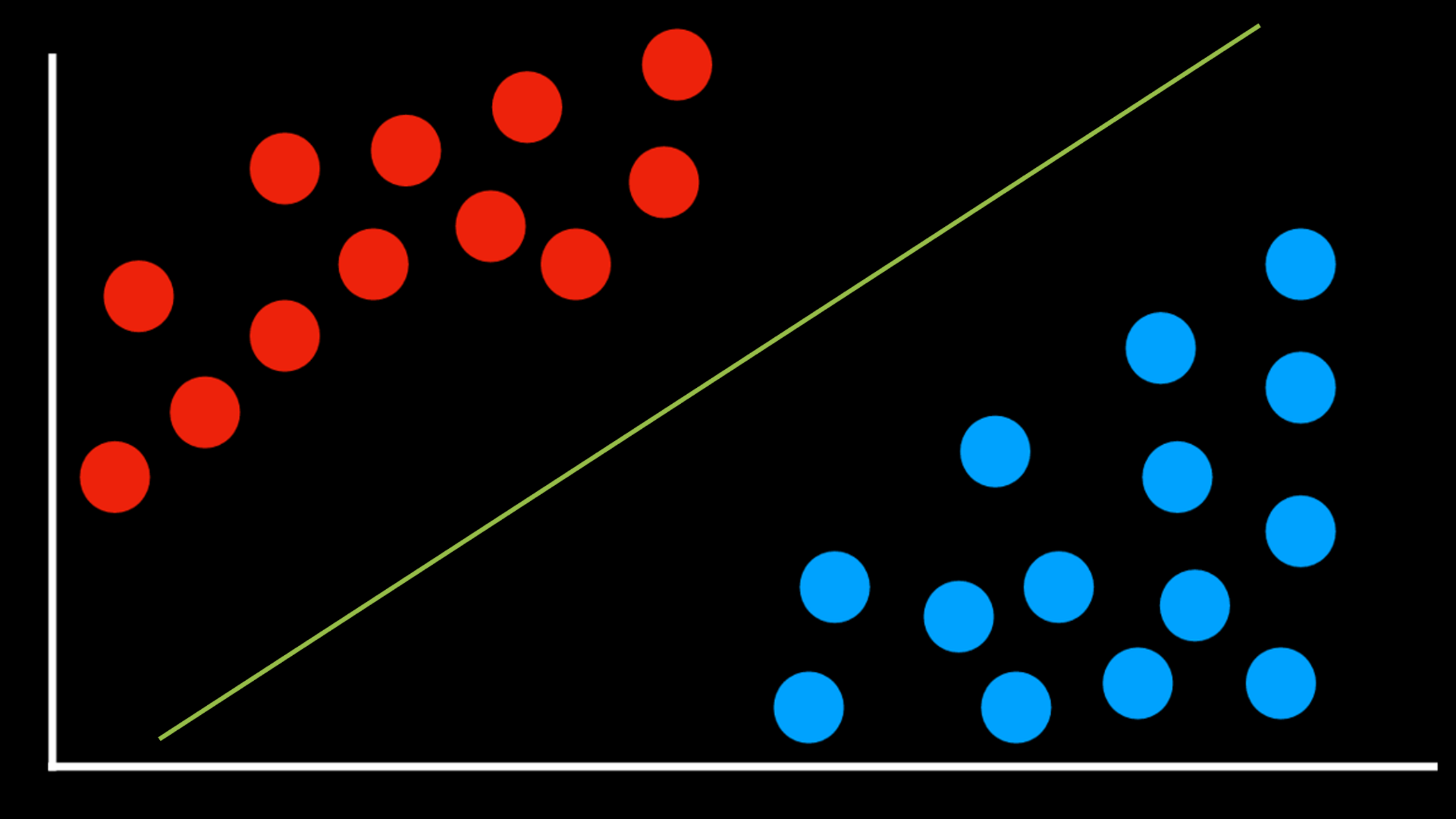






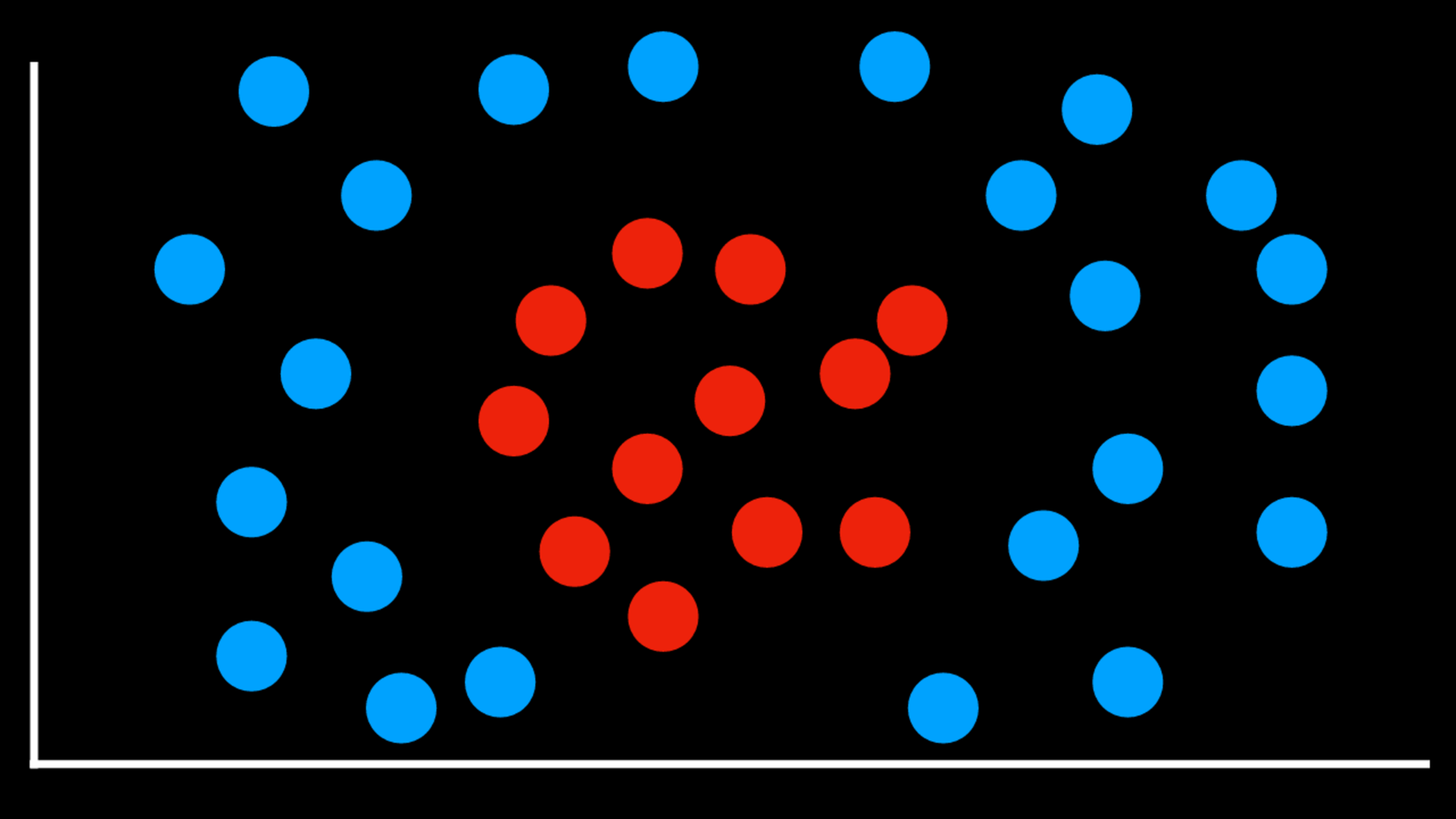


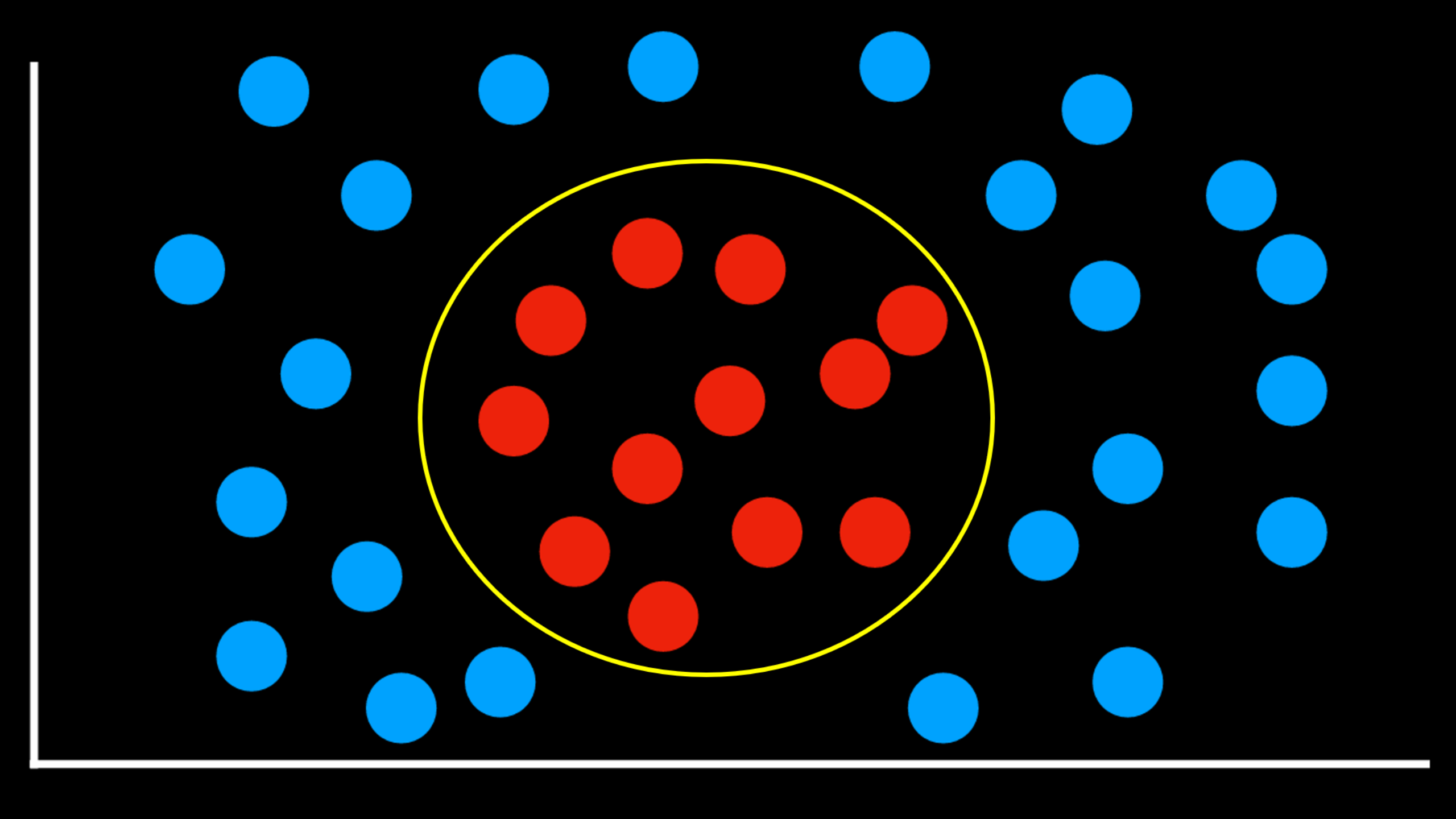




maximum margin separator

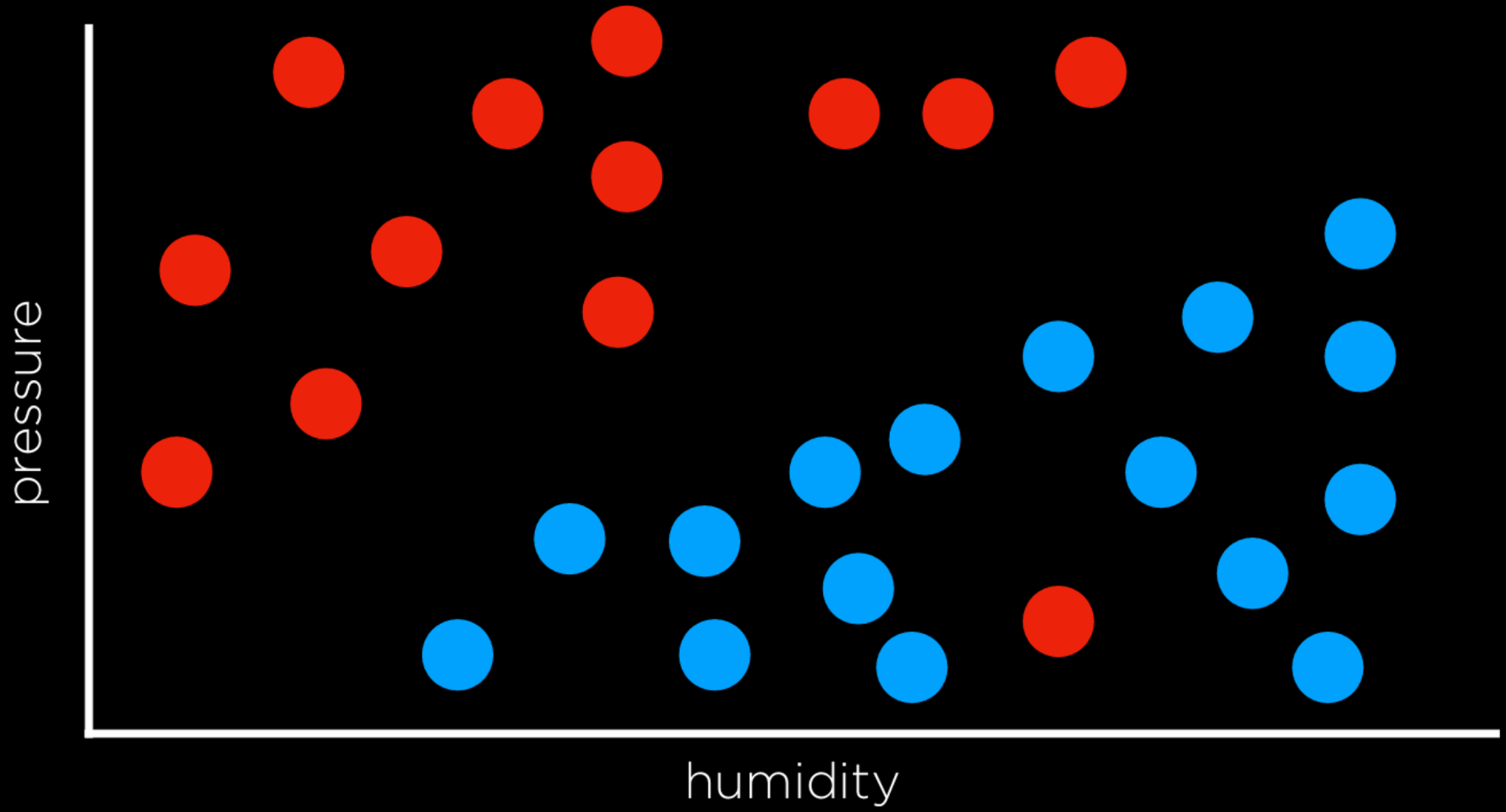
boundary that maximizes the distance between any of the data points

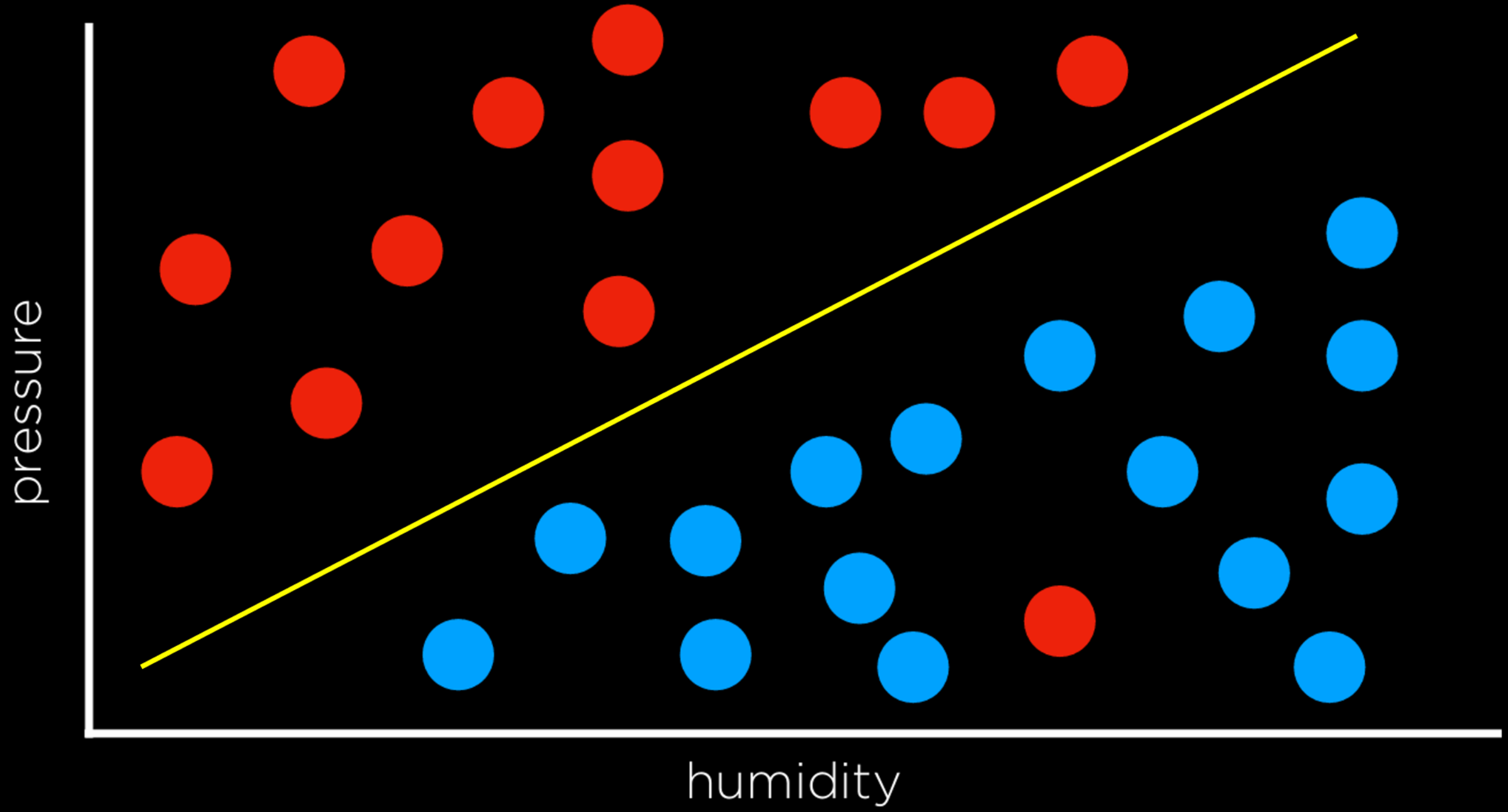


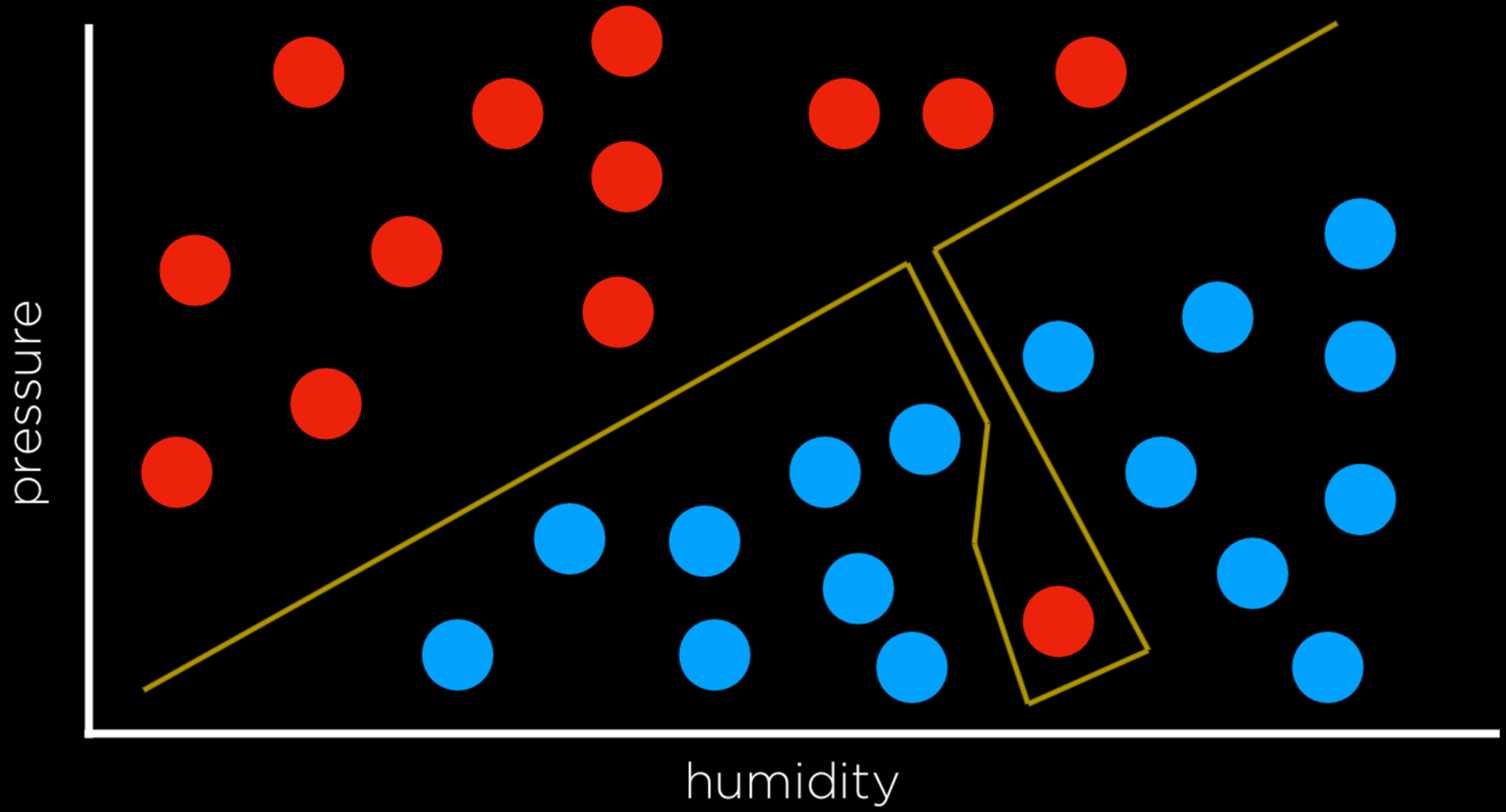


overfitting

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

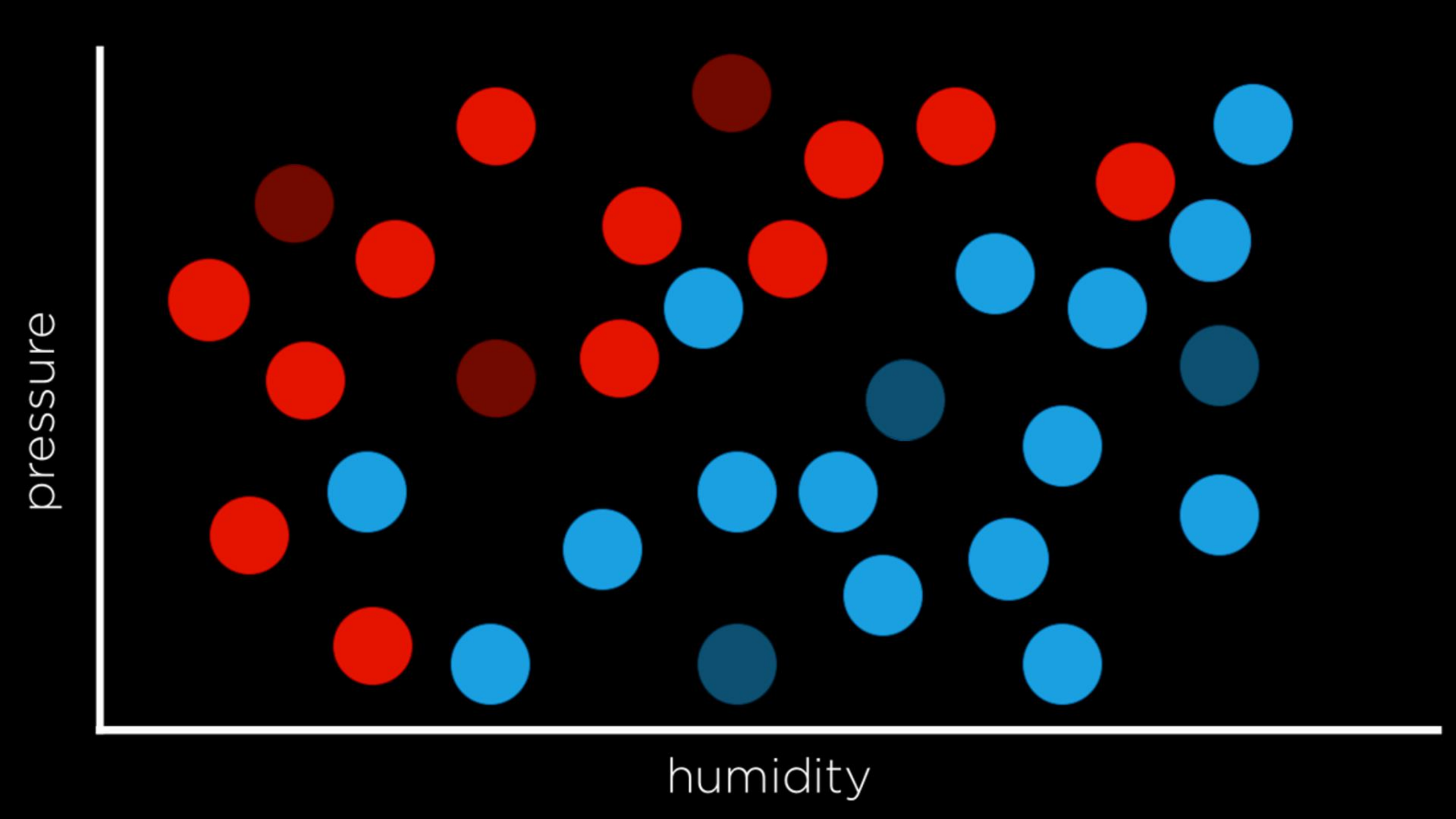
regularization

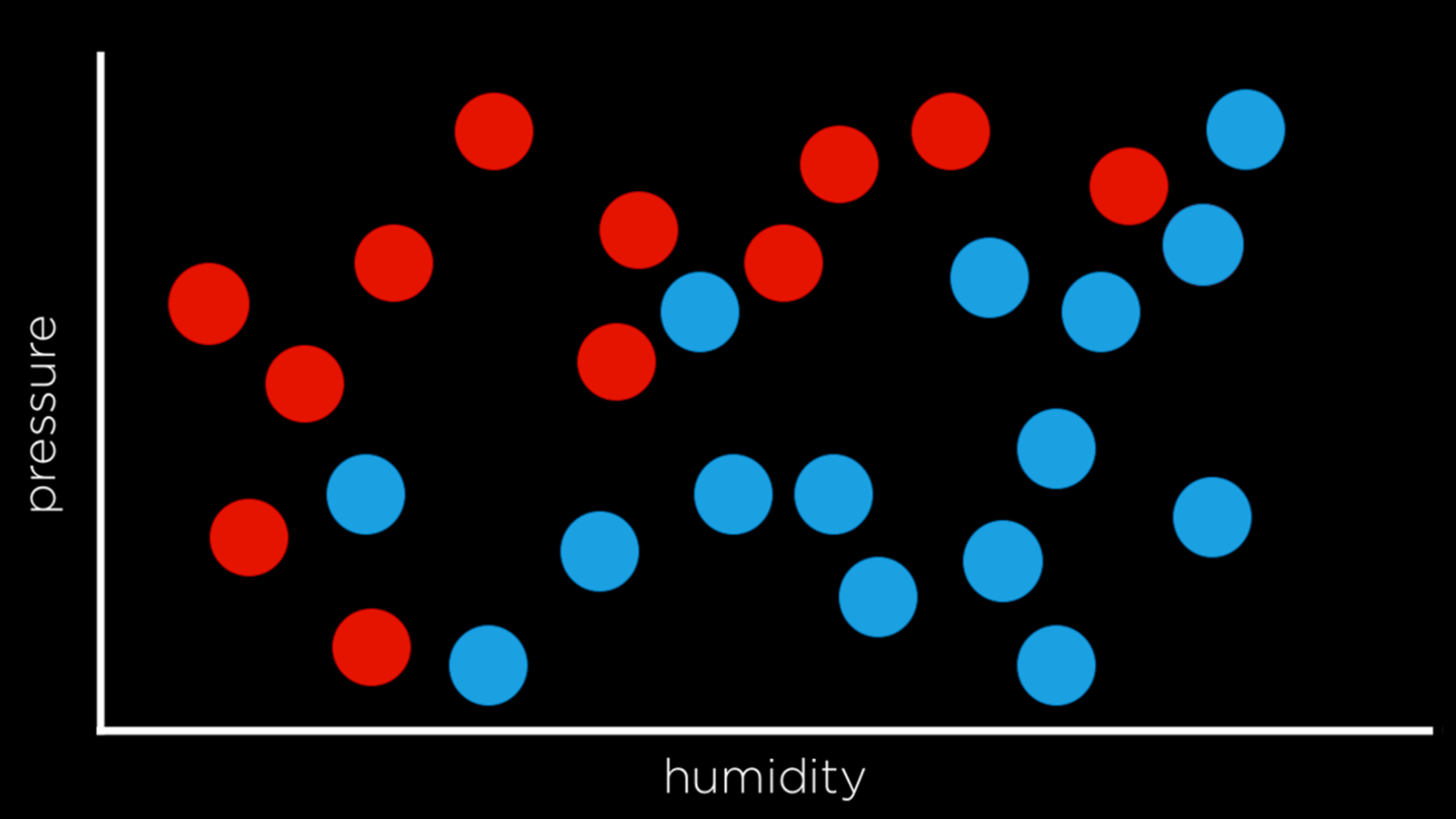
penalizing hypotheses that are more complex
to favor simpler, more general hypotheses

$$cost(\mathrm{h}) = loss(\mathrm{h}) + \lambda complexity(\mathrm{h})$$

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


pressure
humidity


pressure
humidity

Imbalanced Classification

Imbalanced classification refers to classification tasks where the number of examples in each class is unequally distributed. Typically, imbalanced classification tasks are binary classification tasks where the majority of examples in the training dataset belong to the normal class and a minority of examples belong to the abnormal class.

THE END