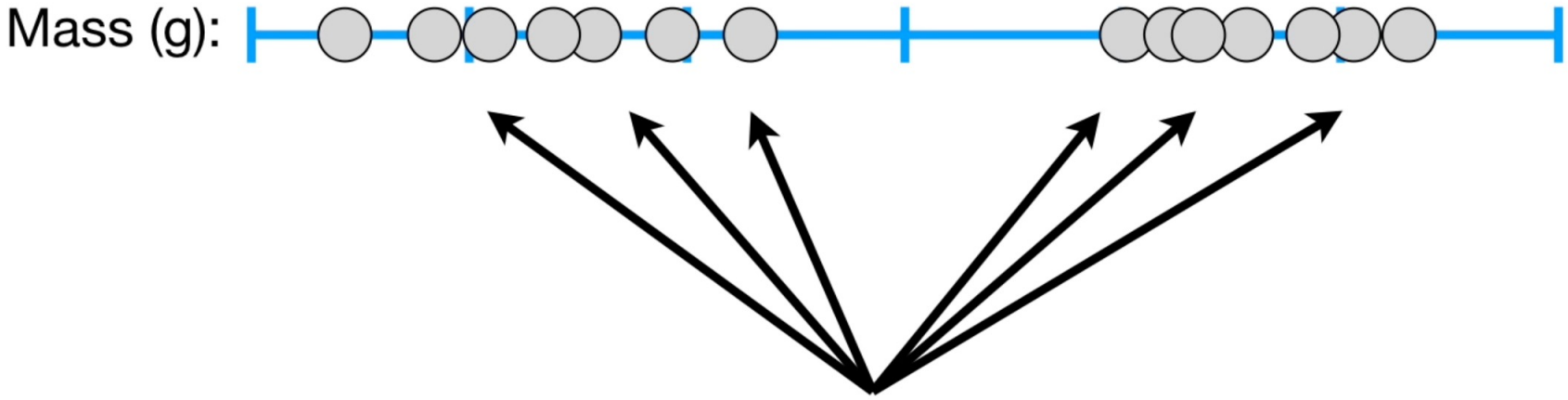
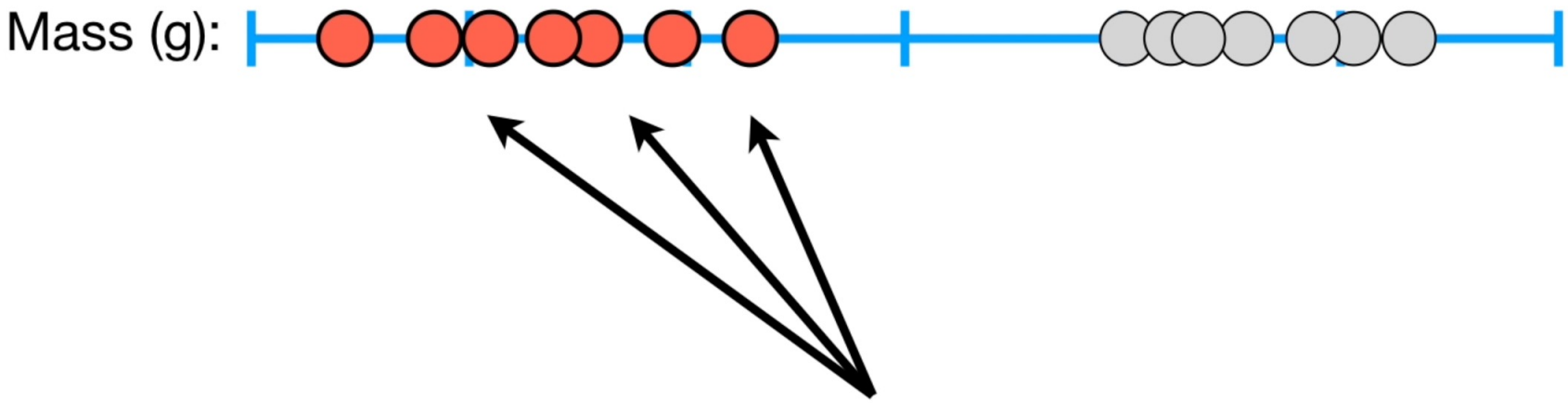
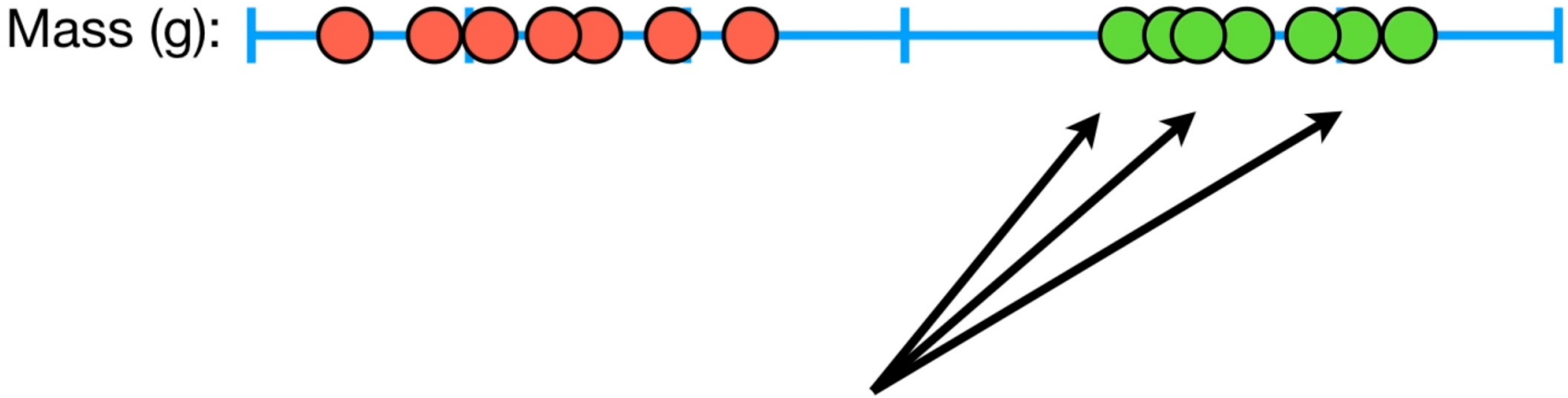
Support Vector Machines



Let’s start by imagining we measured the mass of a bunch of mice…

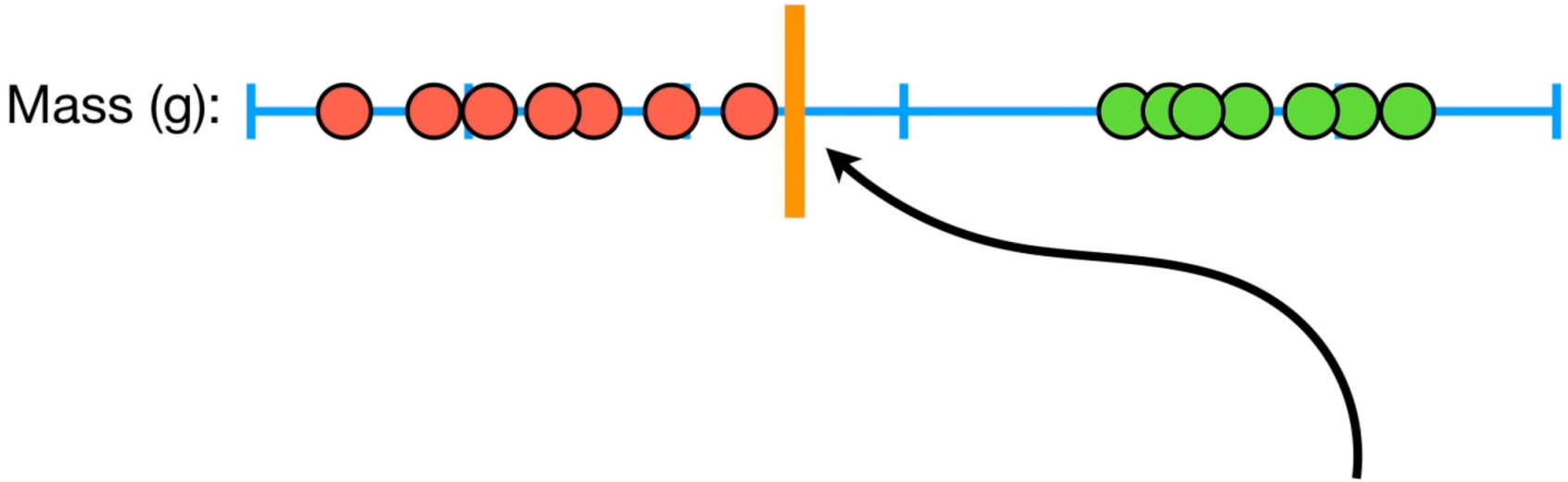


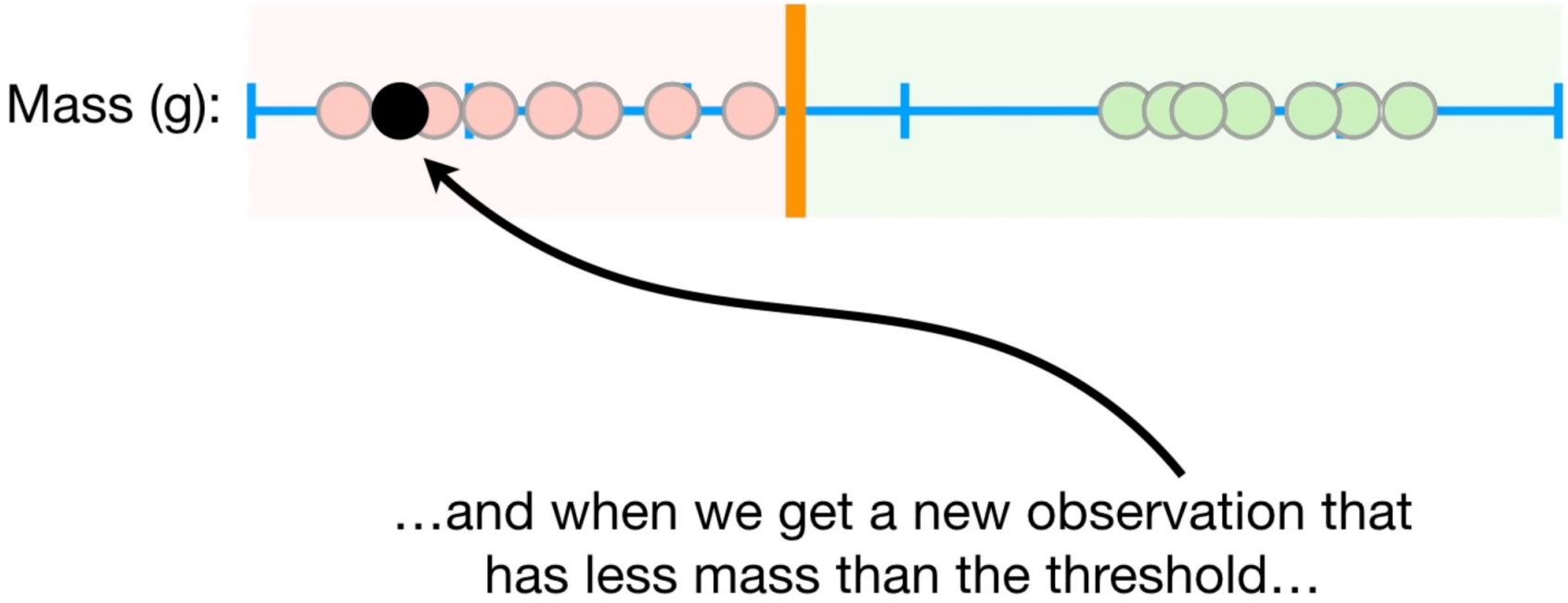
The **red dots** represent mice are ***not obese***…

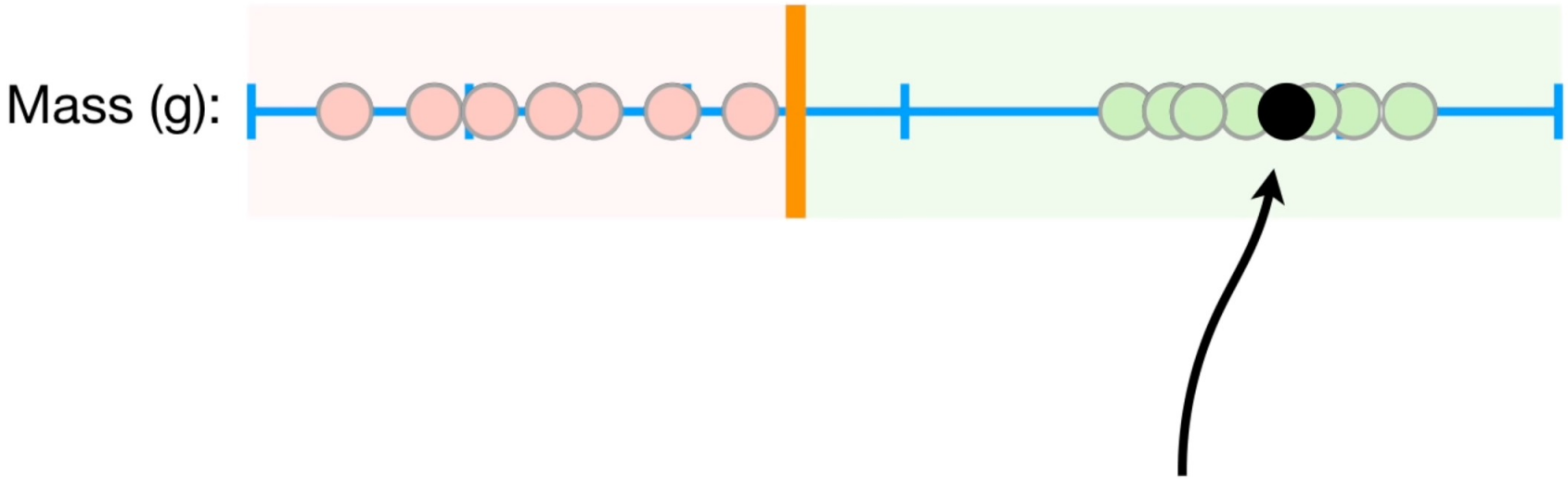


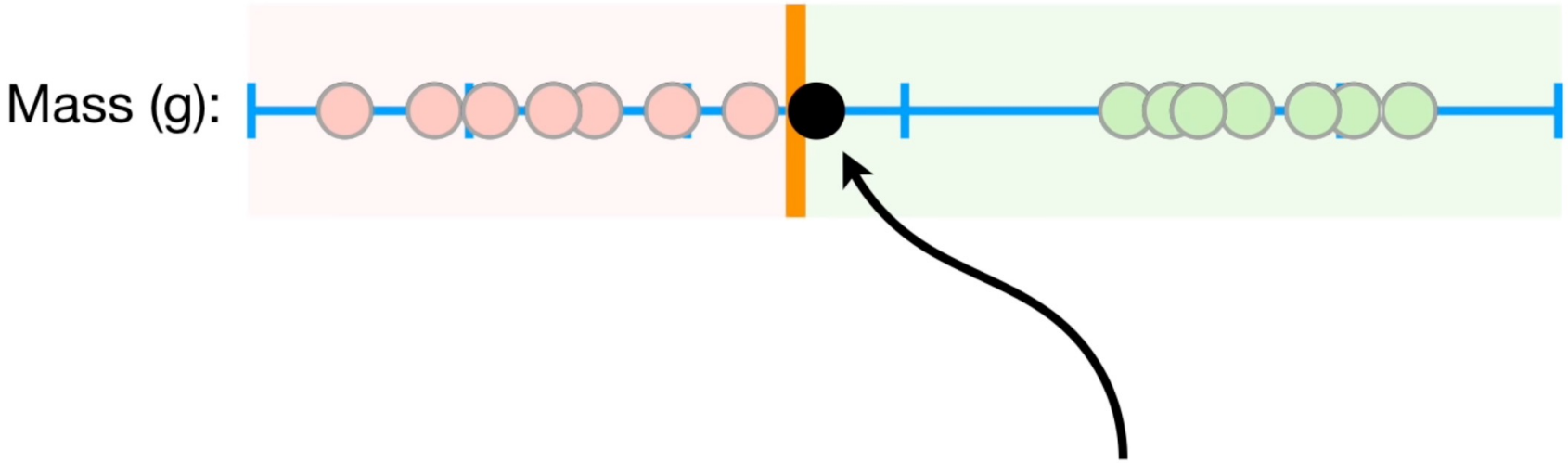
…and the **green dots** represent mice are ***obese***.

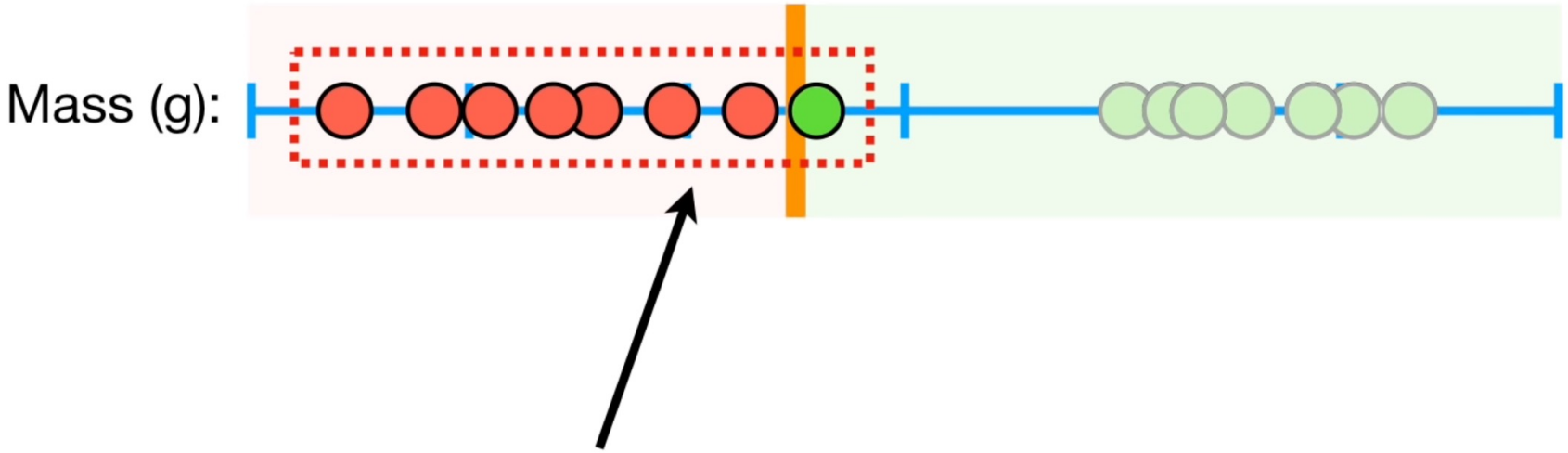
As a Naïve Approach, How we classify a new data point as obese or not obese?


Mass (g):
Based on these observations, we can pick a threshold…

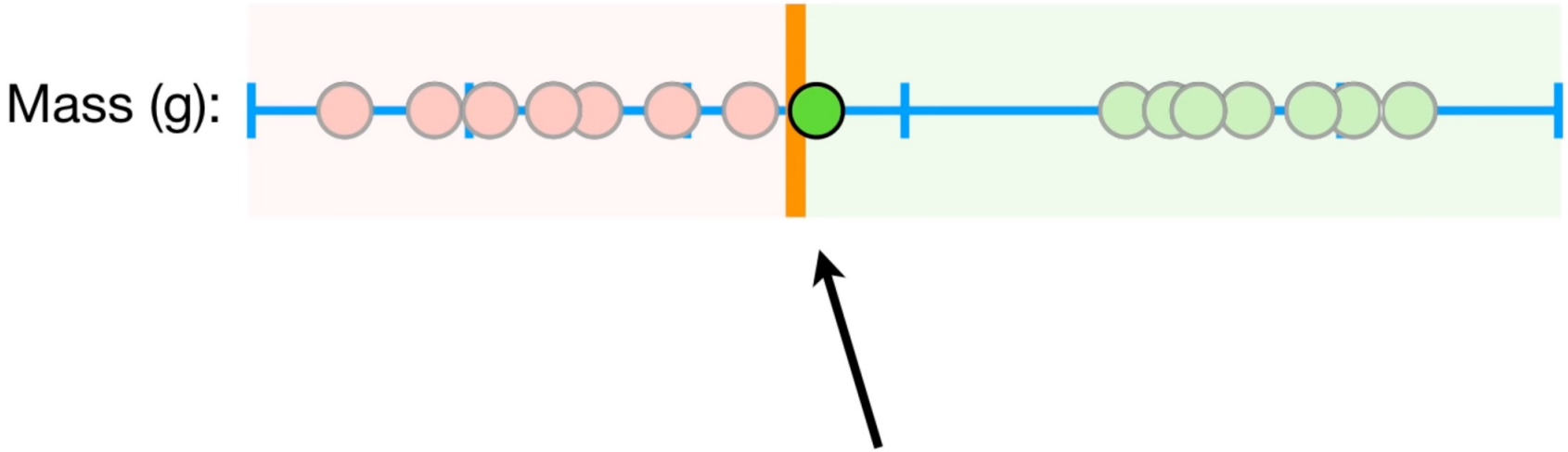

Mass (g):
…and when we get a new observation that
has less mass than the threshold…

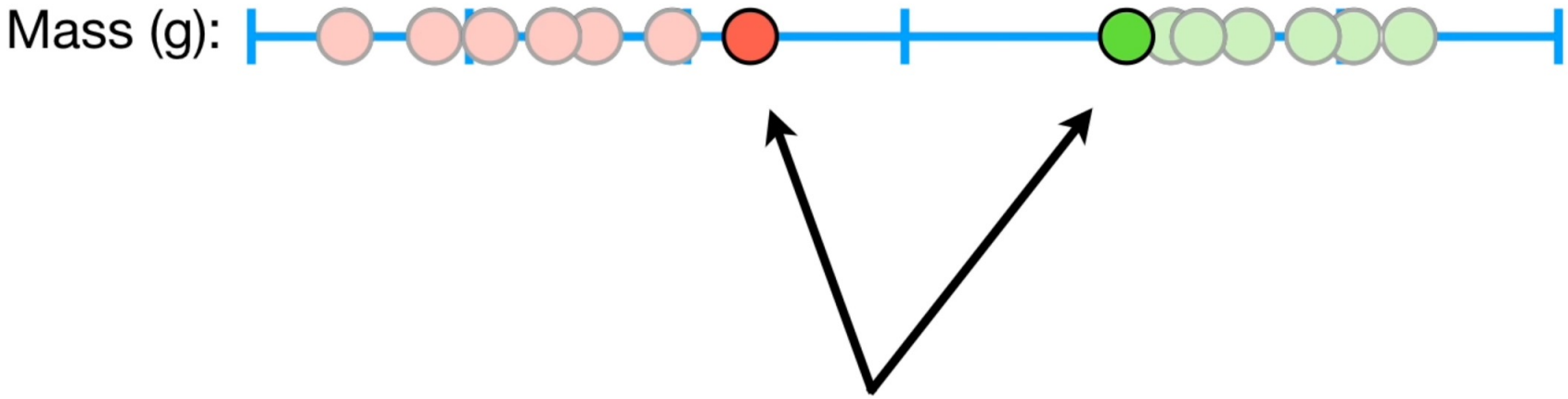

Mass (g):
And when we get a new observation with more mass than the threshold…

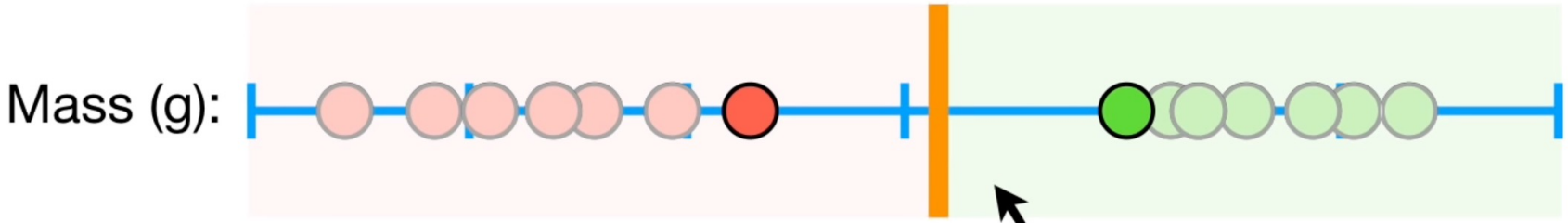

Mass (g):
However, what if get a new observation here?



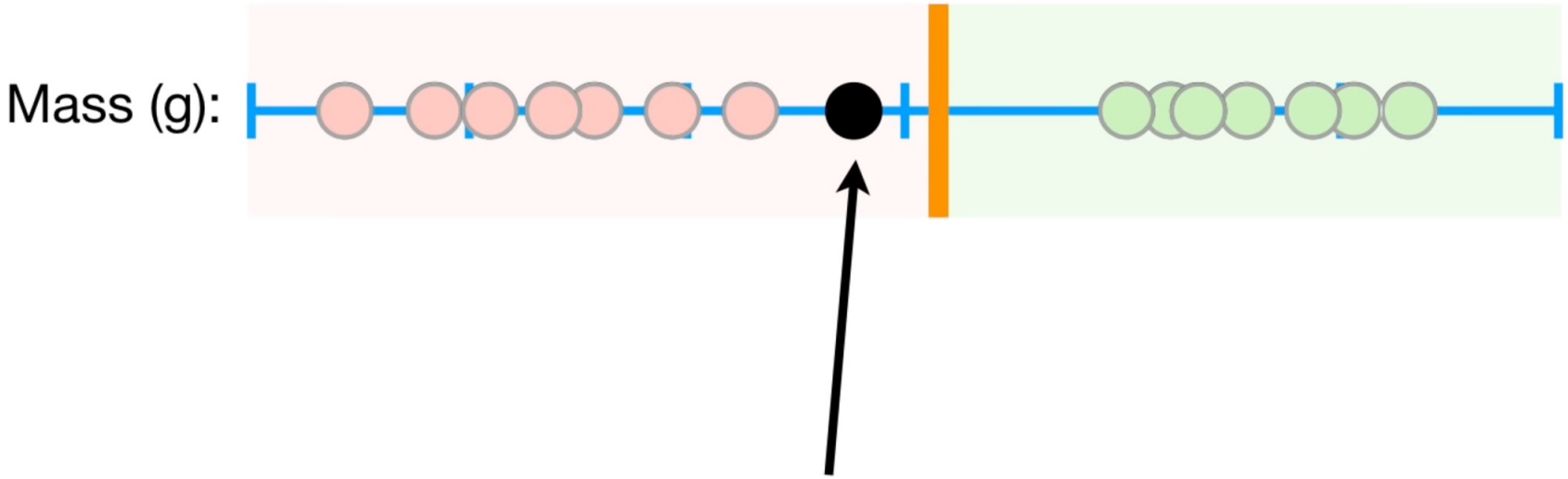
But that doesn't make sense, because it is much closer to the observations that are ***not obese***.

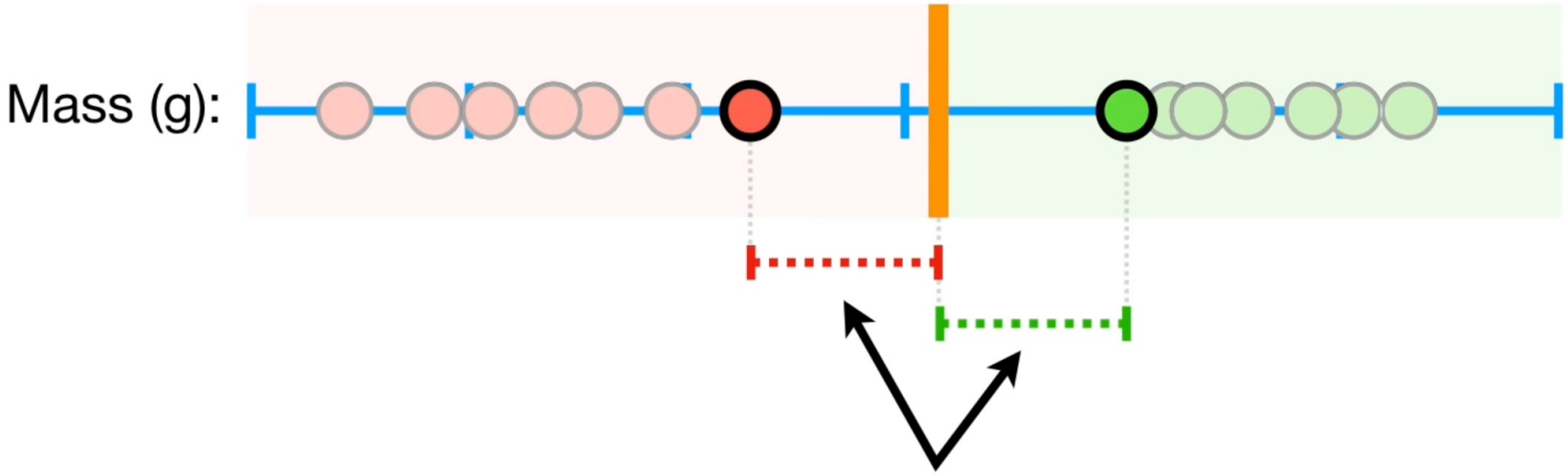

Mass (g):
So this threshold is pretty lame.


Mass (g):
…we can focus on the observations on
the edges of each cluster…

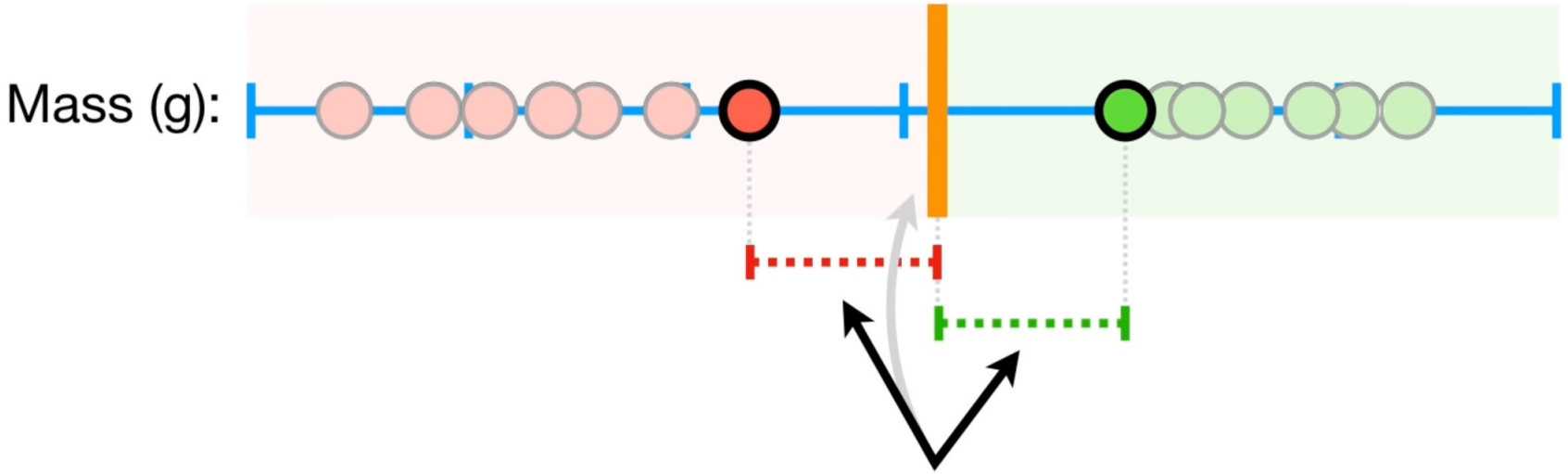


…and use the midpoint between them as the threshold.

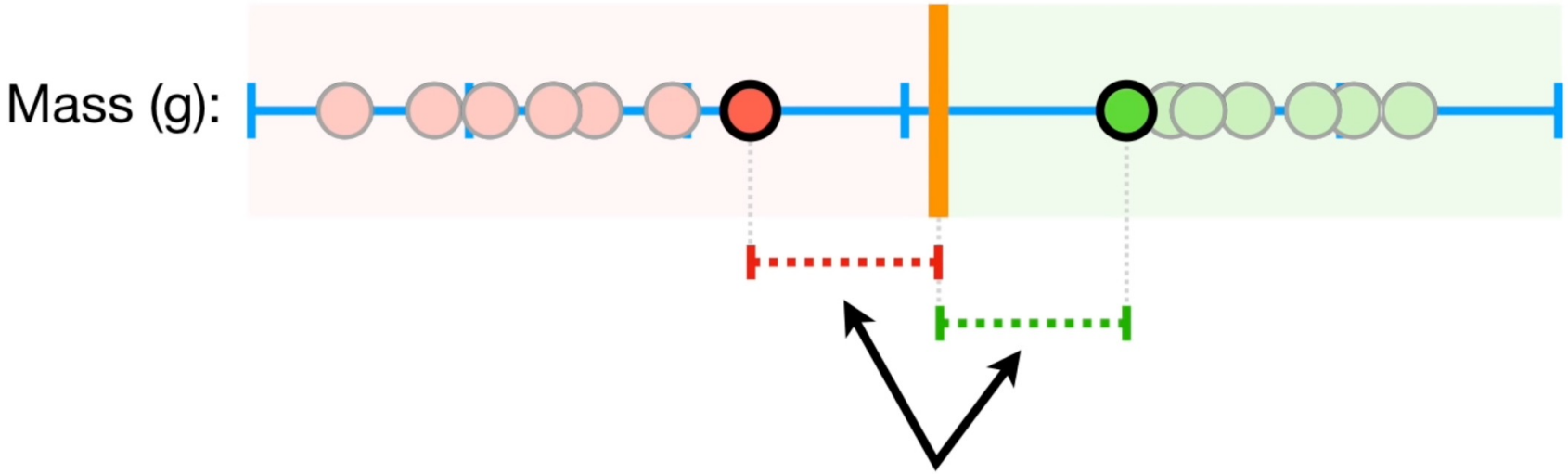

Mass (g):
Now, when a new observation falls
on the left side of the threshold…



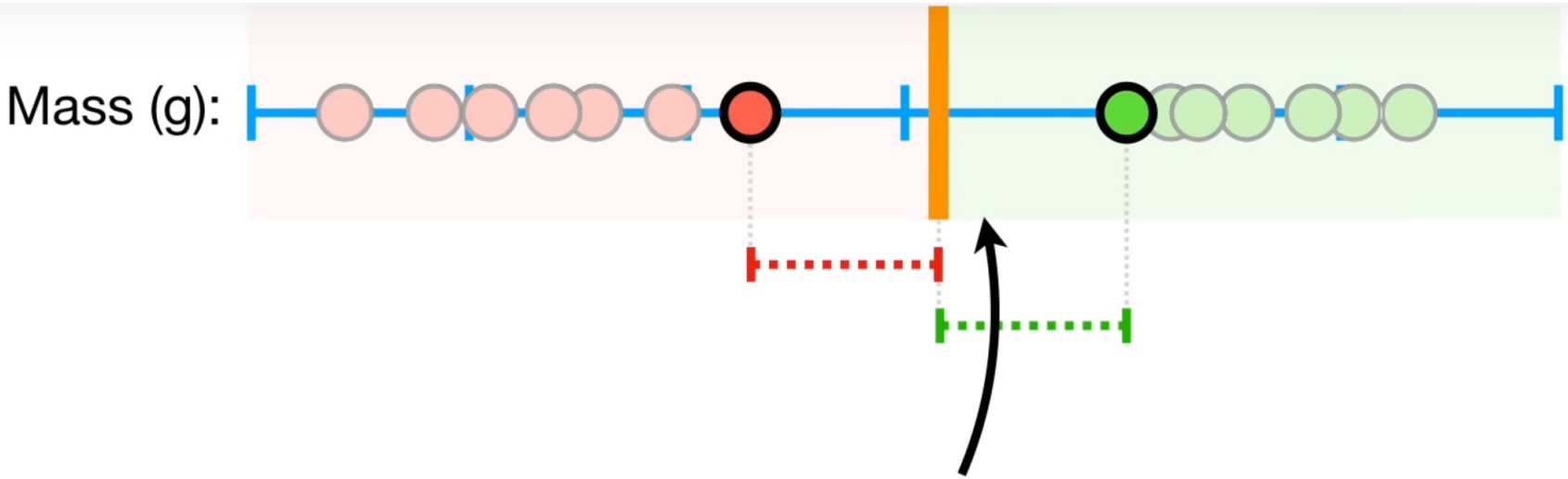
…the distances between the observations and the threshold are the same and both reflect the **margin**.



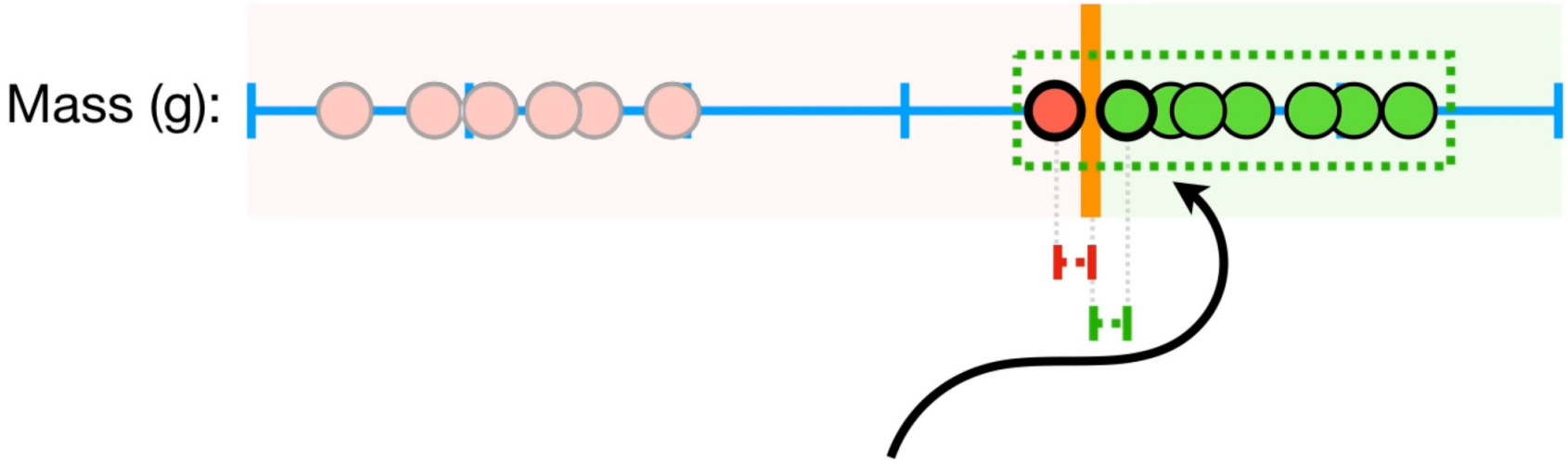
When the threshold is halfway between the two observations, the **margin** is as large as it can be.



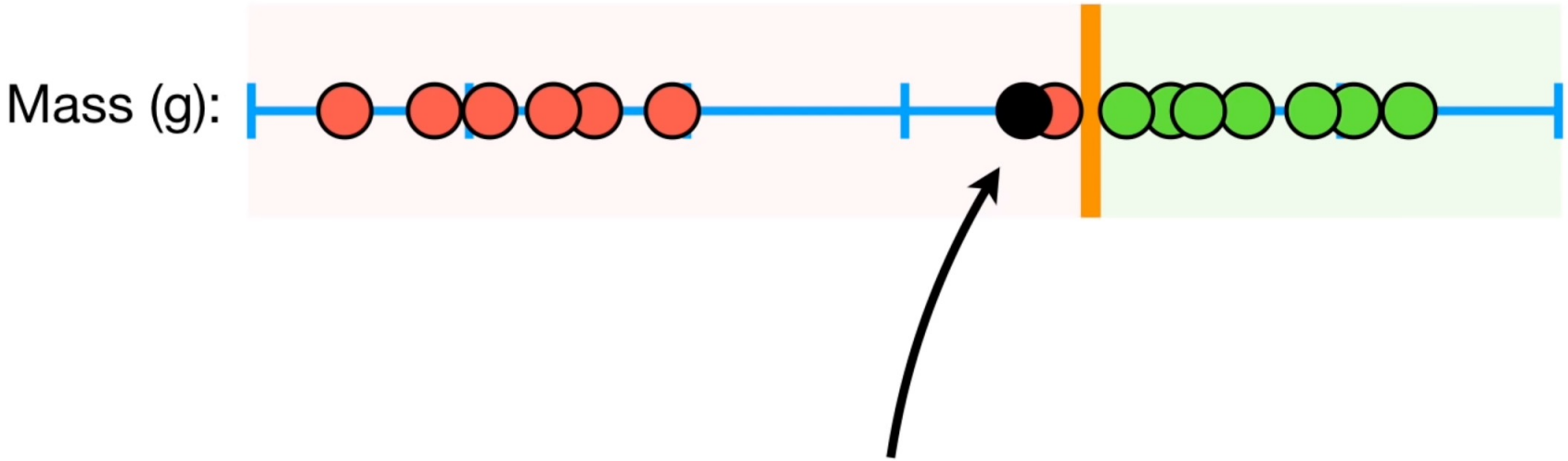
When we use the threshold that gives us the largest **margin** to make classifications…

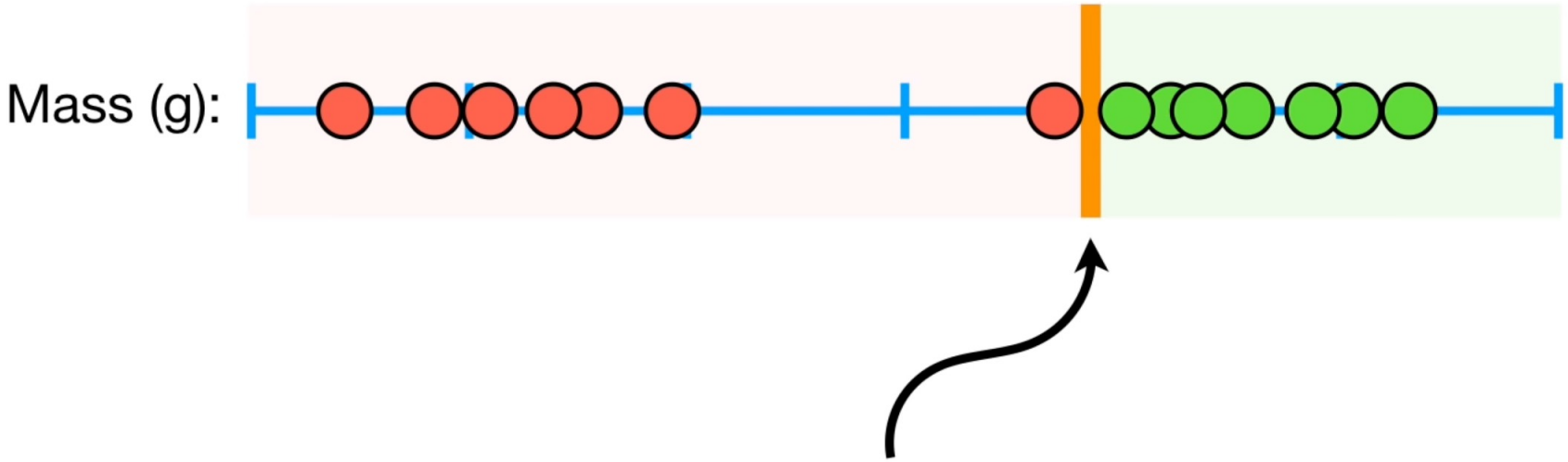


…we are using a
Maximal Margin Classifier.



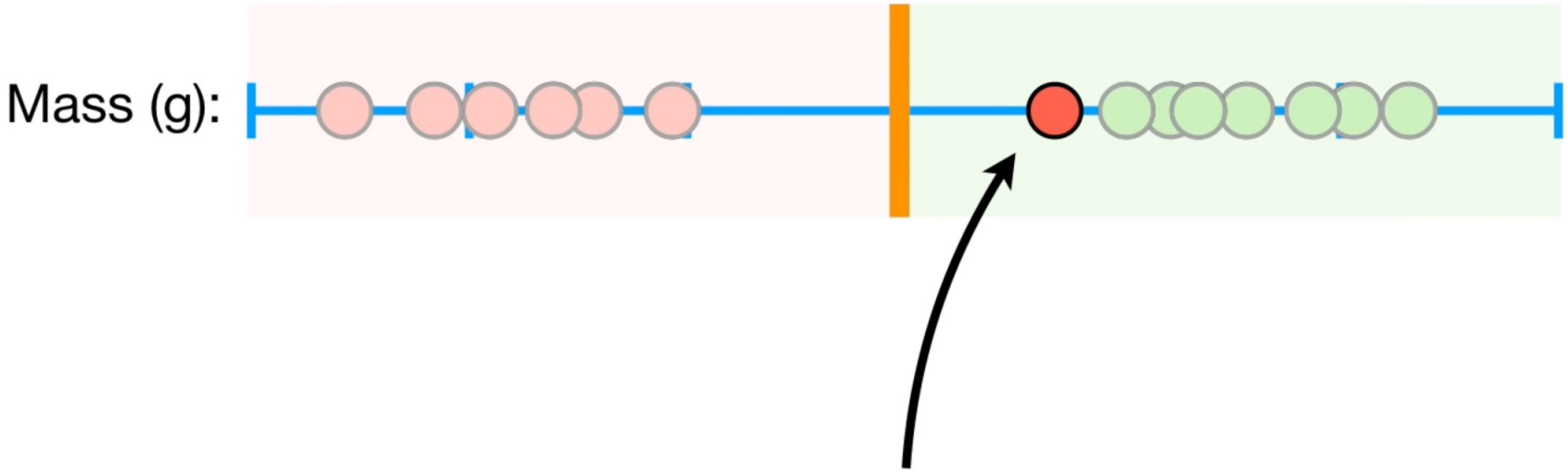
In this case, the **Maximum Margin Classifier** would be super close to the ***obese*** observations…

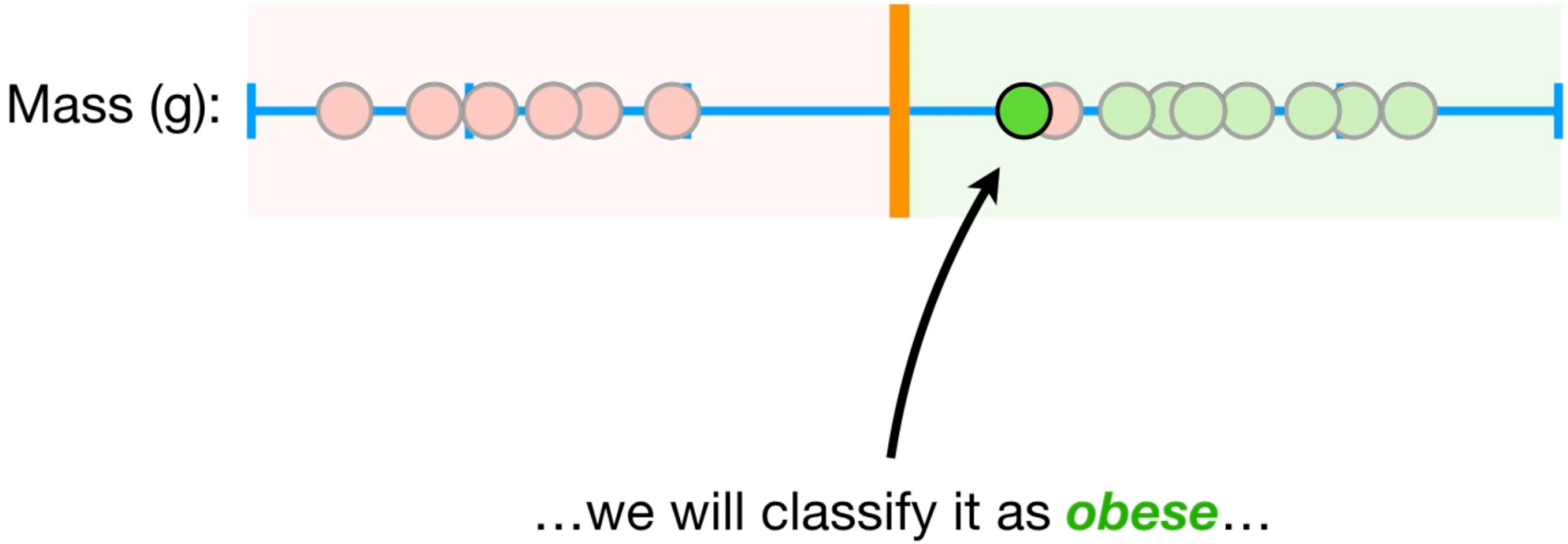

Mass (g):
Now, if we got this new observation…

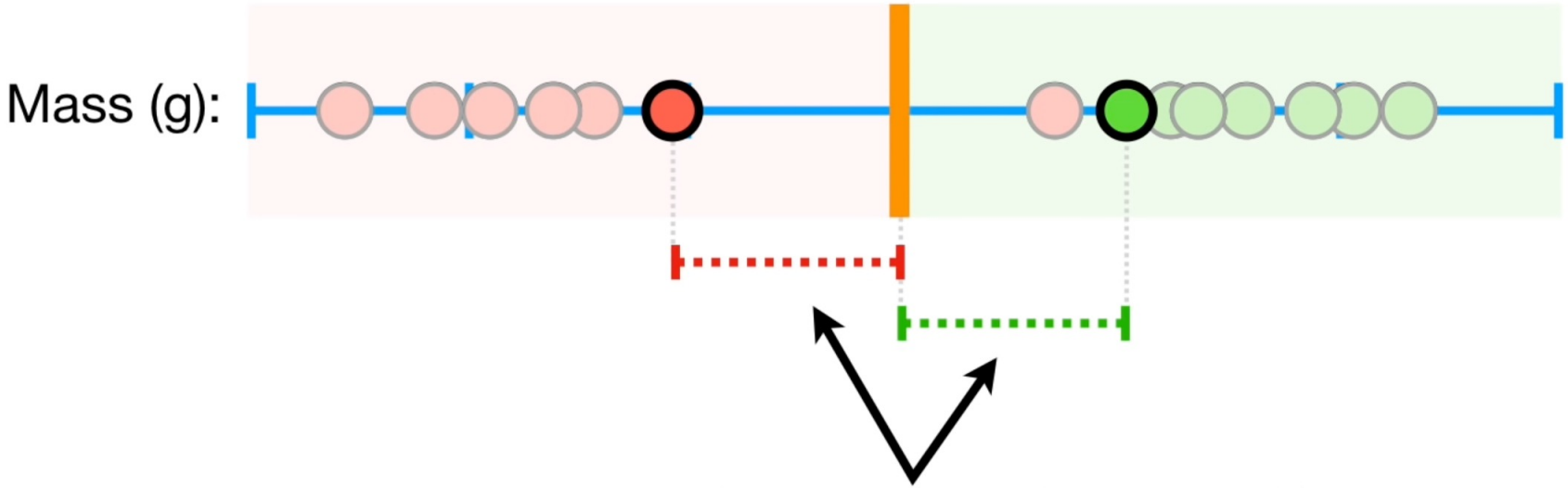


So **Maximal Margin Classifiers** are *super sensitive to outliers* in the training data and that makes them pretty lame.

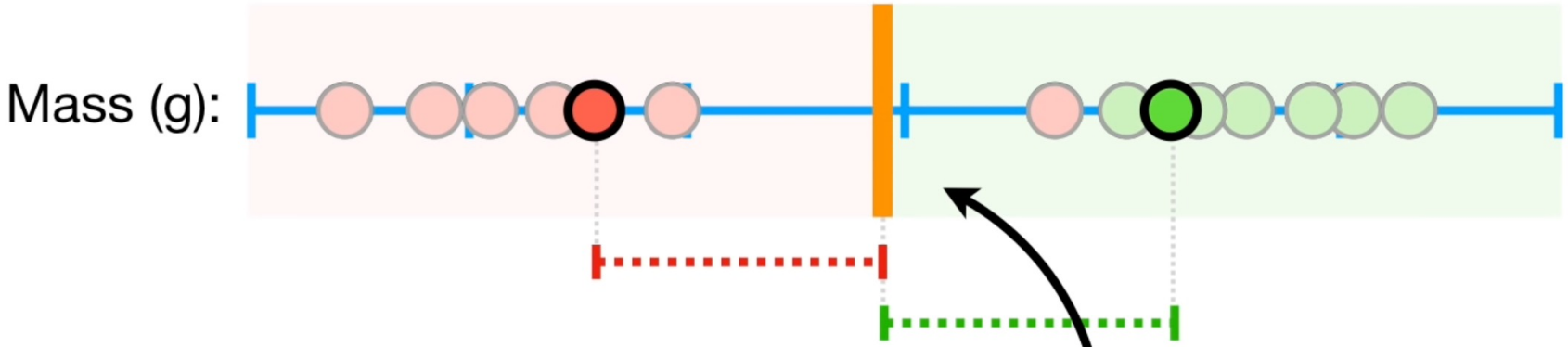
To make a threshold that is not so sensitive to outliers we must **allow misclassifications**.


Mass (g):
…then we will misclassify this observation.

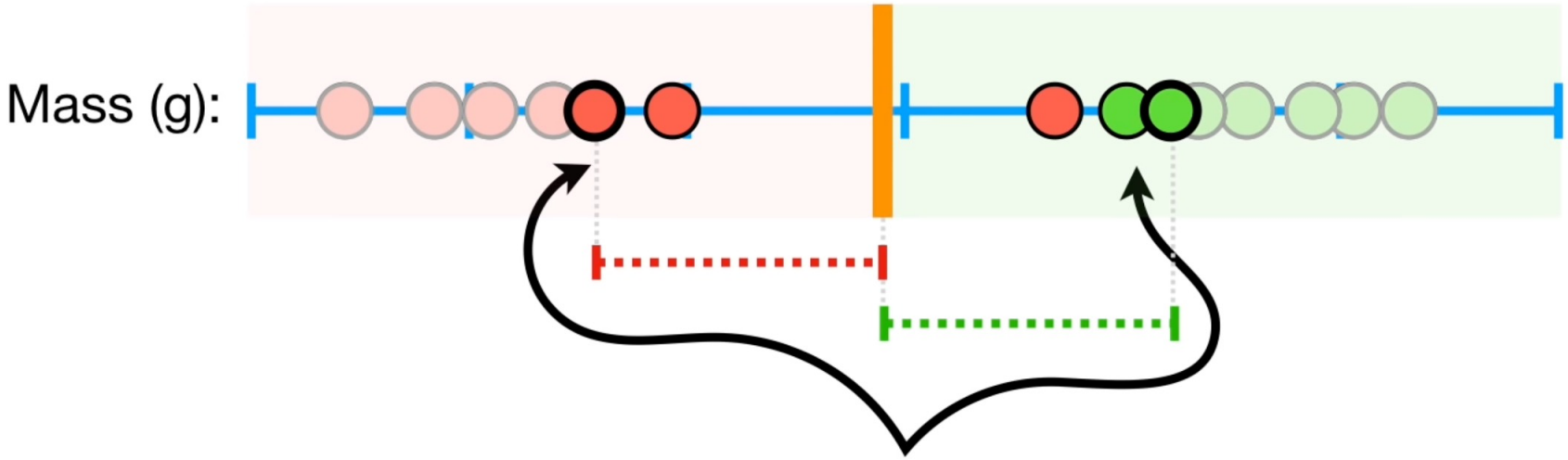

Mass (g):
…we will classify it as *obese*…



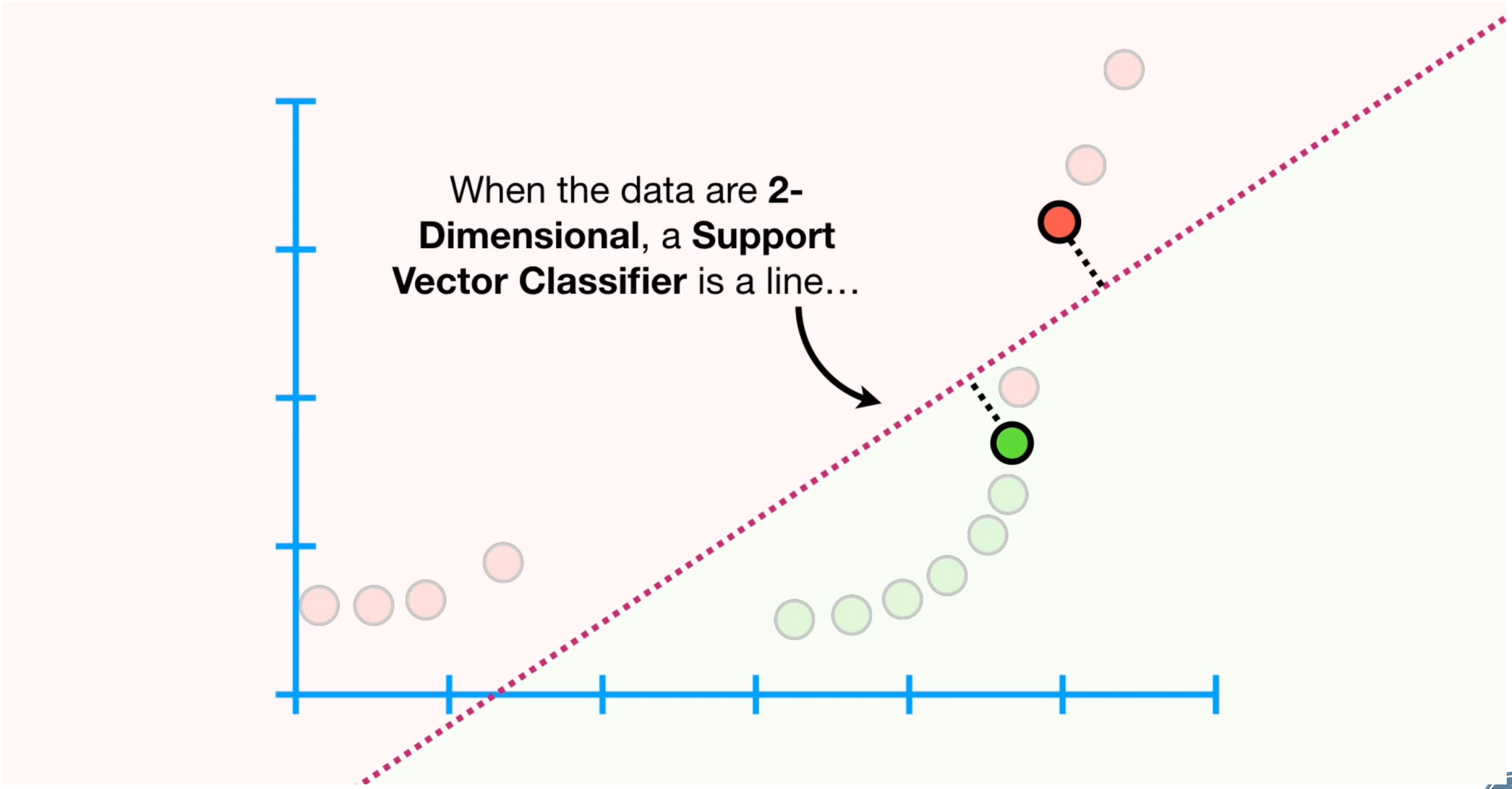
When we allow misclassifications, the distance between the observations and the threshold is called a **Soft Margin**.

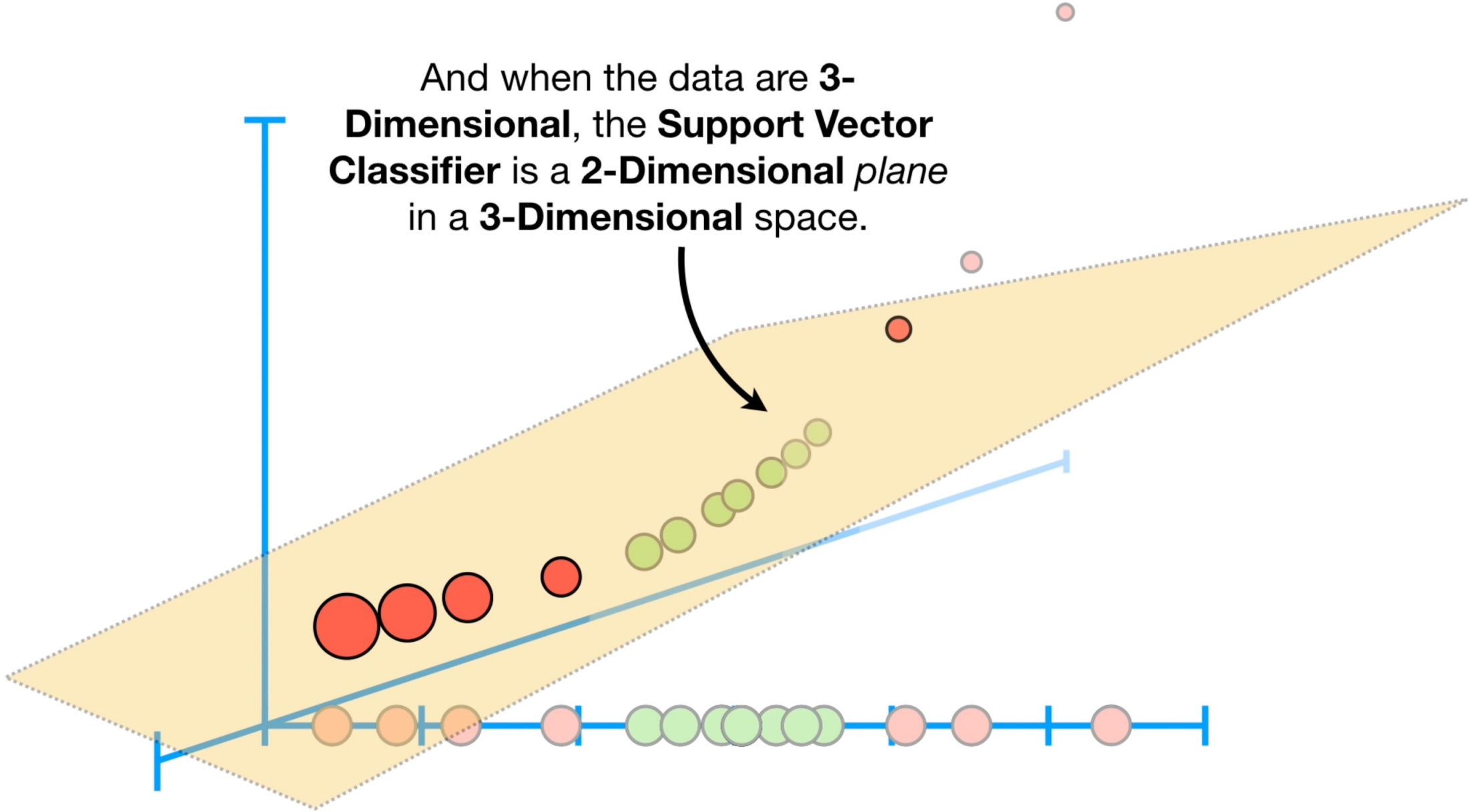


…then we are using a **Soft Margin Classifier** aka a **Support Vector Classifier** to classify observations.



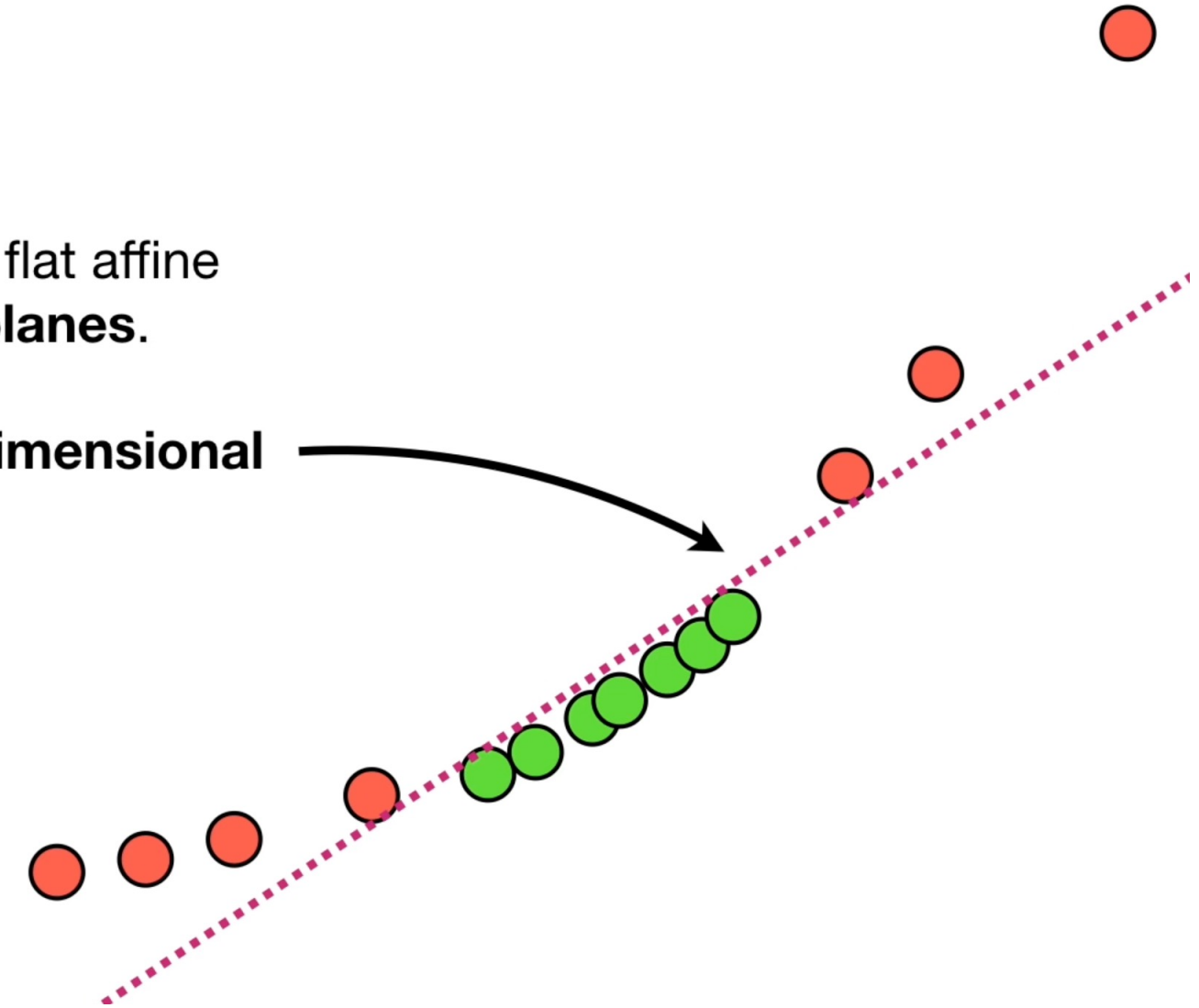
The name **Support Vector Classifier** comes from the fact that the observations on the edge *and within* the **Soft Margin** are called **Support Vectors**.


When the data are **2-Dimensional**, a **Support Vector Classifier** is a line…

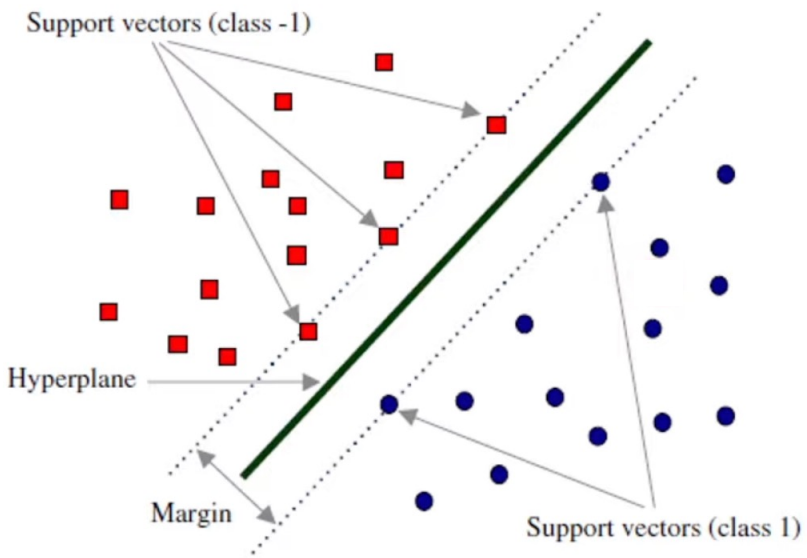

And when the data are **3-Dimensional**, the **Support Vector Classifier** is a **2-Dimensional** *plane* in a **3-Dimensional** space.

NOTE: Technically speaking, all flat affine subspaces are called **hyperplanes**.

So, technically speaking, this **1-Dimensional** *line* is a **hyperplane**…

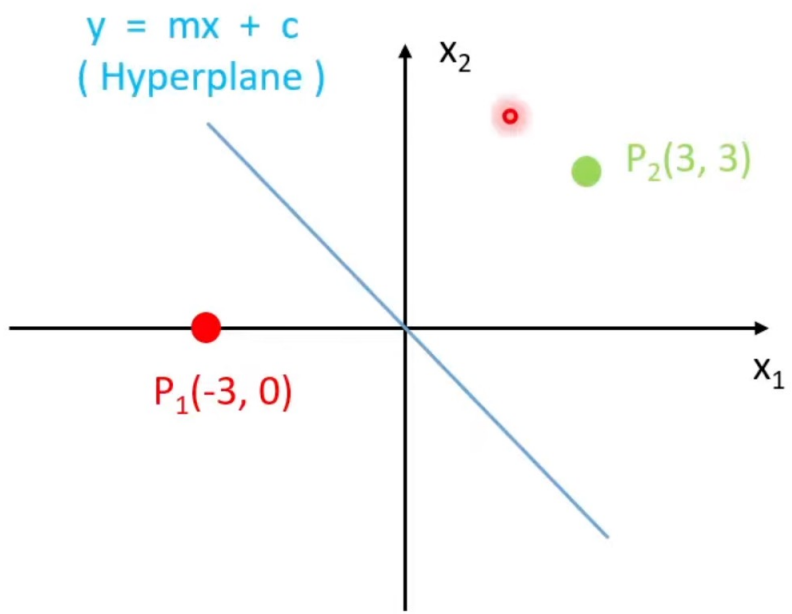


Support Vector Machine Classifier

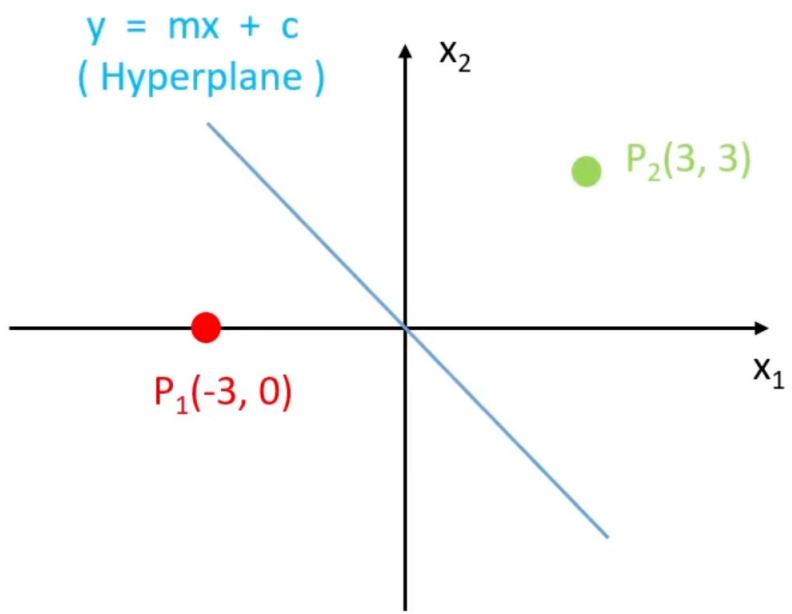


- Hyperplane
- Support Vectors
- Margin

Support Vector Machine Classifier



Support Vector Machine Classifier



Let slope, m = - 1

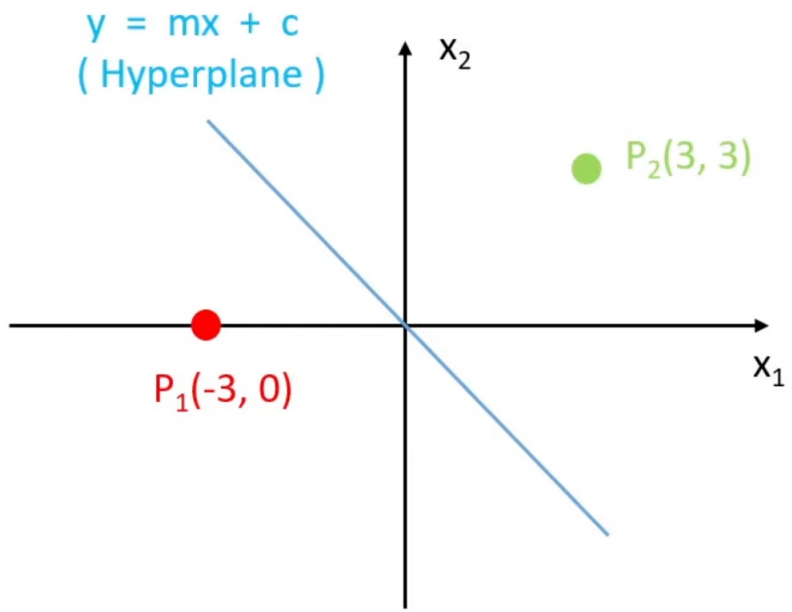
Intercept, c = 0

w --> parameters of the line
(m, c) = (-1, 0)

Support Vector Machine Classifier

$P_1(-3, 0)$

$$w^T x = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} -3 & 0 \end{bmatrix}$$



$P_2(3, 3)$

Siddhardhan

Let slope, m = - 1

Intercept, c = 0

w --> parameters of the line

(m, c) = (-1, 0)

Support Vector Machine Classifier

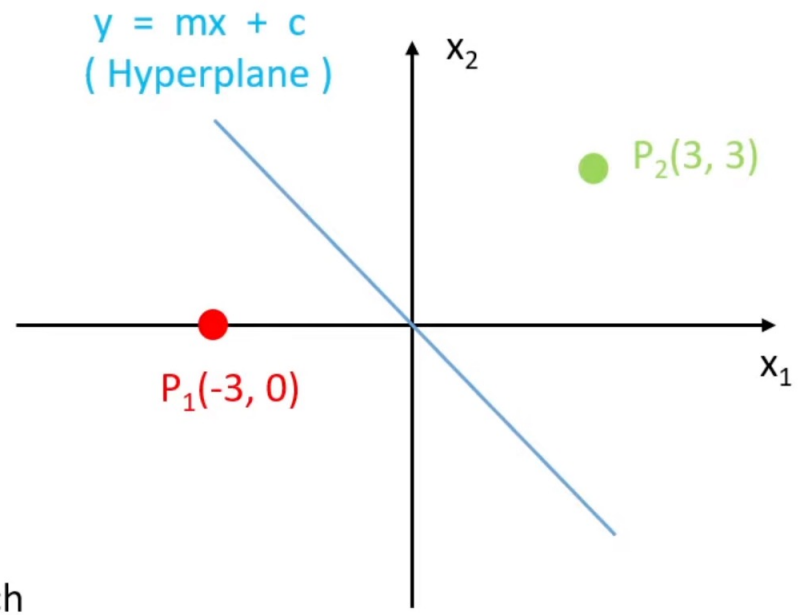
$P_1(-3, 0)$

$w^T x = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} -3 & 0 \end{bmatrix}$

$w^T x = 3$

(Positive)

Inference: For all the points which lie in the left side of the hyperplane, $w^T x$ value will be Positive



Let slope, m = - 1

Intercept, c = 0

w --> parameters of the line

(m, c) = (-1, 0)

$P_2(3, 3)$

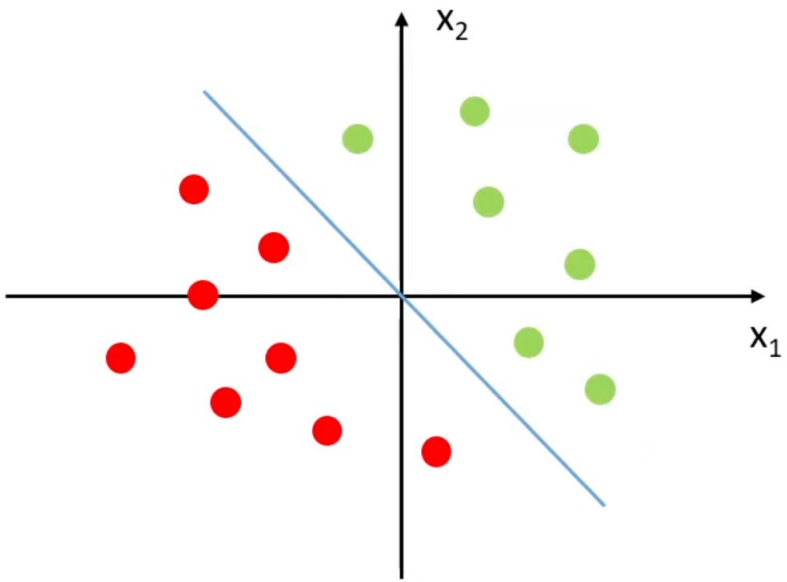
$w^T x = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} 3 & 3 \end{bmatrix}$

$w^T x = -3$

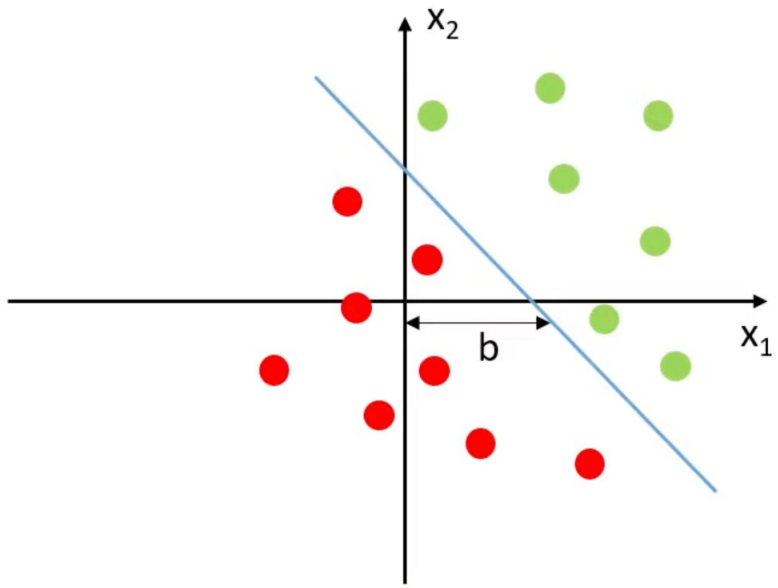
(Negative)

Inference: For all the points which lie in the right side of the hyperplane, $w^T x$ value will be Negative

Support Vector Machine Classifier



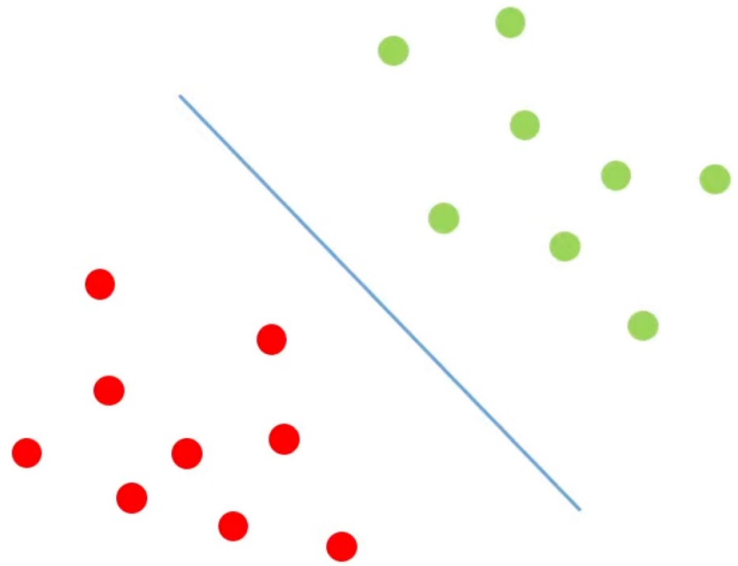
$w^T x = \text{Label}$



$w^T x + b = \text{Label}$

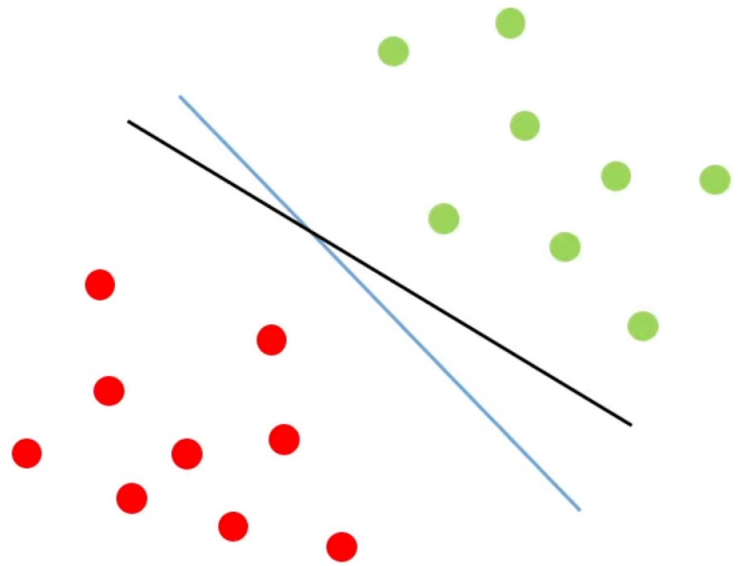
Support Vector Machine Classifier

Which is the best Hyperplane?



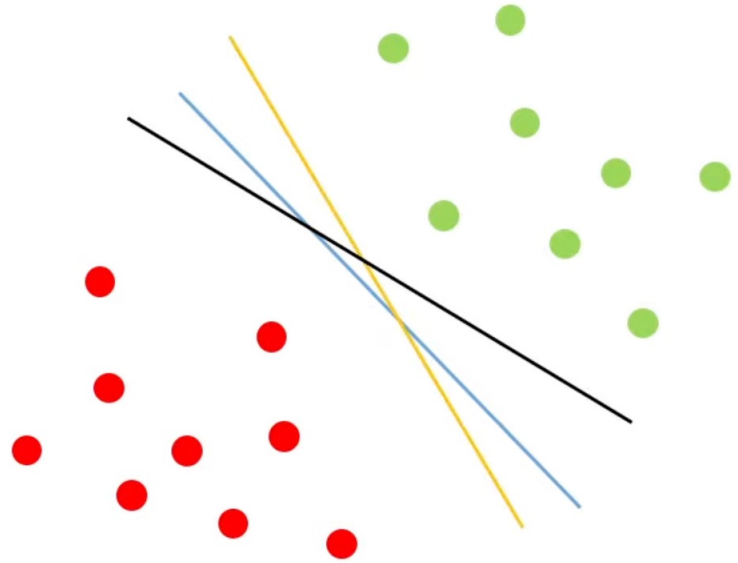
Support Vector Machine Classifier

Which is the best Hyperplane?



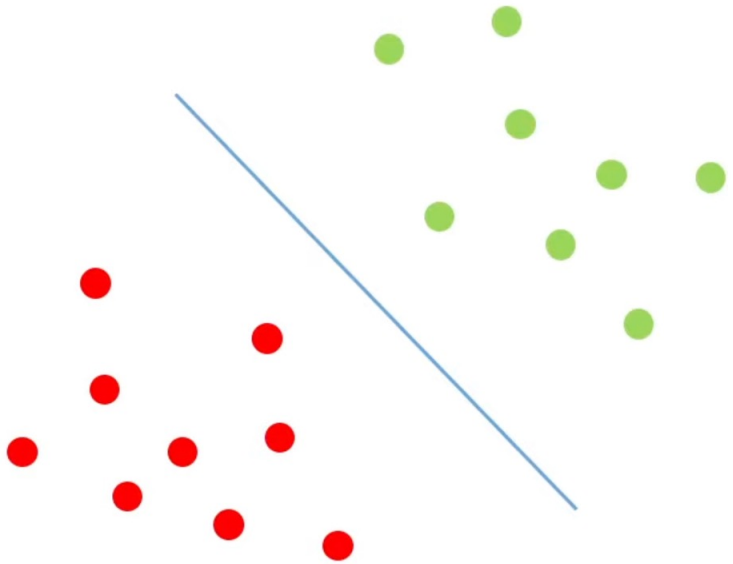
Support Vector Machine Classifier

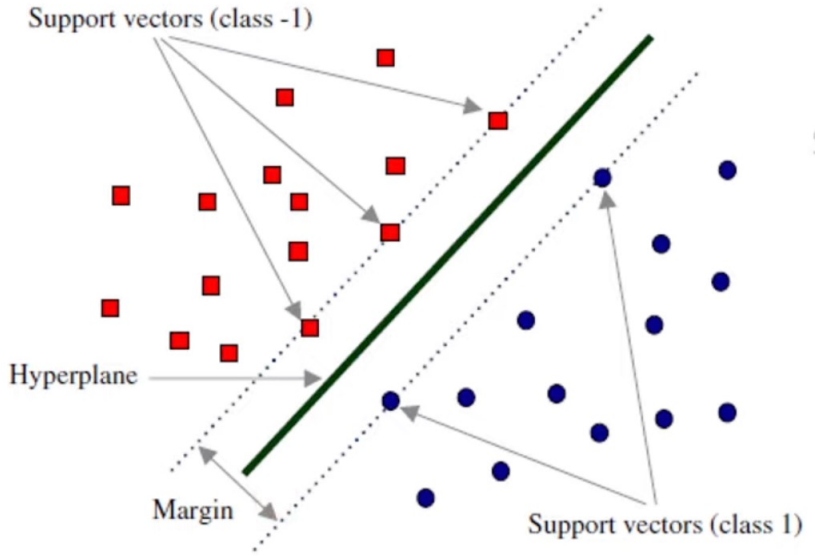
Which is the best Hyperplane?



Support Vector Machine Classifier

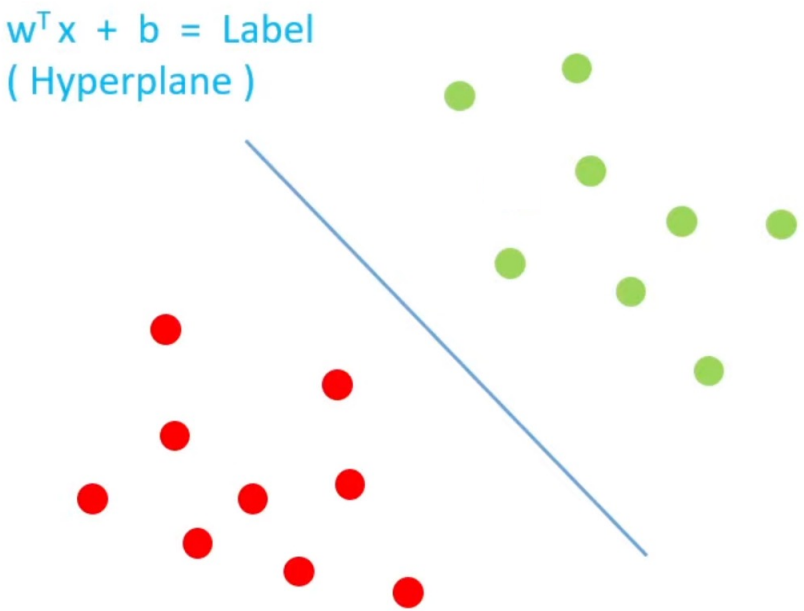
Which is the best Hyperplane?





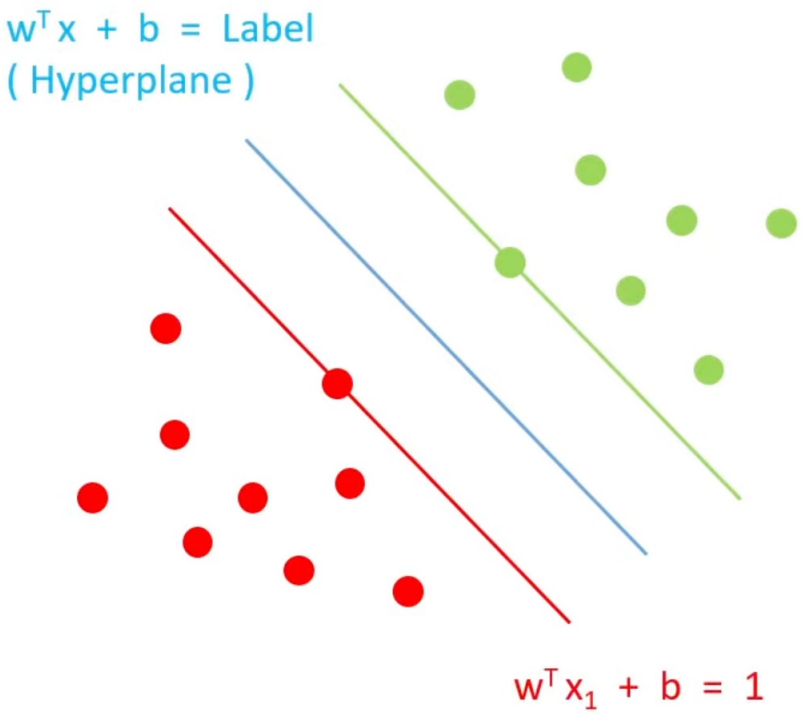
Support Vector Machine Classifier

Optimization for Maximum margin:



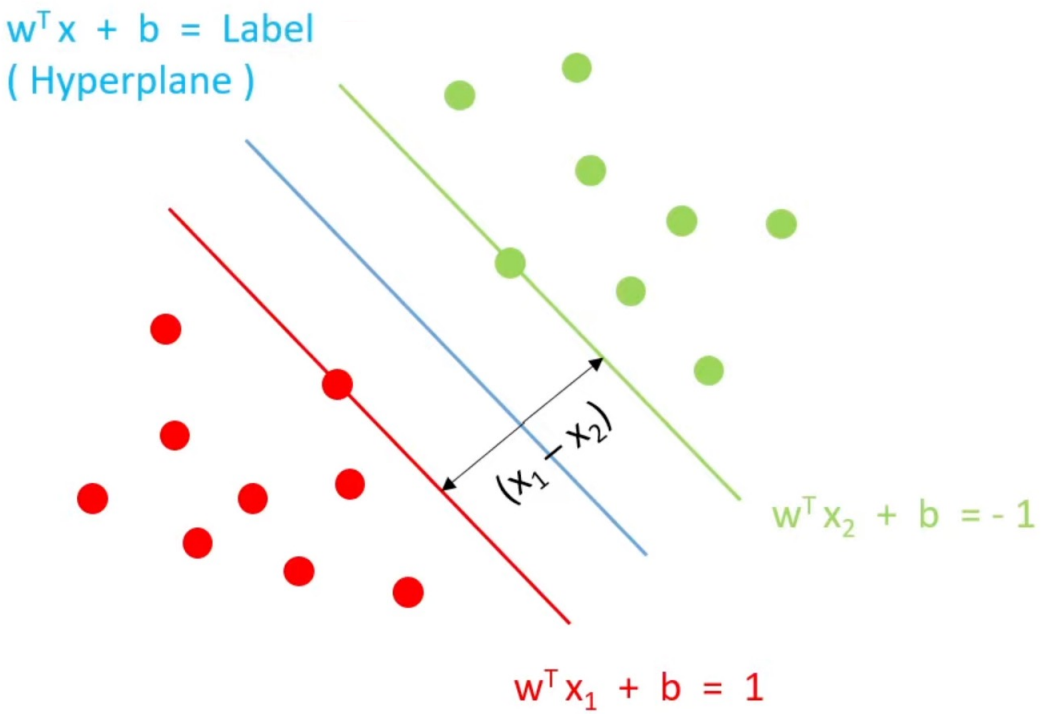
Support Vector Machine Classifier

Optimization for Maximum margin:



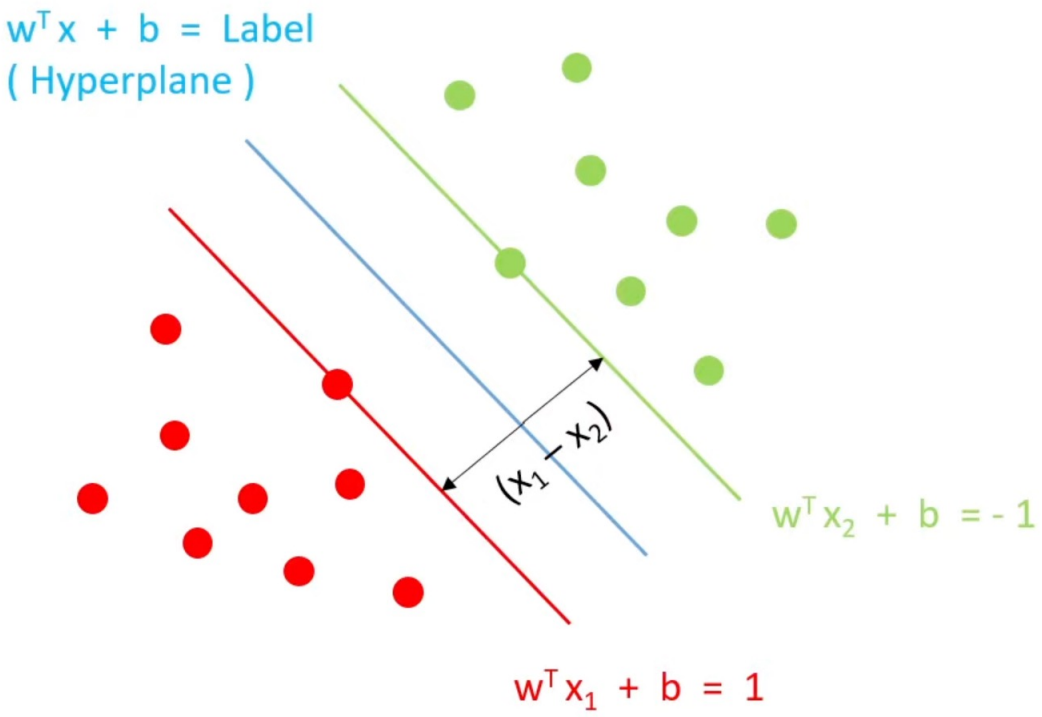
Support Vector Machine Classifier

Optimization for Maximum margin:



Support Vector Machine Classifier

Optimization for Maximum margin:



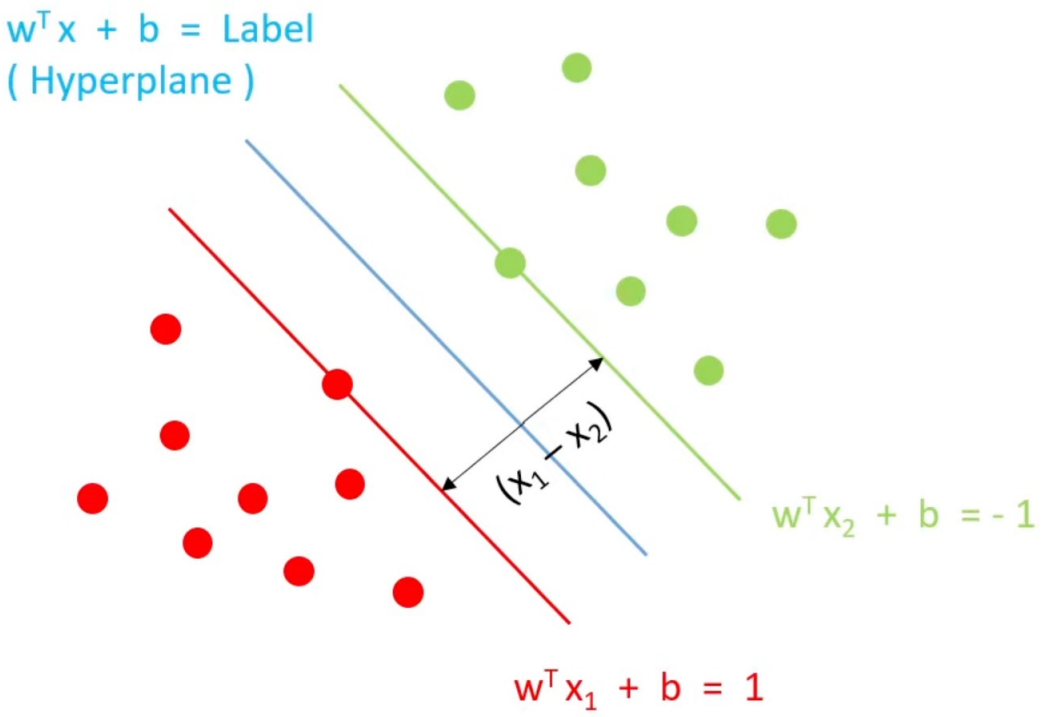
$$w^T x_1 + b = 1$$

$$(-)\quad w^T x_2 + b = -1$$

$$w^T (x_1 - x_2) = 2$$

Support Vector Machine Classifier

Optimization for Maximum margin:



$$w^T x_1 + b = 1$$

$$(-)\quad w^T x_2 + b = -1$$

$$w^T (x_1 - x_2) = 2$$

Divide by

(magnitude of the vector)

$$\frac{w^T}{||w||}(x_1 - x_2) = \frac{2}{||w||}$$

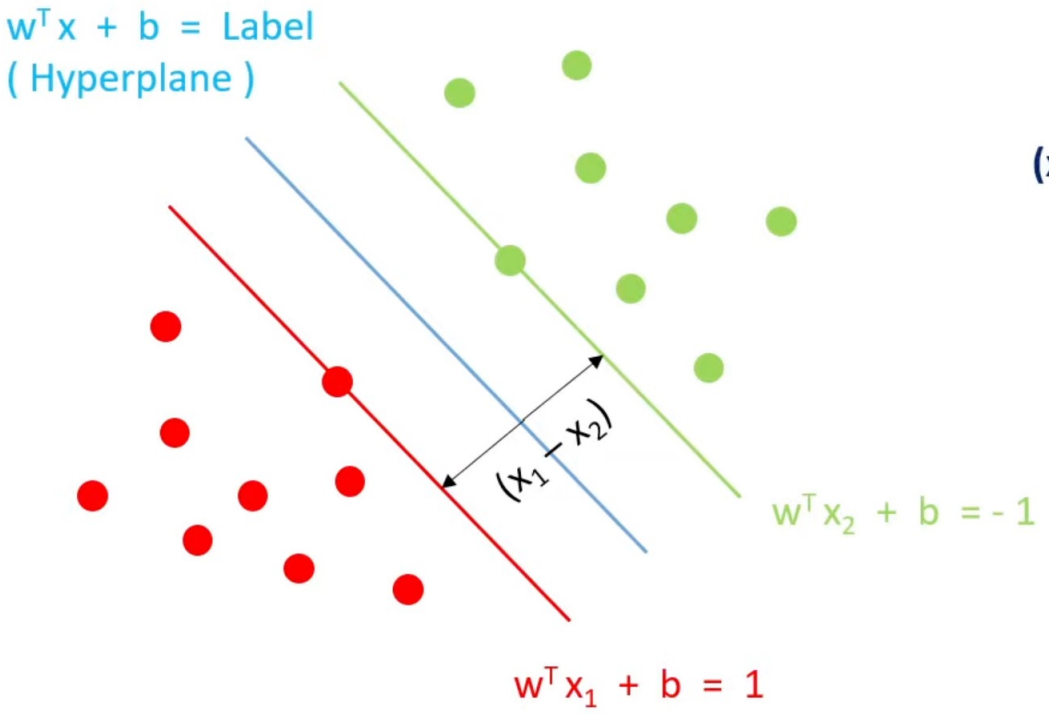
$$\mathbf{(x_1 - x_2) = \frac{2}{||w||}}$$ (margin)

Support Vector Machine Classifier

Optimization for Maximum margin:

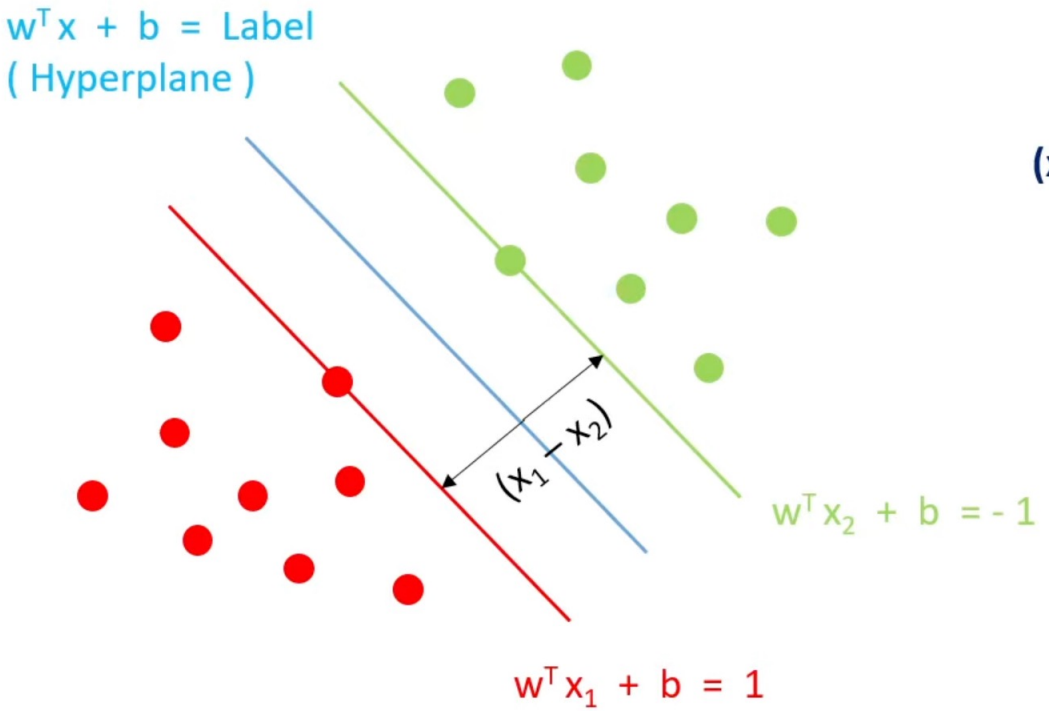
$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases} \quad \text{(Label)}$$

$$(x_1 - x_2) = \frac{2}{||\, w \,||} \quad \text{(margin)}$$



Support Vector Machine Classifier

Optimization for Maximum margin:



$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases} \quad \text{(Label)}$$

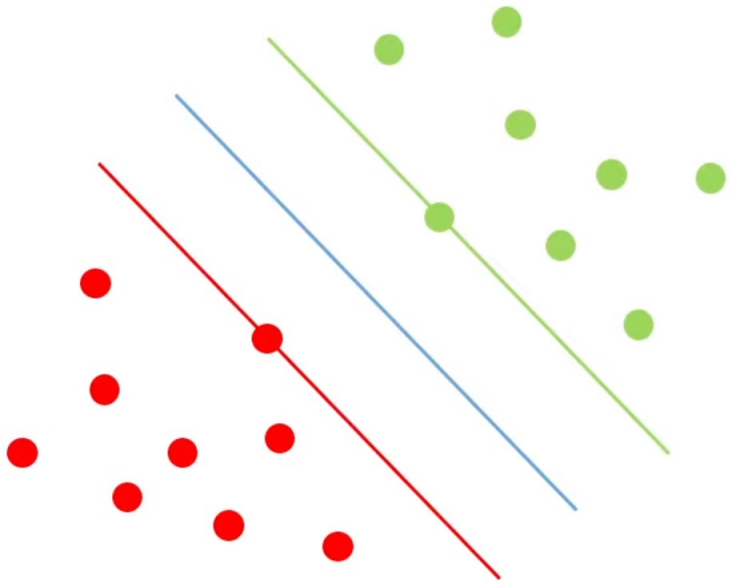
$$(x_1 - x_2) = \frac{2}{||\, w \,||} \quad \text{(margin)}$$

$$\max \left(\frac{2}{||\, w \,||} \right) \quad \text{Such that,}$$

$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases}$$

Support Vector Machine Classifier

Maximum margin without overfitting:



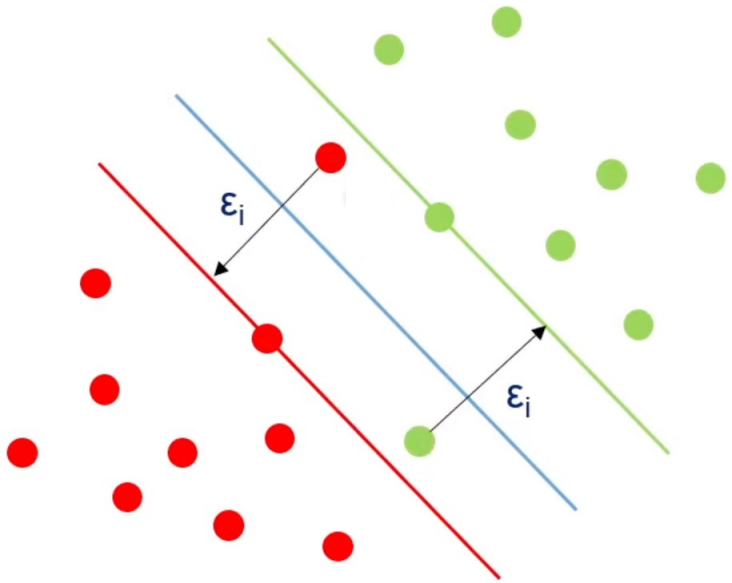
$$\max \left(\frac{2}{||w||} \right) \quad \text{Such that,}$$

$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases}$$

$$\min \left(\frac{||w||}{2} \right) + c * \Sigma \varepsilon_i$$

Support Vector Machine Classifier

Maximum margin without overfitting:



$$\max \left(\frac{2}{||\, w \,||} \right) \quad \text{Such that,}$$

$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases}$$

$$\min \left(\frac{||\, w \,||}{2} \right) + c * \sum \varepsilon_i$$

c --> Number of errors

ε_i --> Error magnitude

Loss in SVM

Loss Function

Loss function measures how far an estimated value is from its true value.

It is helpful to determine which model performs better & which parameters are better.

$$\textit{Loss} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

For Support Vector Machine Classifier "Hinge Loss" is used as the Loss Function.

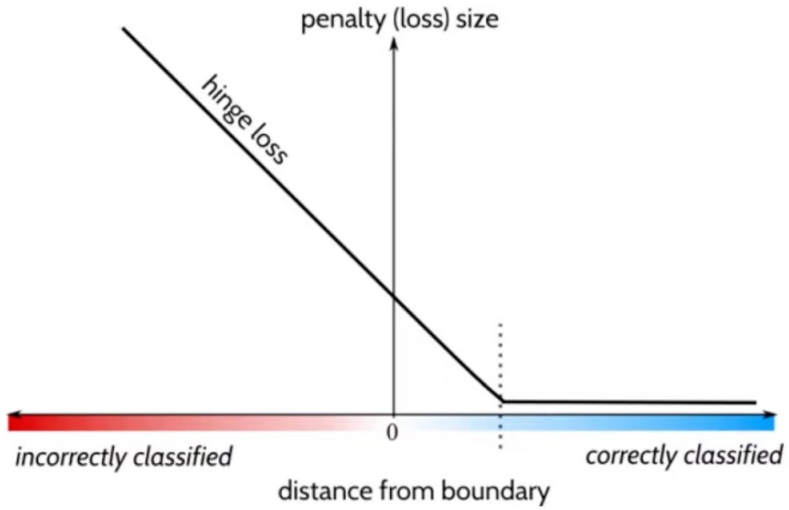
Hinge Loss

Hinge Loss is one of the types of Loss Function, mainly used for maximum margin classification models.

Hinge Loss incorporates a margin or distance from the classification boundary into the loss calculation. Even if new observations are classified correctly, they can incur a penalty if the margin from the decision boundary is not large enough.

$$L = \max(0, 1 - y_i (w^T x_i + b))$$

0 - for correct classification
1 - for wrong classification



Hinge Loss

Misclassification :

$y_i = 1 \quad \hat{y}_i = -1$

$L = (1 - (1)(-1)$

$L = (1 + 1)$

$L = 2$ (High loss Value)

$y_i = -1 \quad \hat{y}_i = 1$

$L = (1 - (-1)(1)$

$L = (1 + 1)$

$L = 2$ (High loss Value)

0 - for correct classification
1 - for wrong classification

Correct classification :

$y_i = 1 \quad \hat{y}_i = 1$

$L = (0 - (1)(1)$

$L = (0 - 1)$

$L = -1$ (Low loss Value)

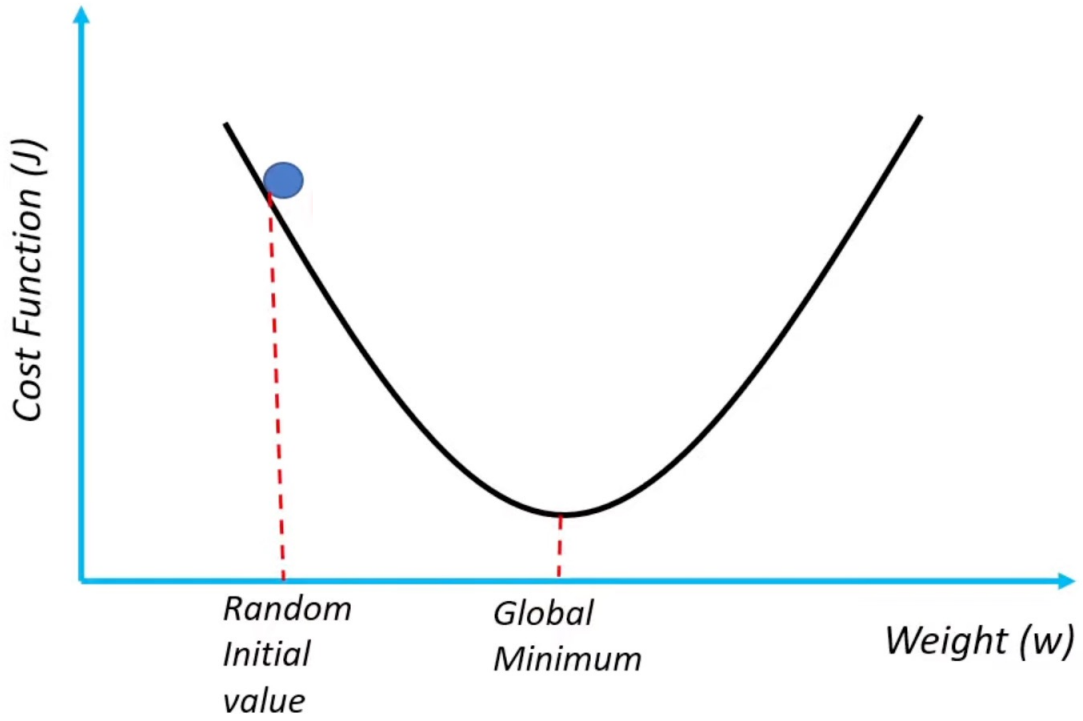
$y_i = -1 \quad \hat{y}_i = -1$

$L = (0 - (-1)(-1)$

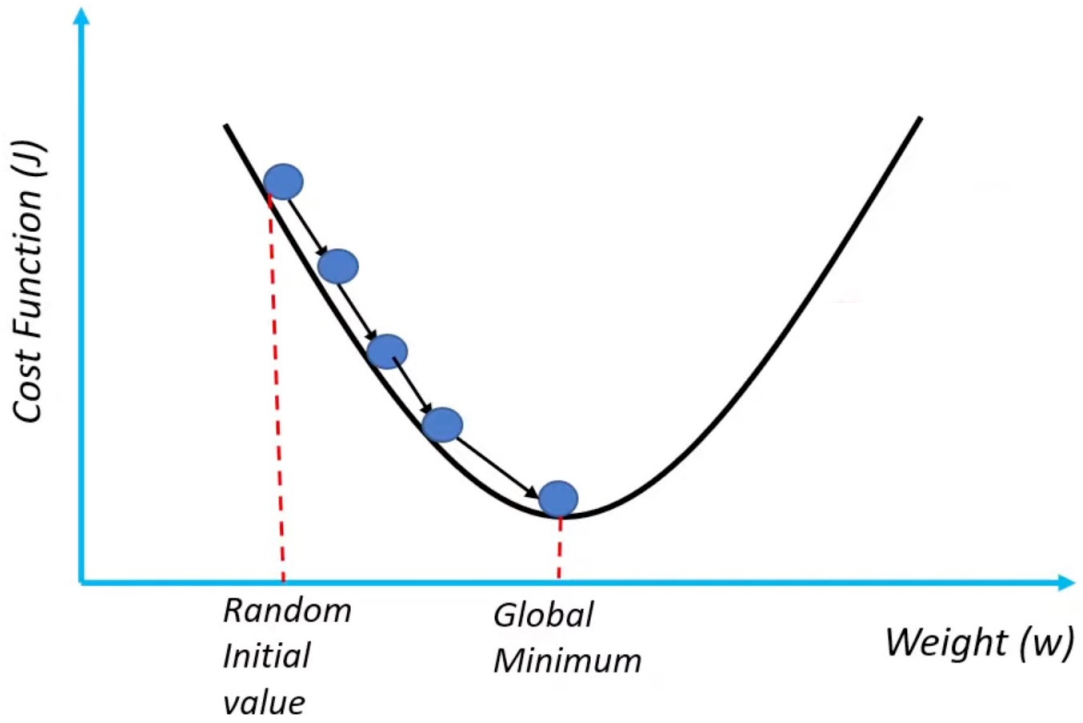
$L = (0 - 1)$

$L = -1$ (Low loss Value)

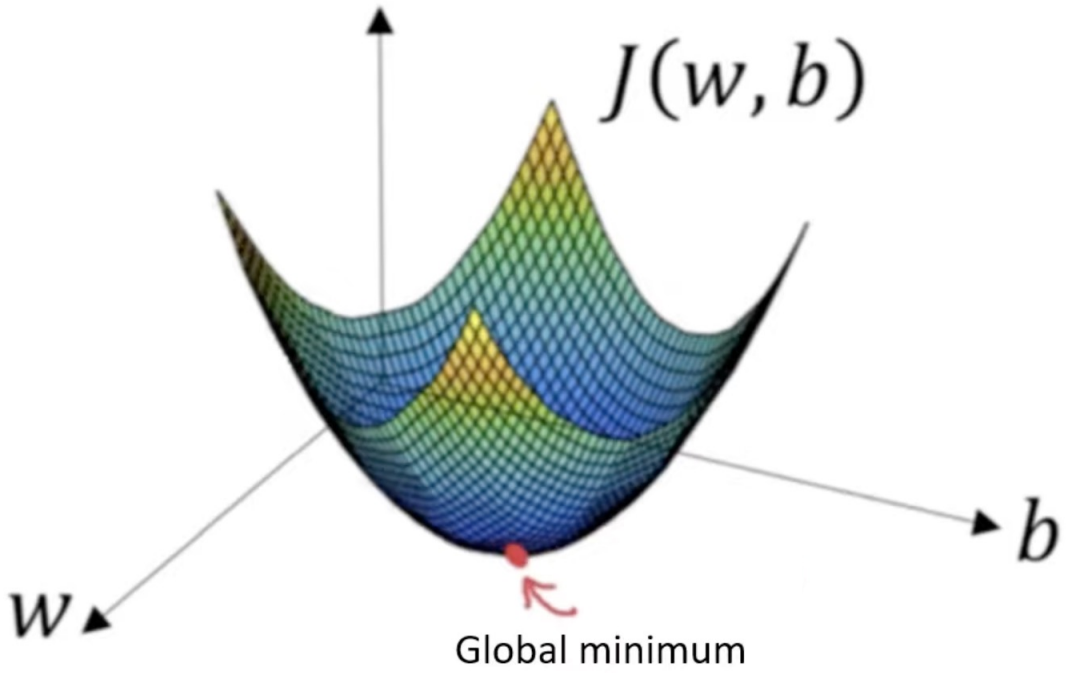
Gradient Descent



Gradient Descent



Gradient Descent in 3 Dimension



Gradient Descent

Gradient Descent is an optimization algorithm used for minimizing the cost function in various machine learning algorithms. It is used for updating the parameters of the learning model.

$$w_2 = w_1 - L * \frac{dJ}{dw}$$

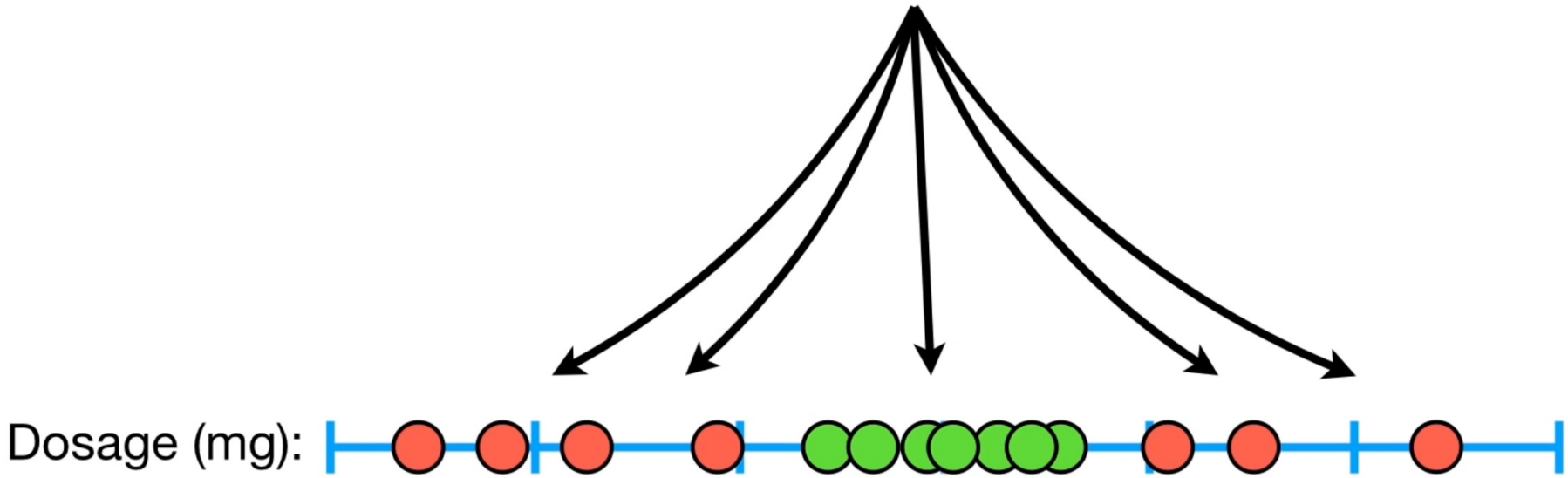
$$b_2 = b_1 - L * \frac{dJ}{db}$$

w --> weight
b --> bias
L --> Learning Rate

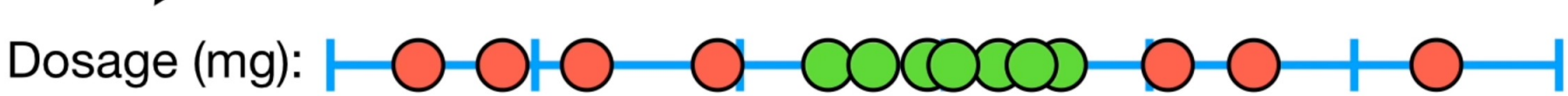
$\frac{dJ}{dw}$ --> Partial Derivative of cost function with respect to w

$\frac{dJ}{db}$ --> Partial Derivative of cost function with respect to b

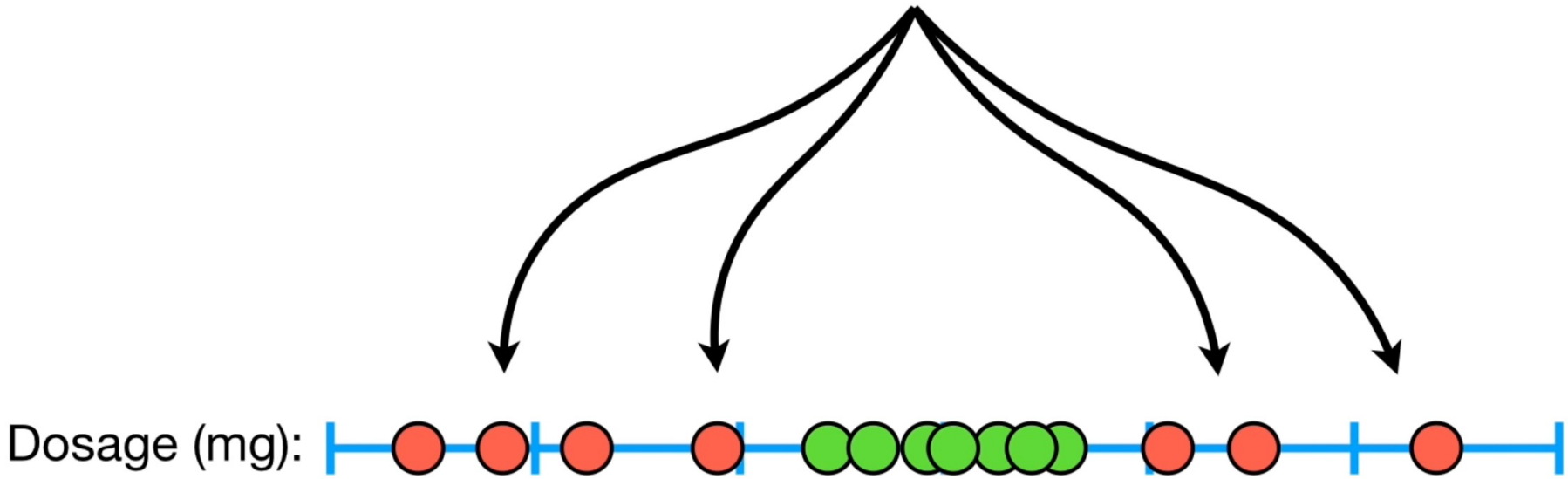
…but what if this was our training data and we had tons of overlap?

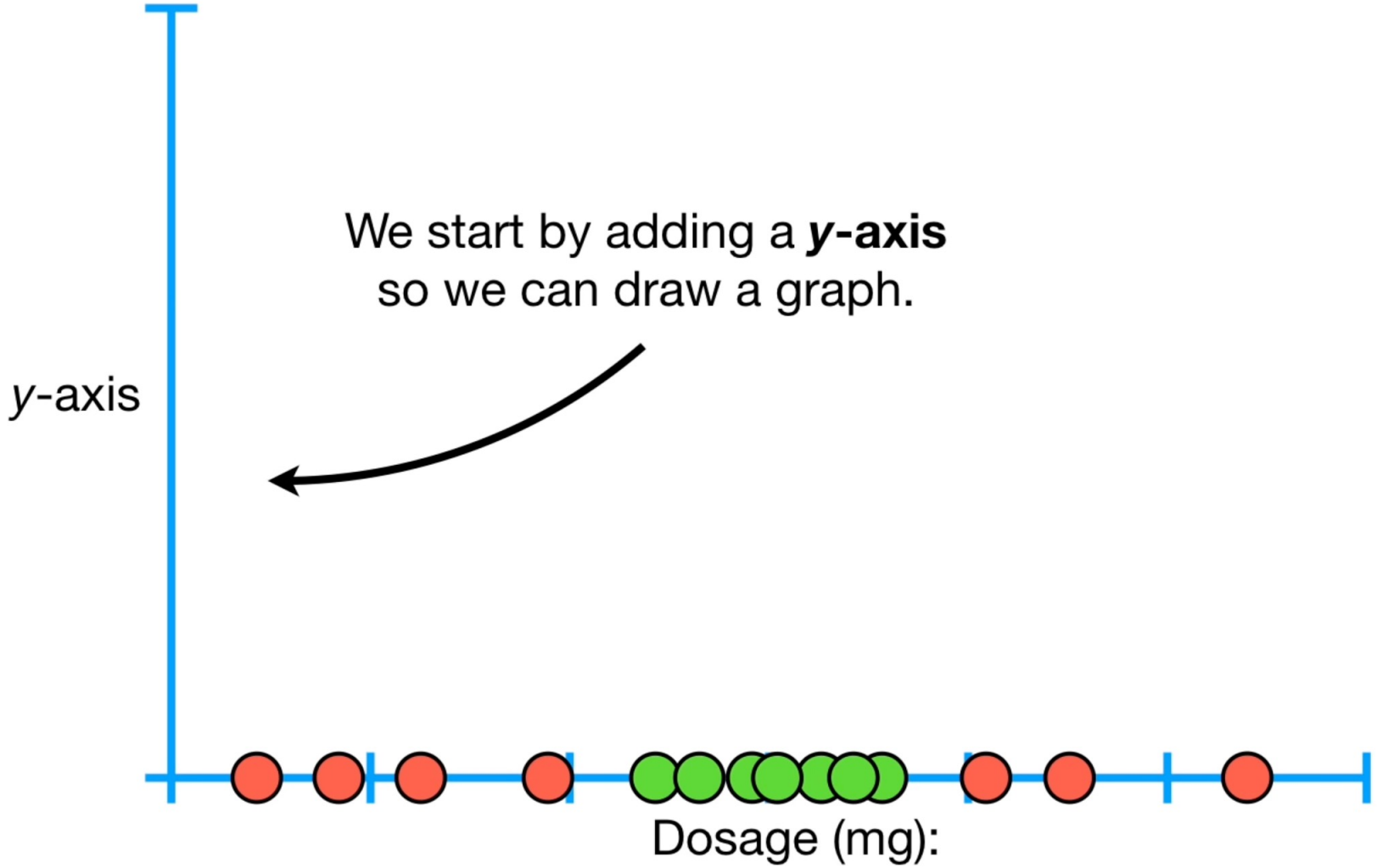


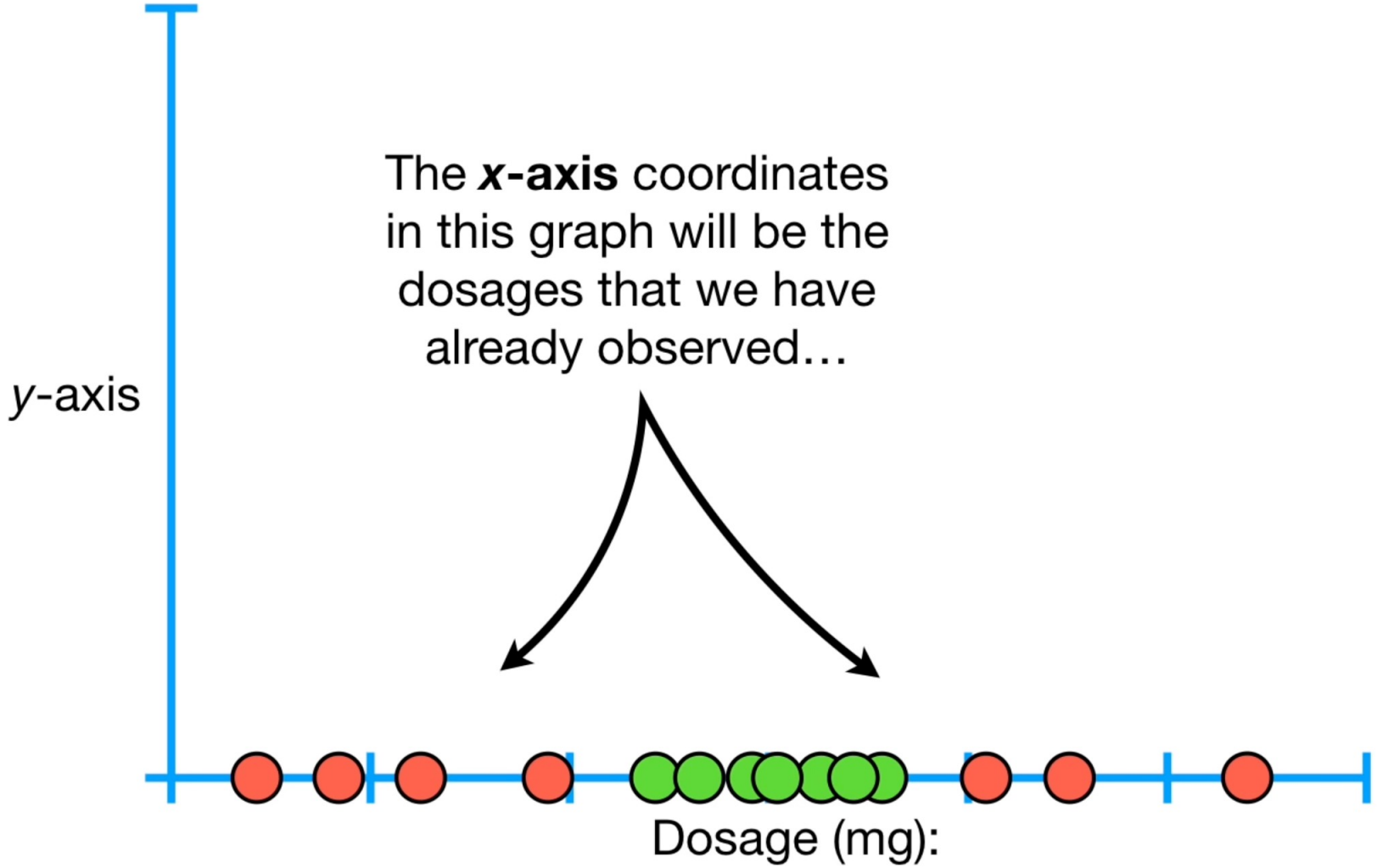
In this new example, with tons of overlap, we are now looking at **Drug Dosages**…

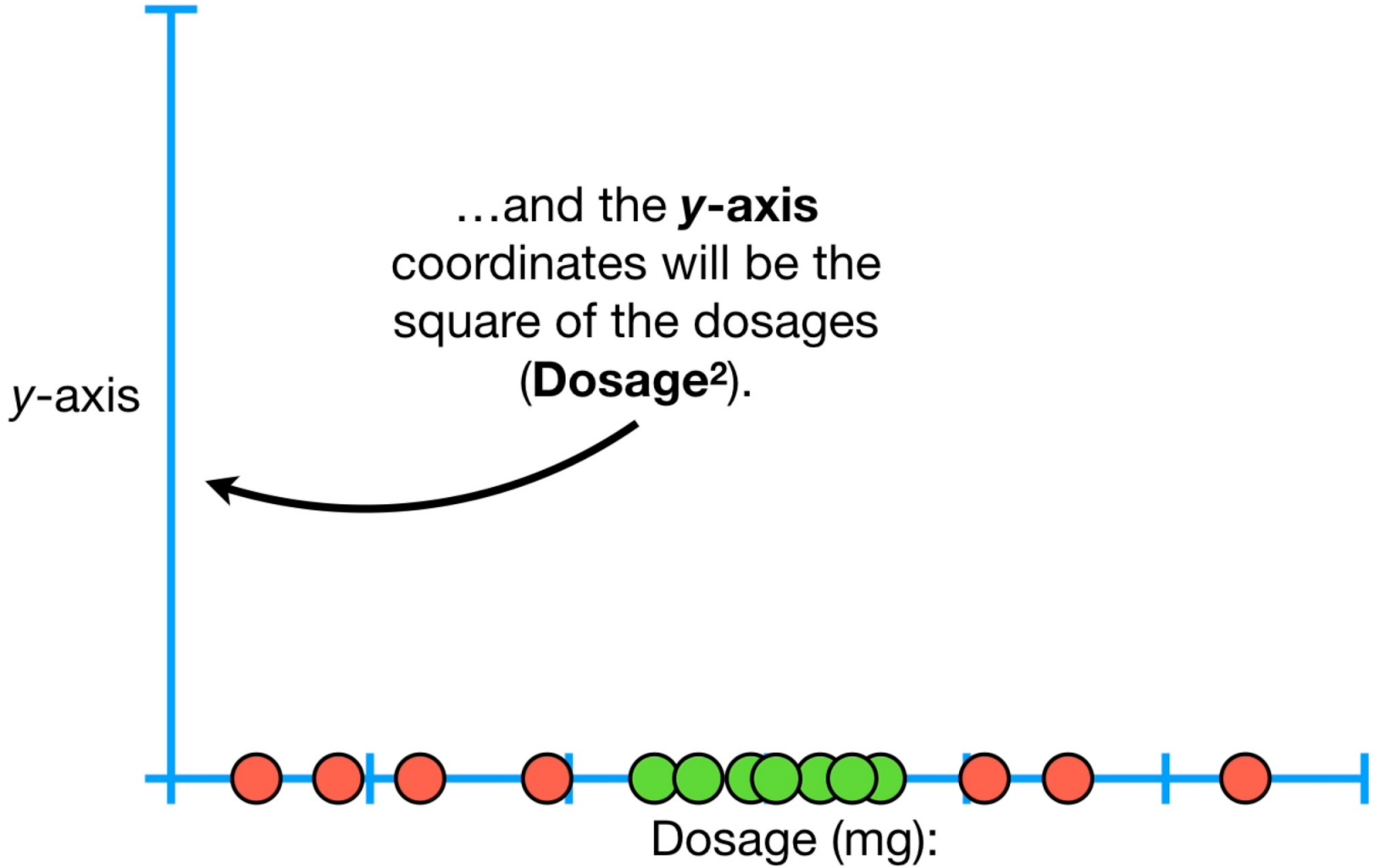


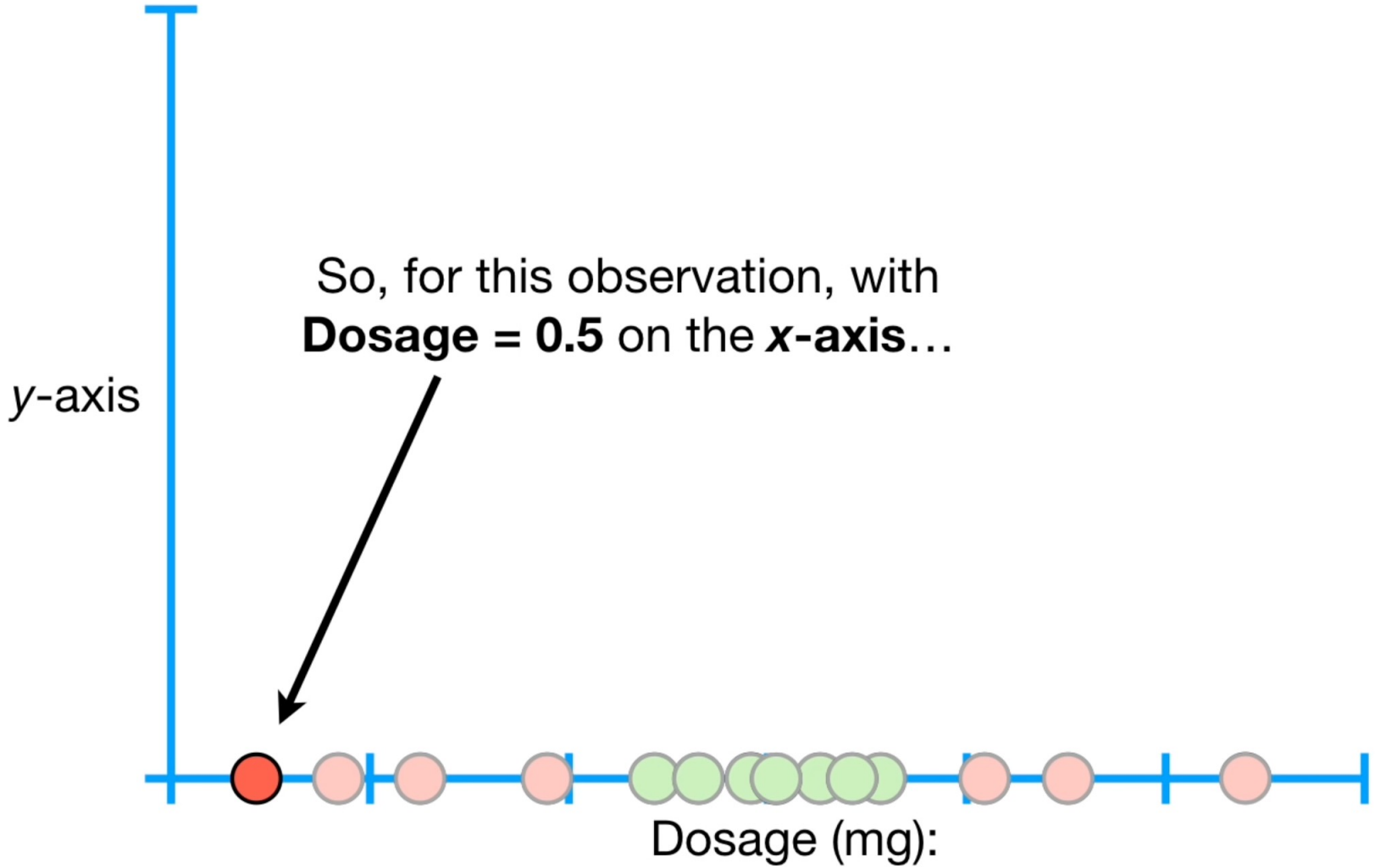
…and the **red dots** represent
patients that were ***not cured***…

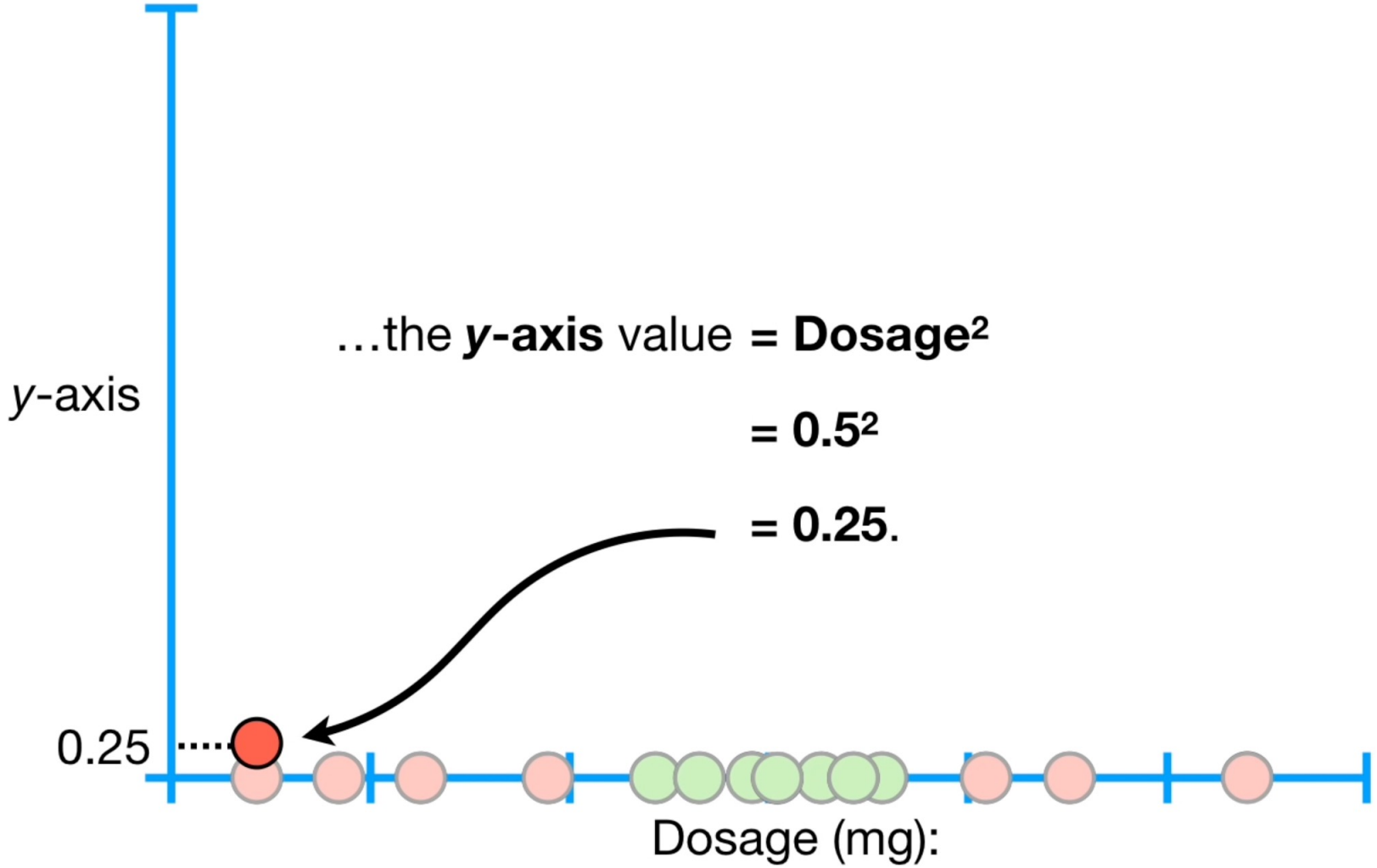


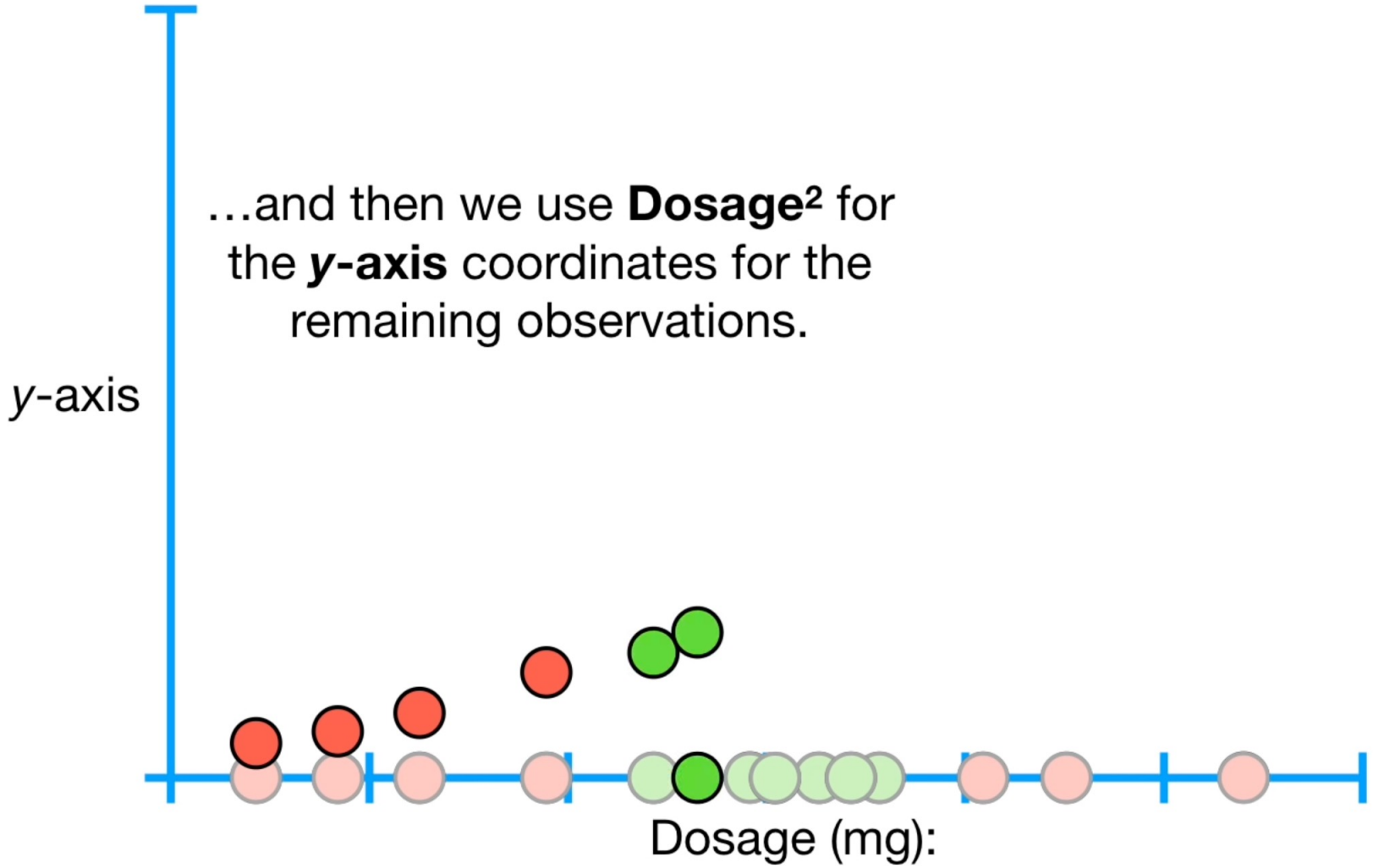

We start by adding a **y-axis**
so we can draw a graph.
y-axis
Dosage (mg):

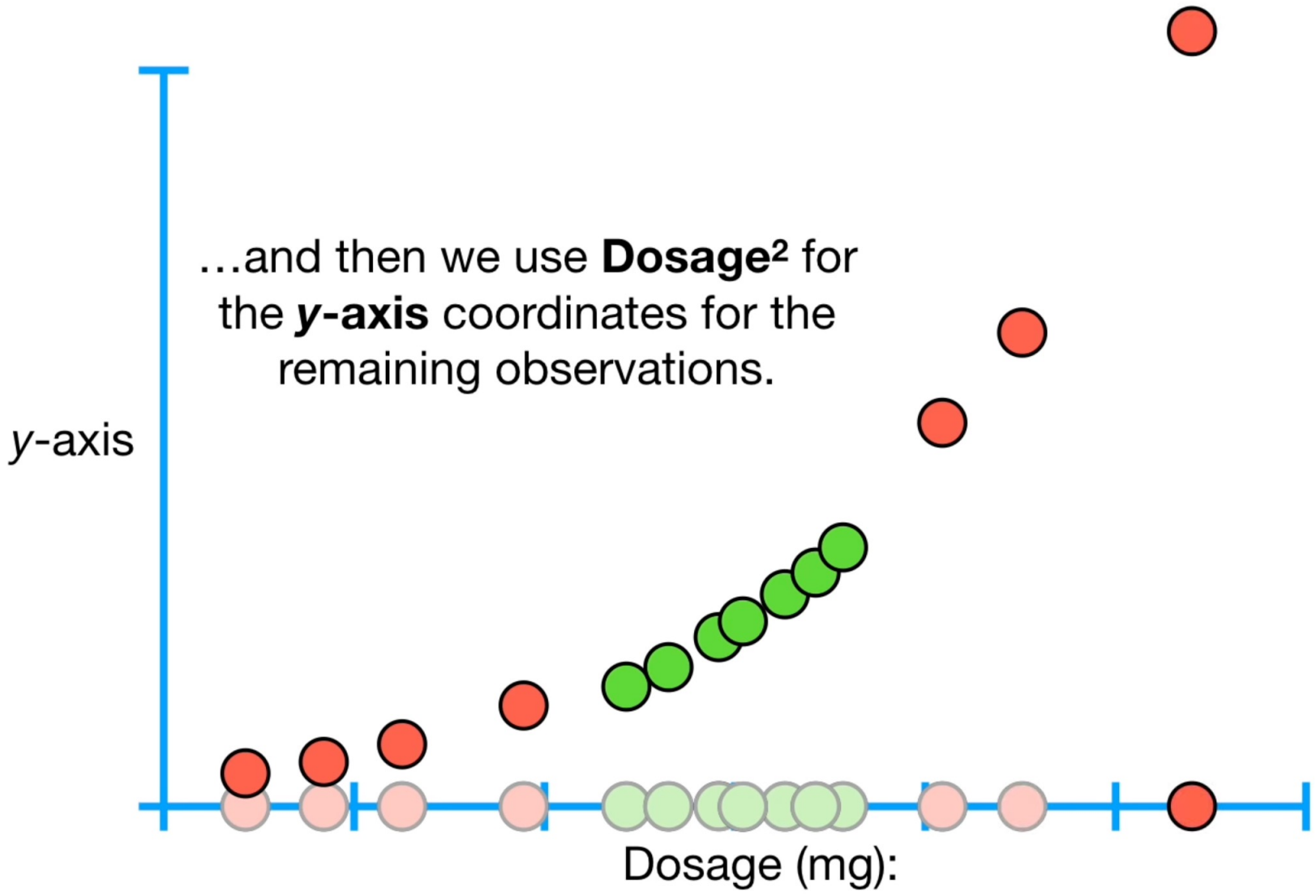

The *x*-axis coordinates in this graph will be the dosages that we have already observed…
y-axis
Dosage (mg):

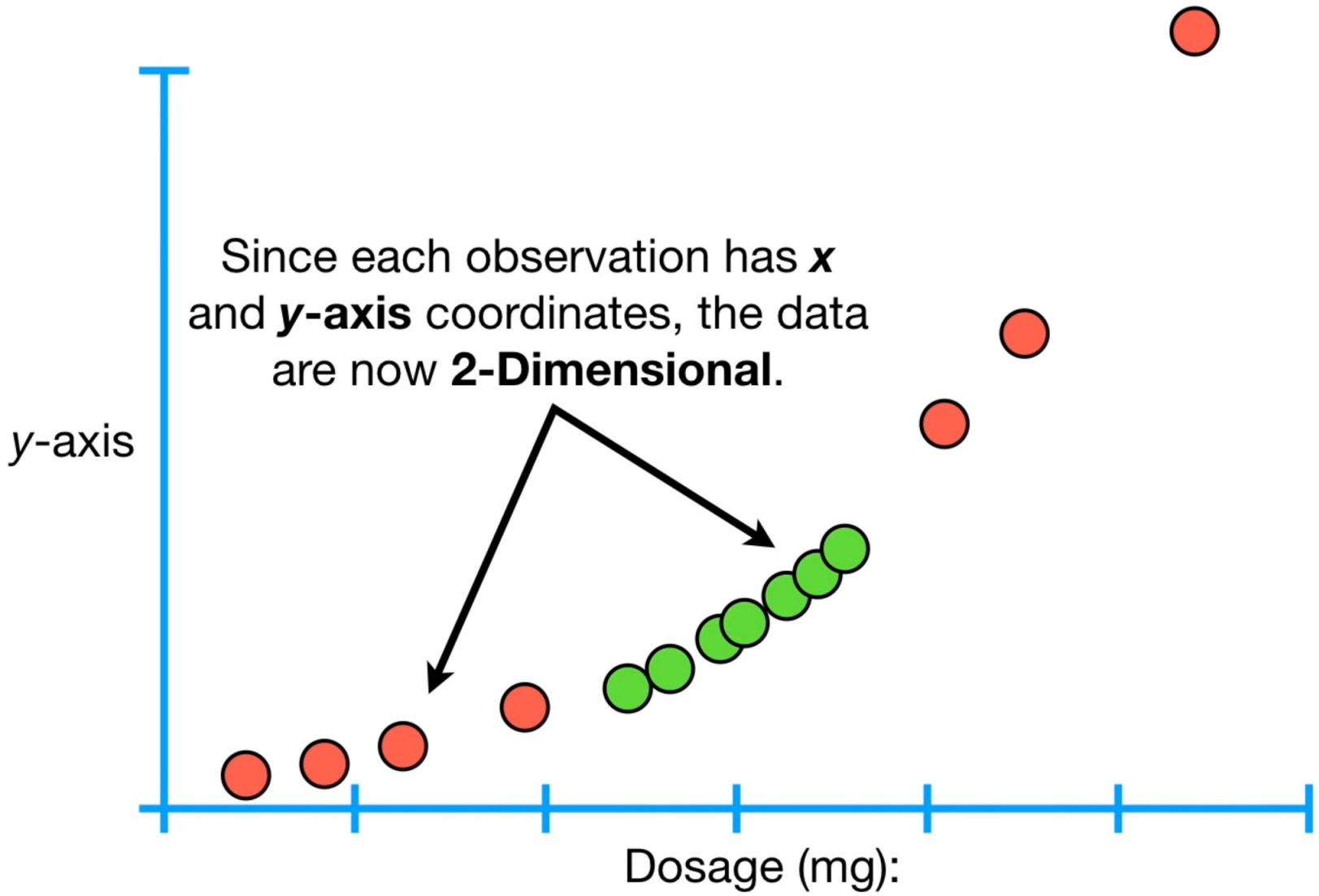

…and the **y-axis** coordinates will be the square of the dosages (**Dosage²**).
y-axis
Dosage (mg):

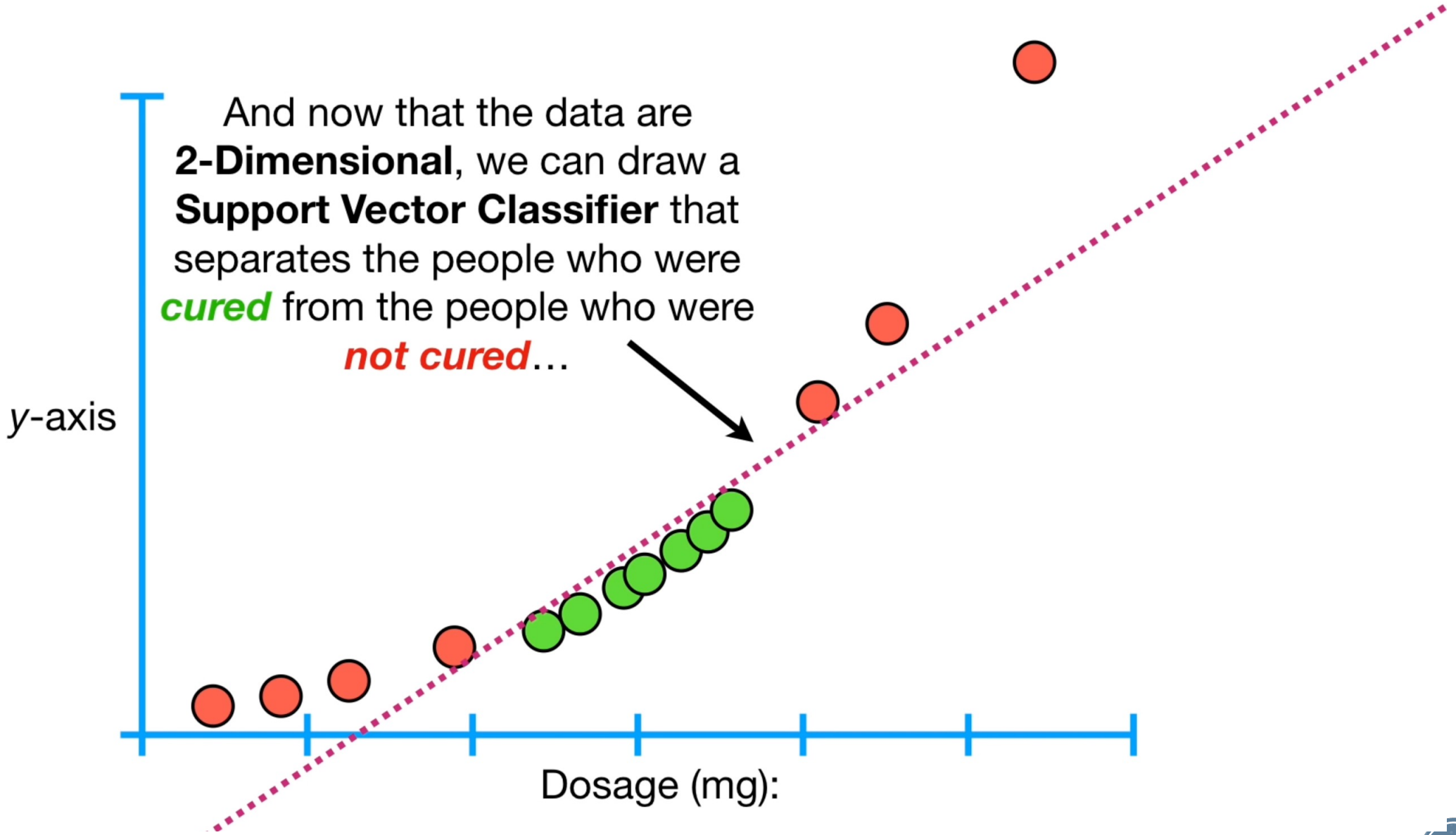

So, for this observation, with
Dosage = 0.5 on the ***x*-axis**…
y-axis
Dosage (mg):

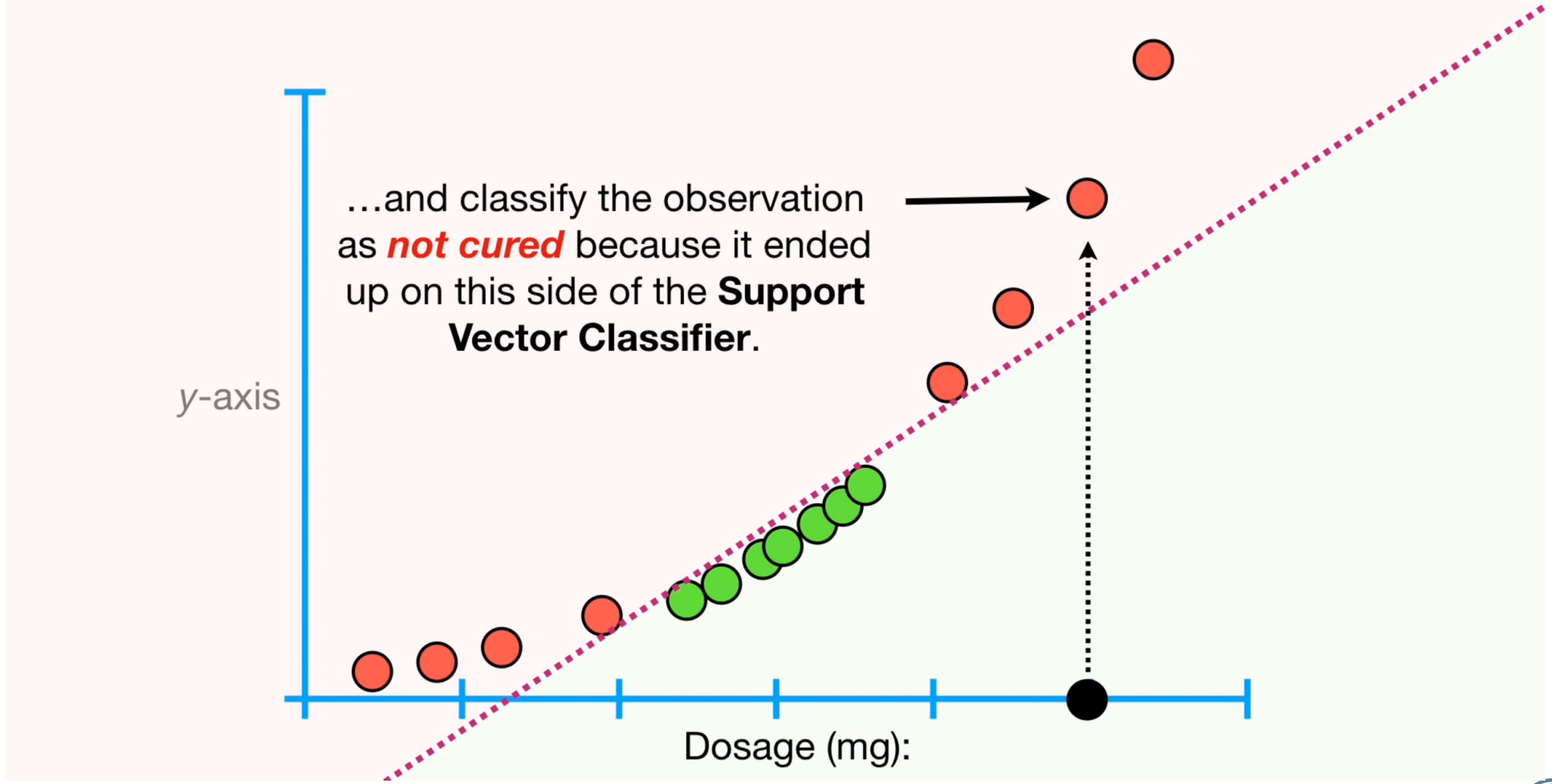

y-axis
…the y-axis value = Dosage²
= 0.5²
= 0.25.
0.25
Dosage (mg):

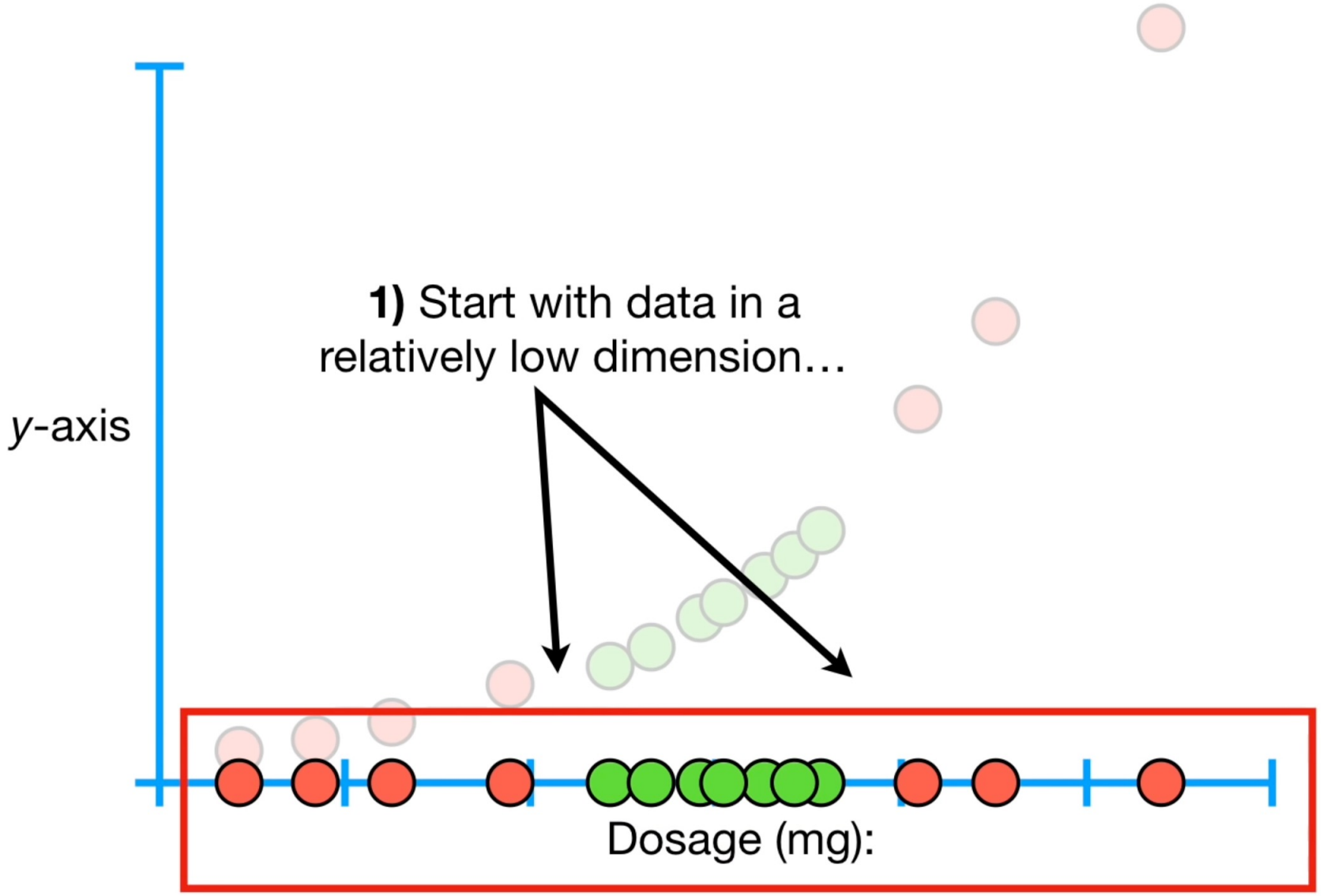

…and then we use **Dosage²** for the ***y*-axis** coordinates for the remaining observations.
y-axis
Dosage (mg):

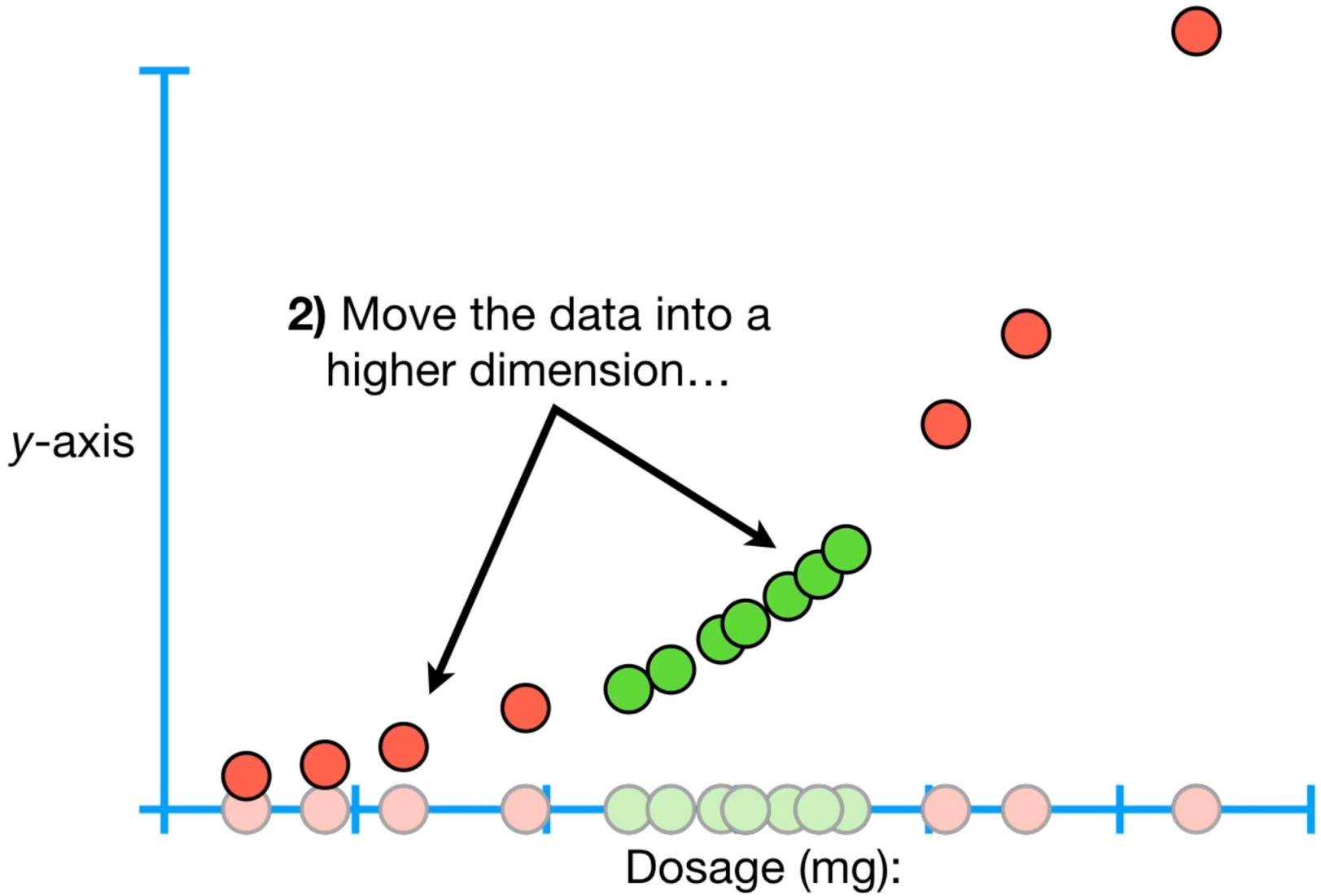

…and then we use **Dosage²** for the ***y*-axis** coordinates for the remaining observations.
y-axis
Dosage (mg):

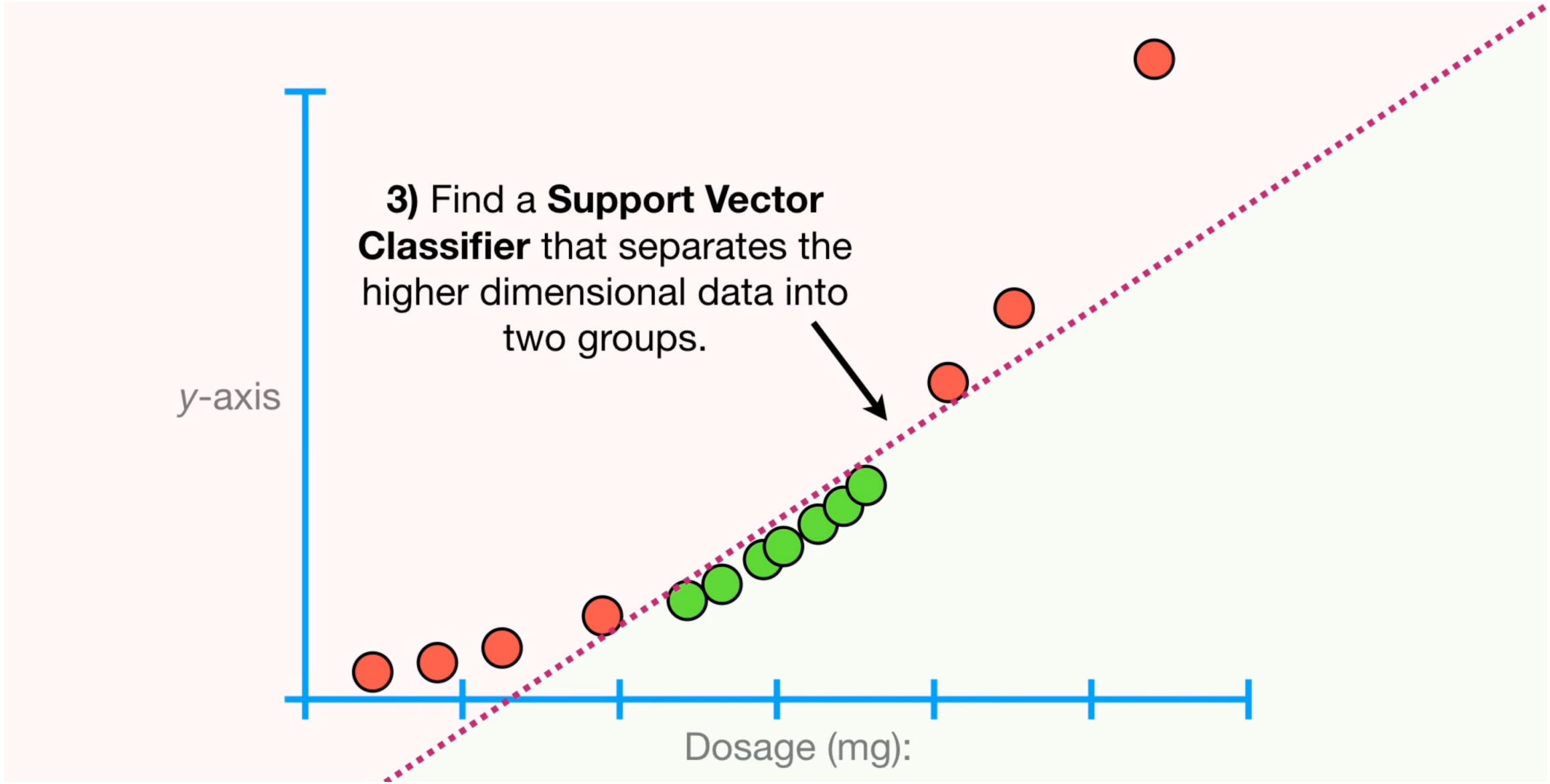

Since each observation has *x* and ***y*-axis** coordinates, the data are now **2-Dimensional**.
y-axis
Dosage (mg):

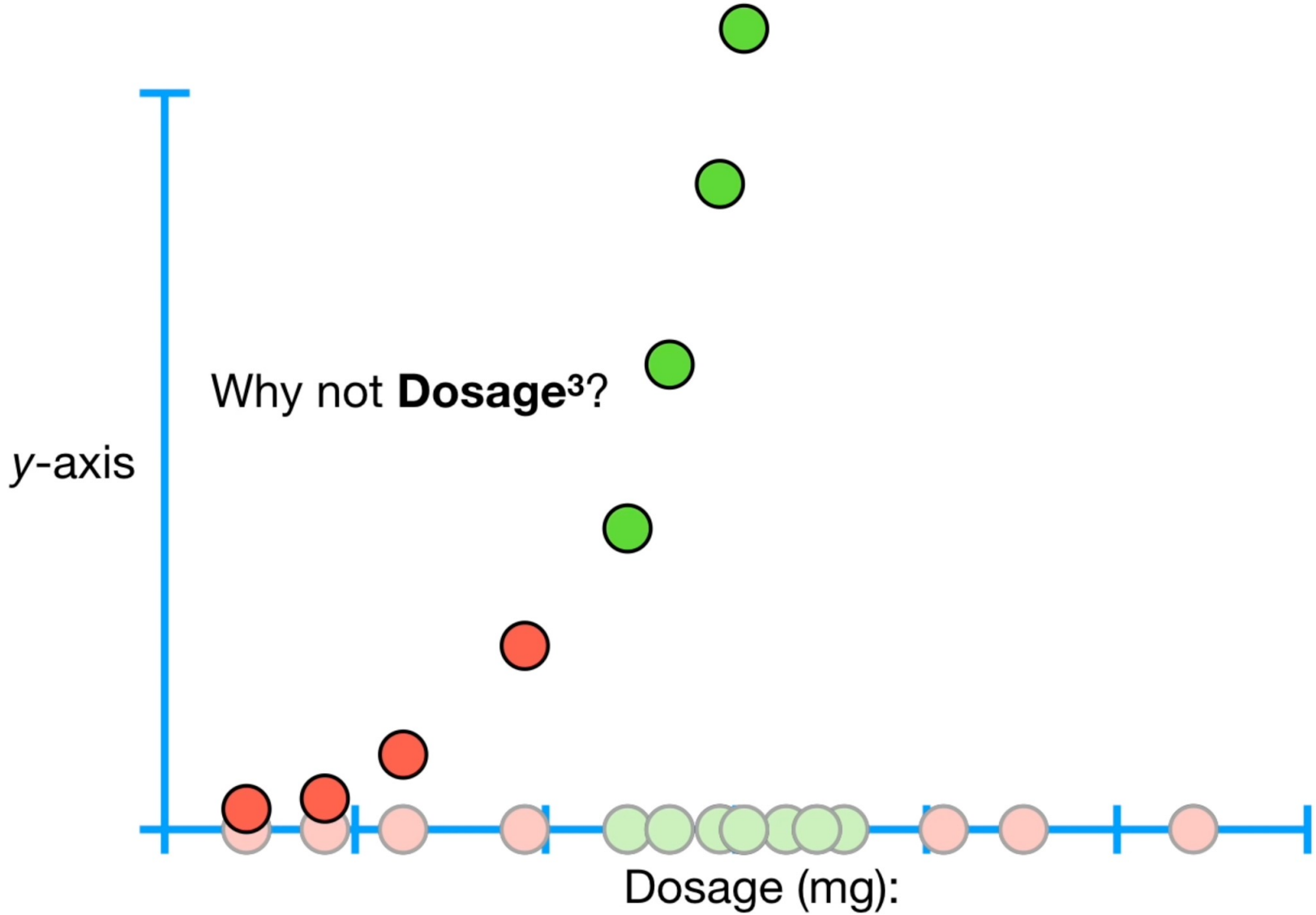

And now that the data are
2-Dimensional, we can draw a
Support Vector Classifier that
separates the people who were
cured from the people who were
not cured…
y-axis
Dosage (mg):

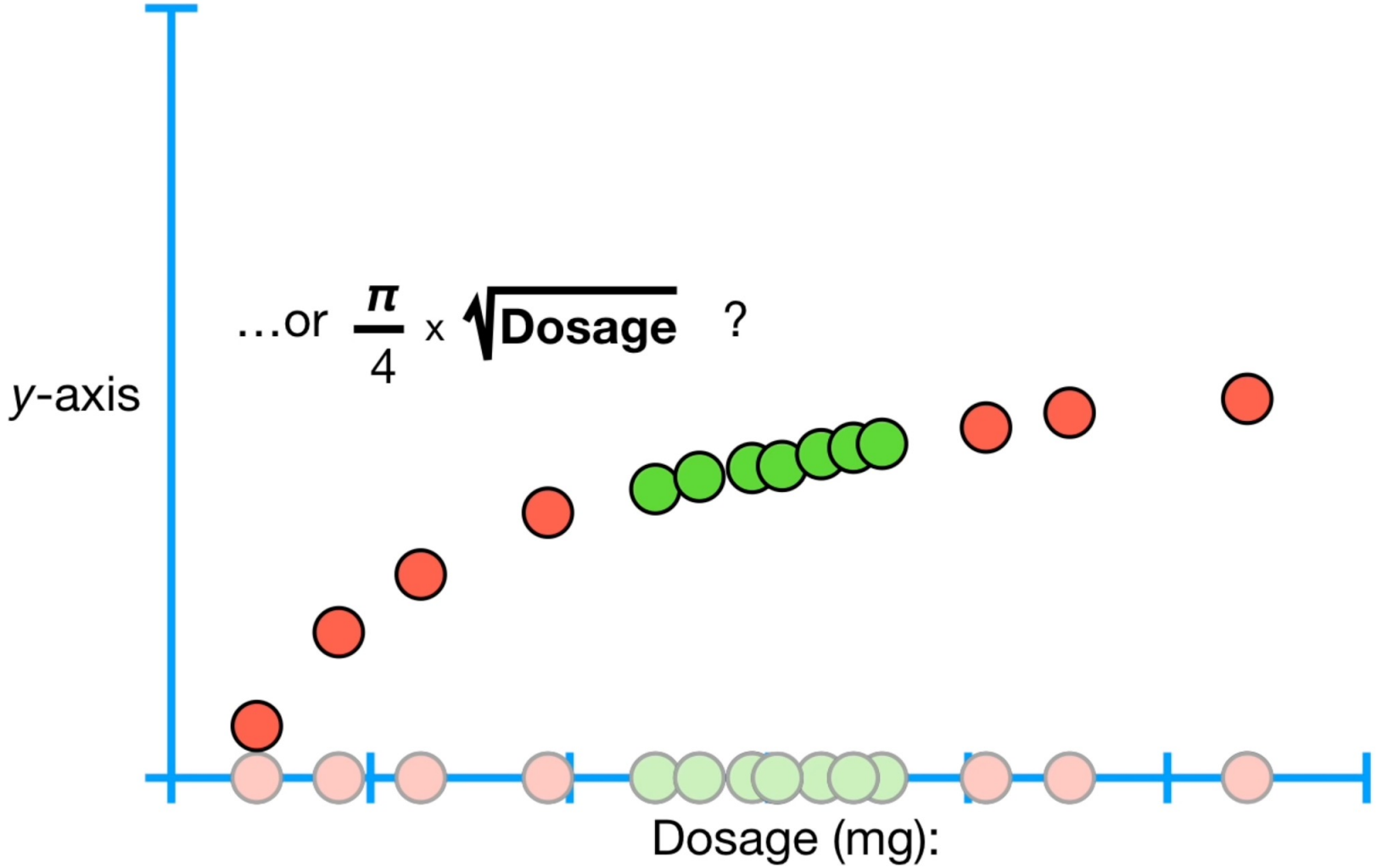

…and classify the observation as *not cured* because it ended up on this side of the **Support Vector Classifier**.
y-axis
Dosage (mg):

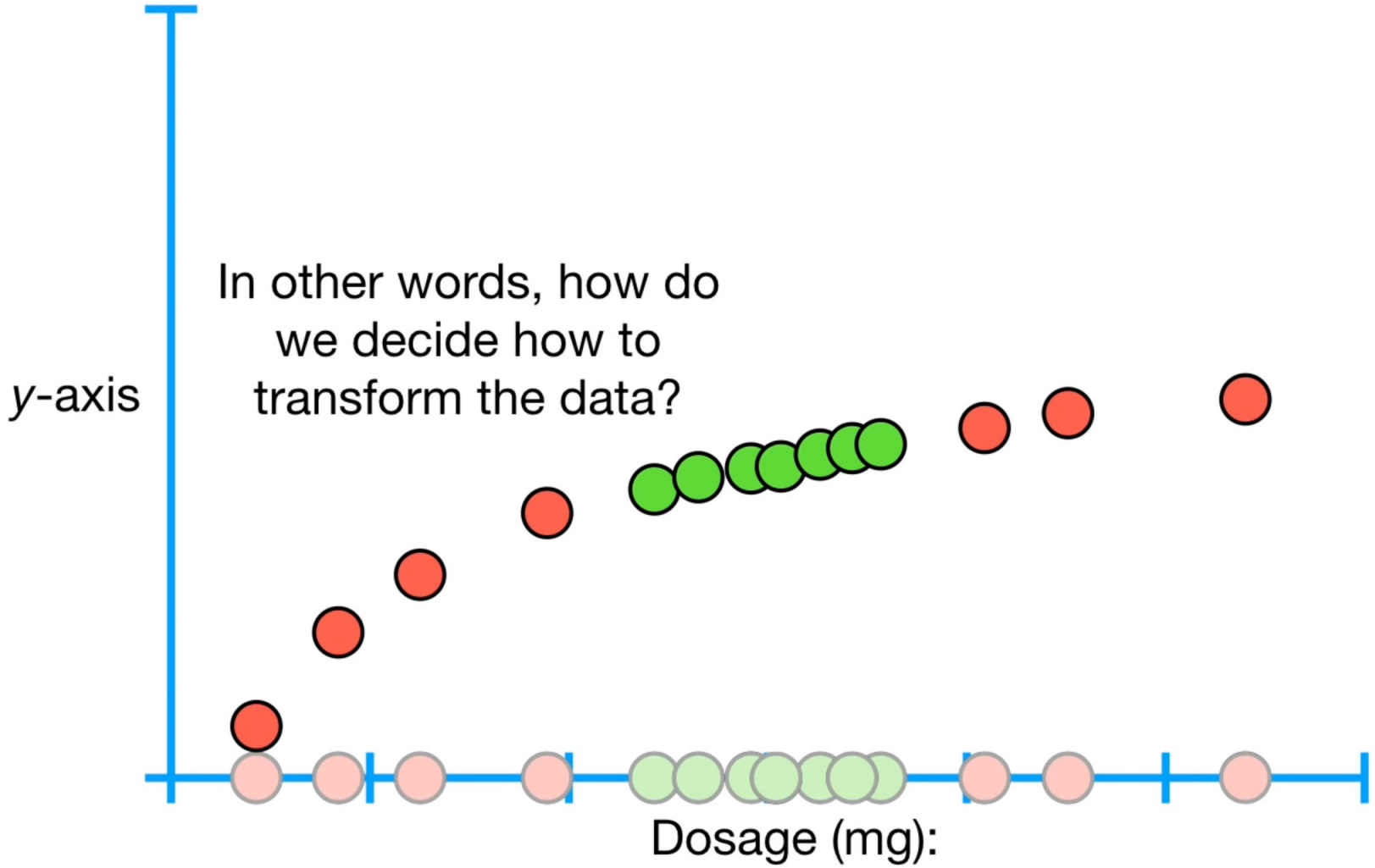

y-axis
1) Start with data in a relatively low dimension…
Dosage (mg):

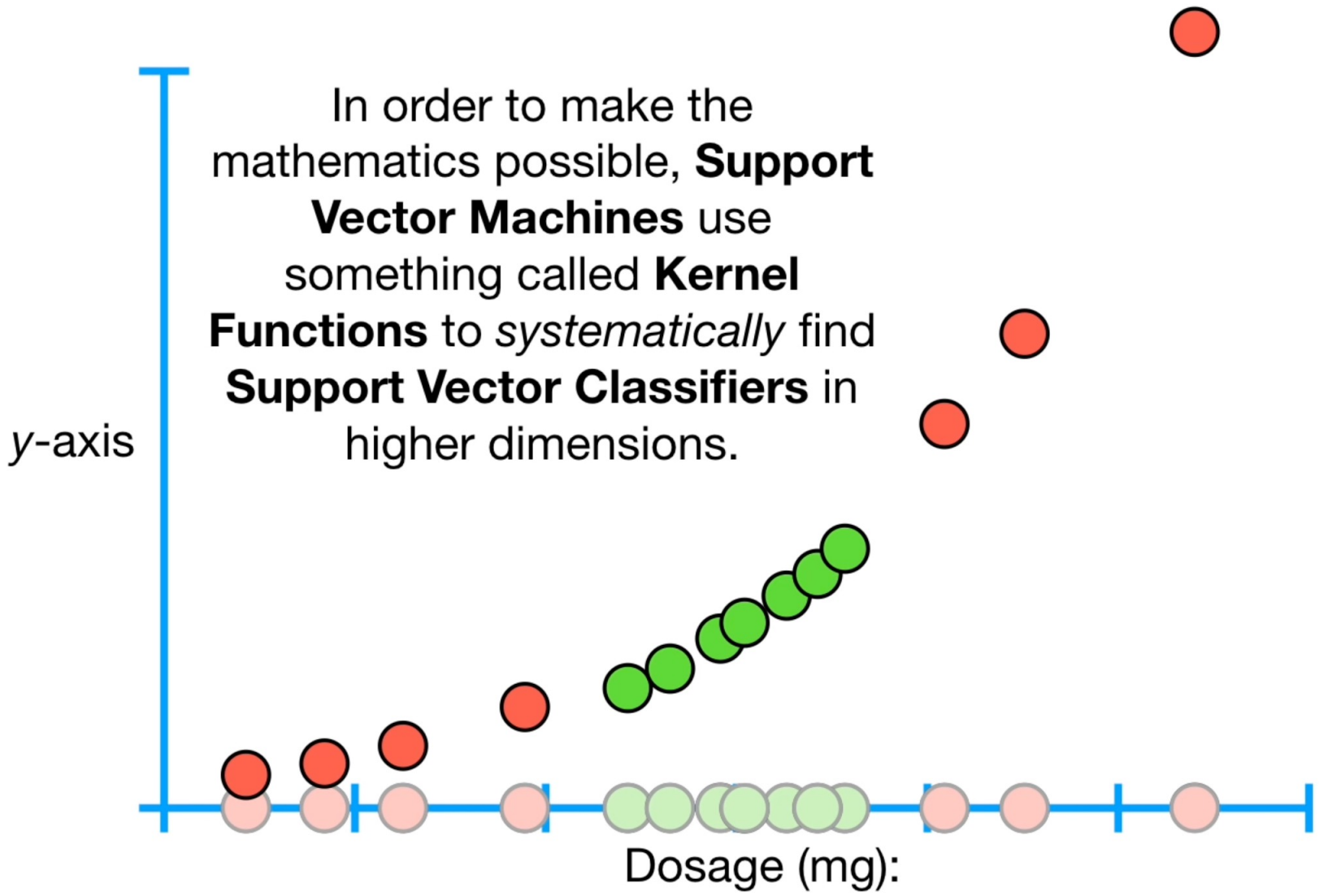

2) Move the data into a higher dimension…
y-axis
Dosage (mg):

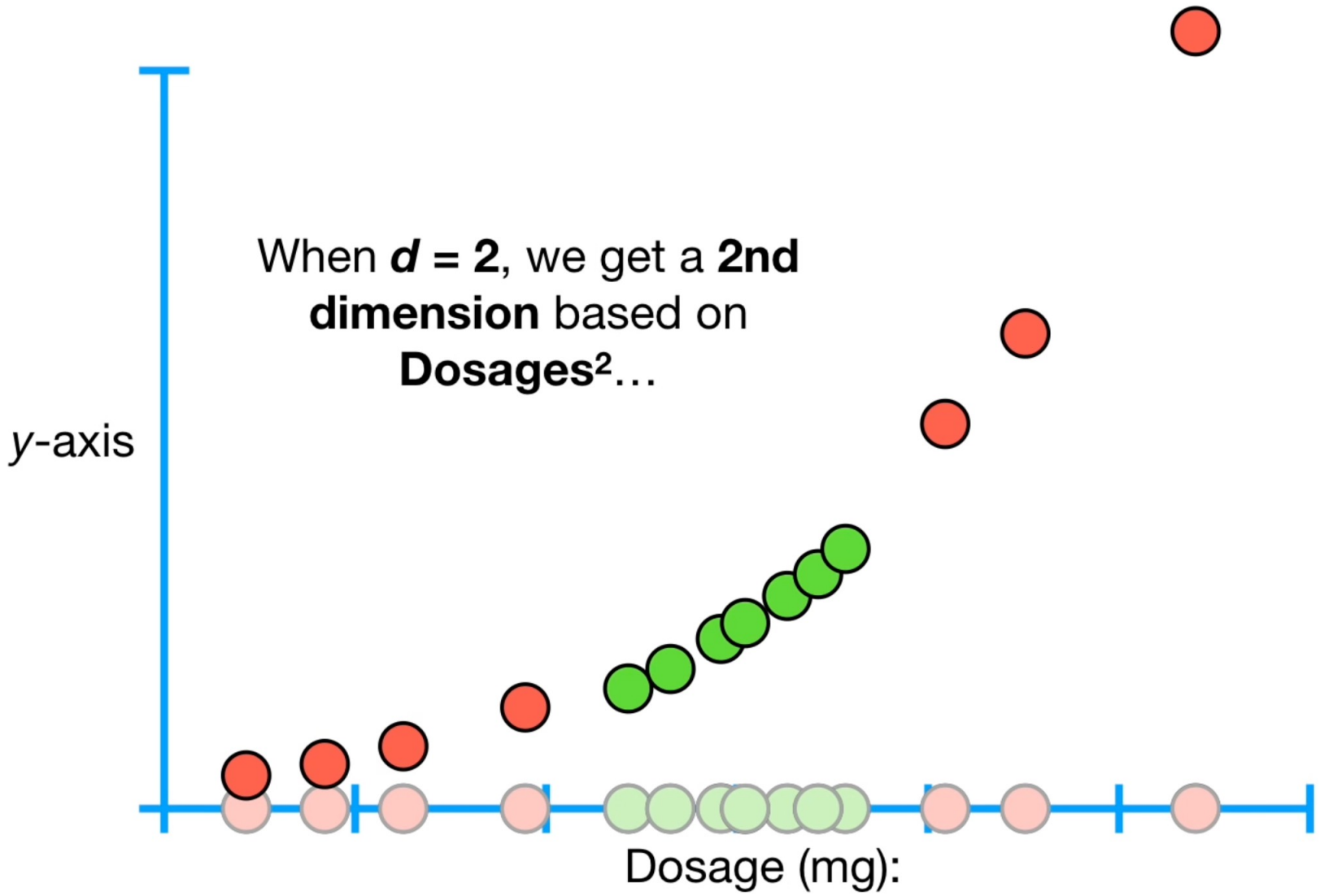

3) Find a **Support Vector Classifier** that separates the higher dimensional data into two groups.
y-axis
Dosage (mg):

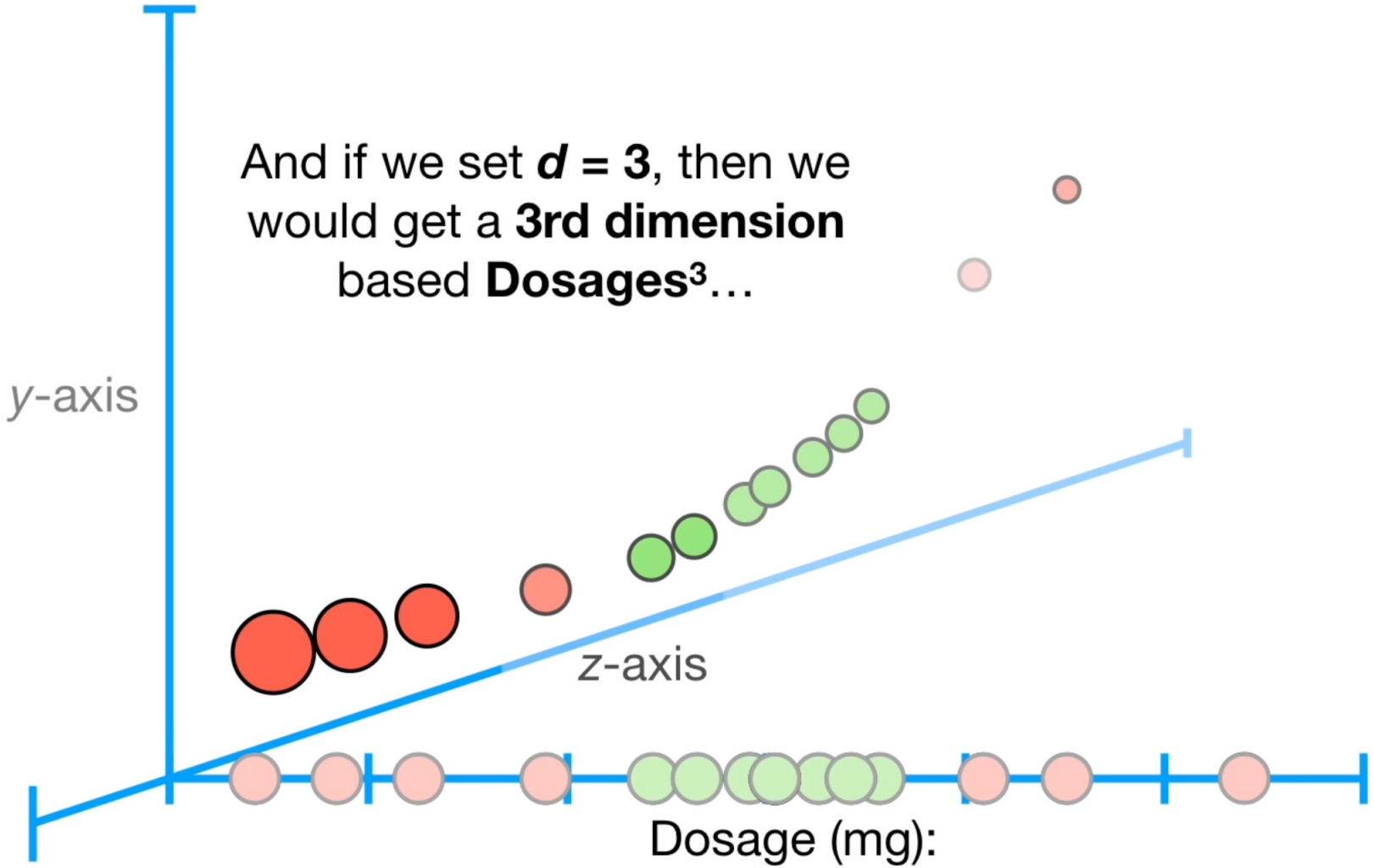

Why not **Dosage**^{3}?
y-axis
Dosage (mg):

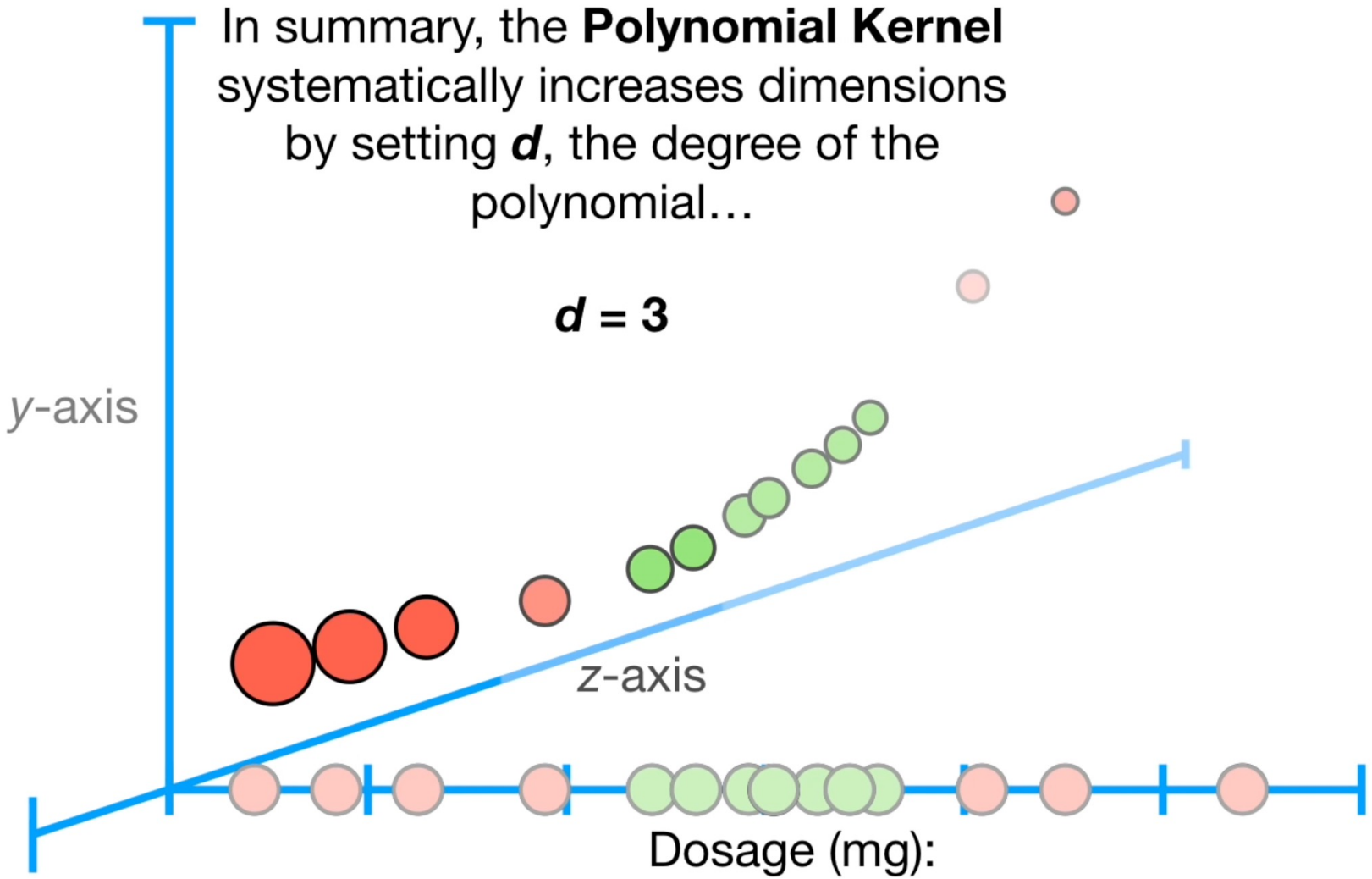

y-axis
…or $\frac{\pi}{4} \times \sqrt{\text{Dosage}}$?
Dosage (mg):

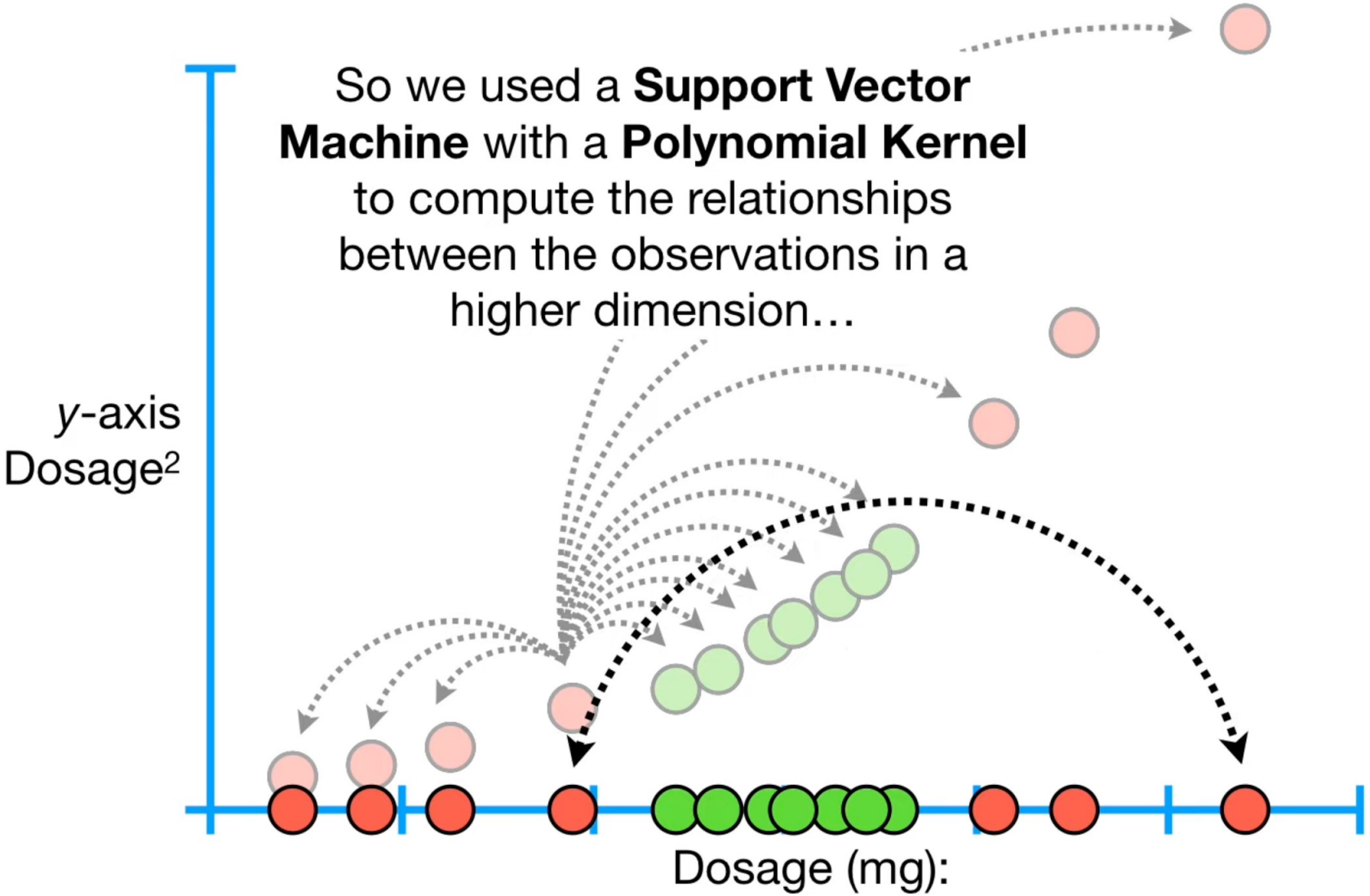

In other words, how do we decide how to transform the data?
y-axis
Dosage (mg):

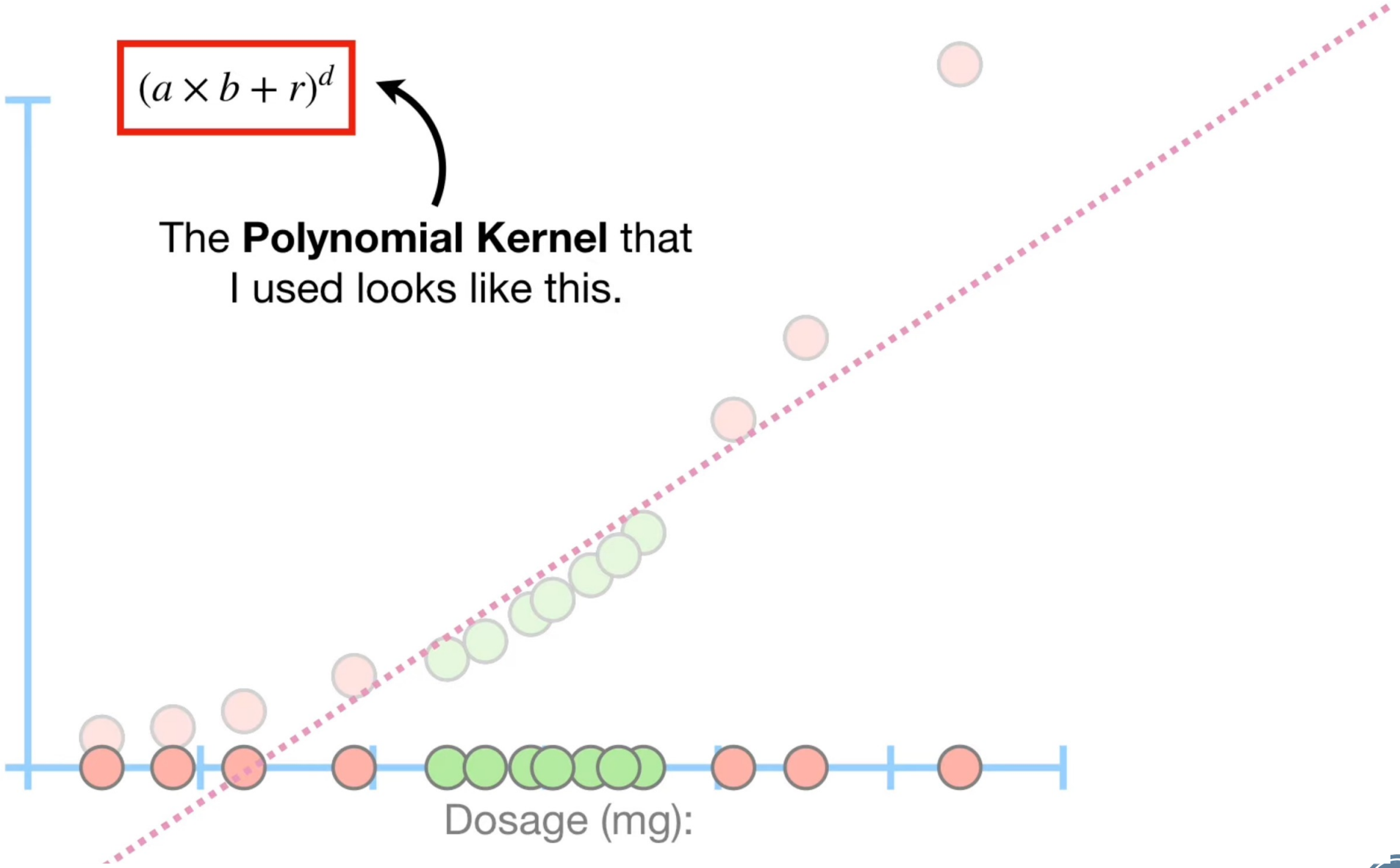

In order to make the mathematics possible, **Support Vector Machines** use something called **Kernel Functions** to *systematically* find **Support Vector Classifiers** in higher dimensions.
y-axis
Dosage (mg):

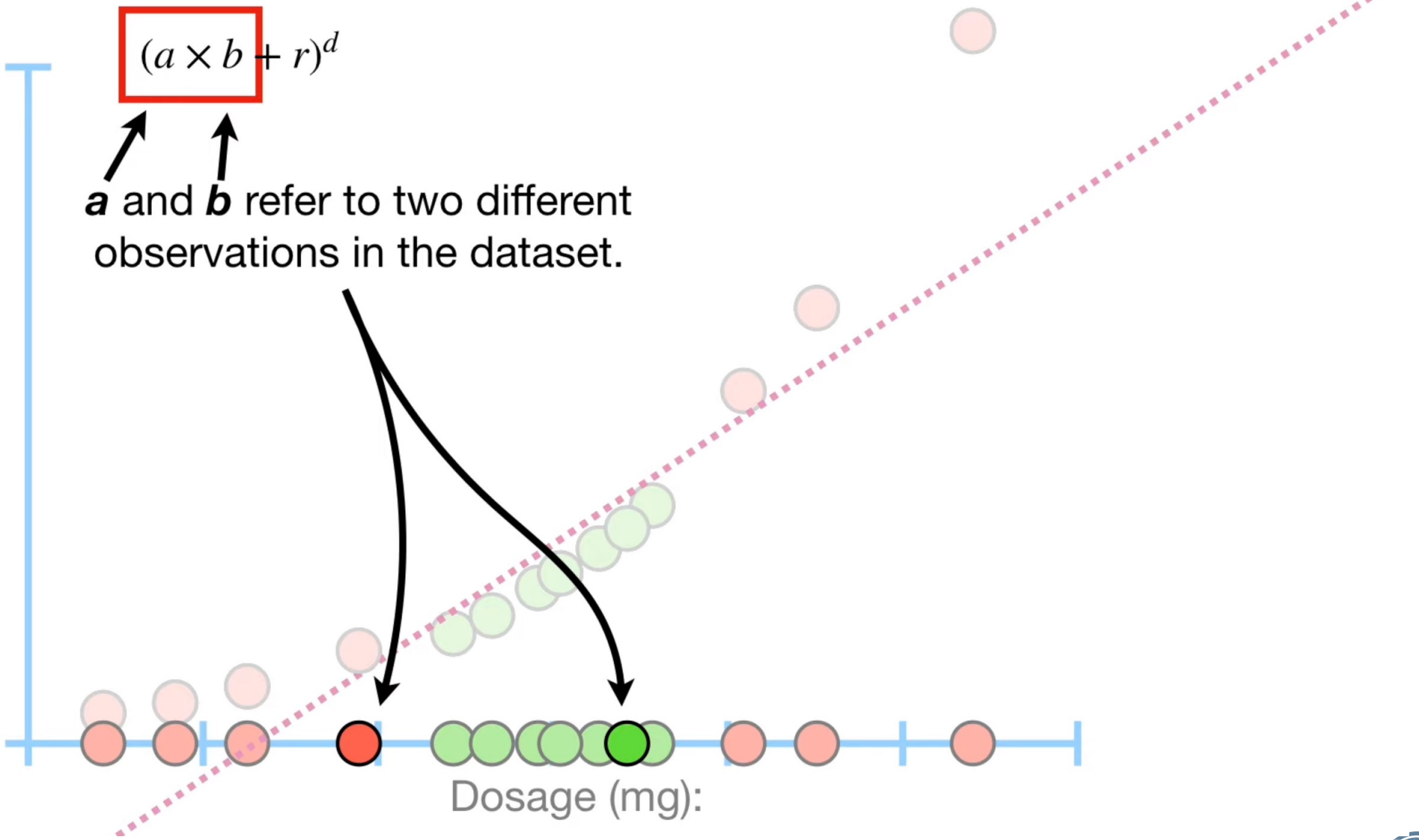

When **d = 2**, we get a **2nd dimension** based on **Dosages²**…
y-axis
Dosage (mg):

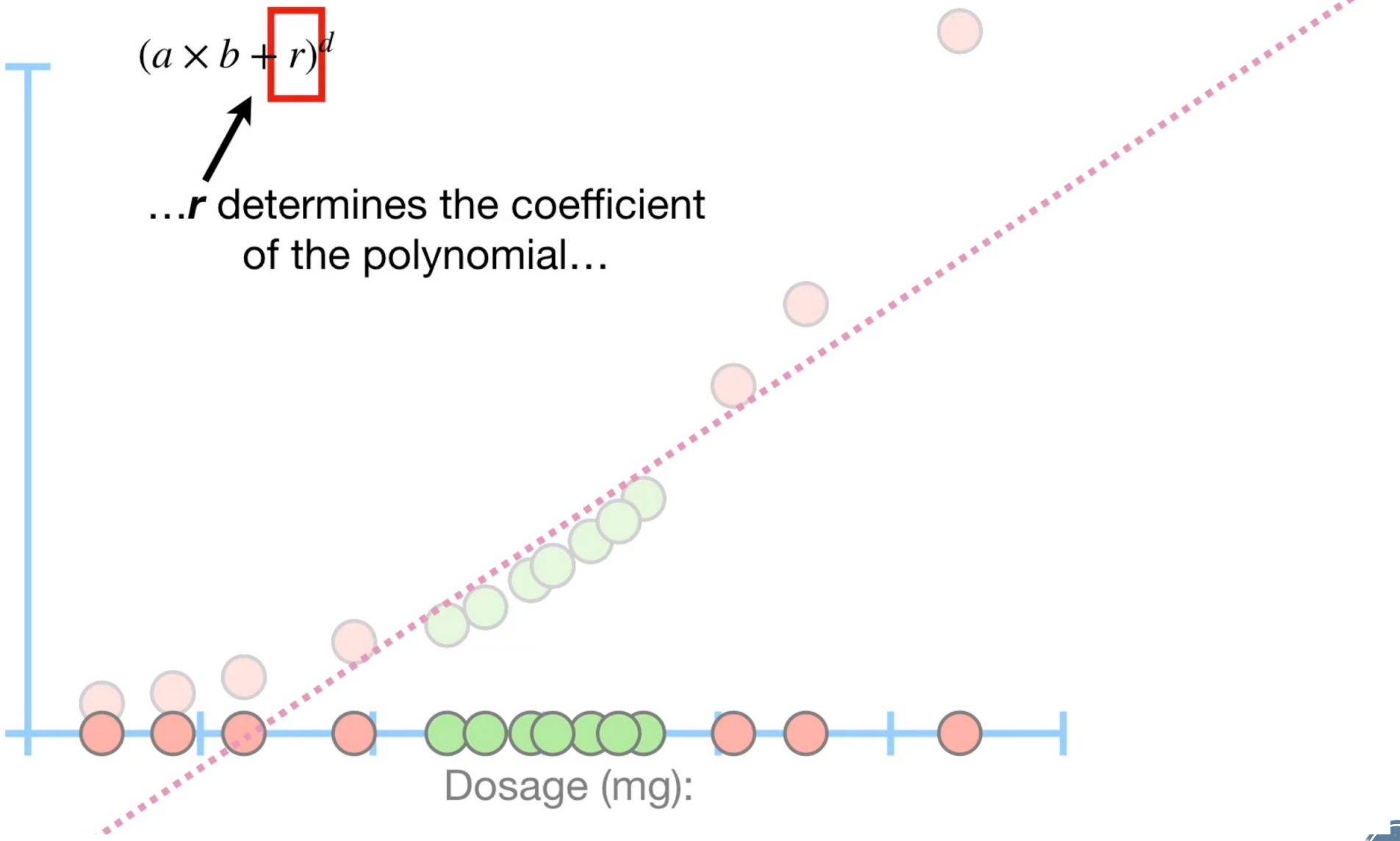

And if we set **d = 3**, then we would get a **3rd dimension** based **Dosages³**…
y-axis
z-axis
Dosage (mg):


In summary, the **Polynomial Kernel** systematically increases dimensions by setting ***d***, the degree of the polynomial…
***d* = 3**
y-axis
z-axis
Dosage (mg):


So we used a **Support Vector Machine** with a **Polynomial Kernel** to compute the relationships between the observations in a higher dimension…
y-axis
Dosage²
Dosage (mg):


$(a \times b + r)^d$
The **Polynomial Kernel** that I used looks like this.
Dosage (mg):

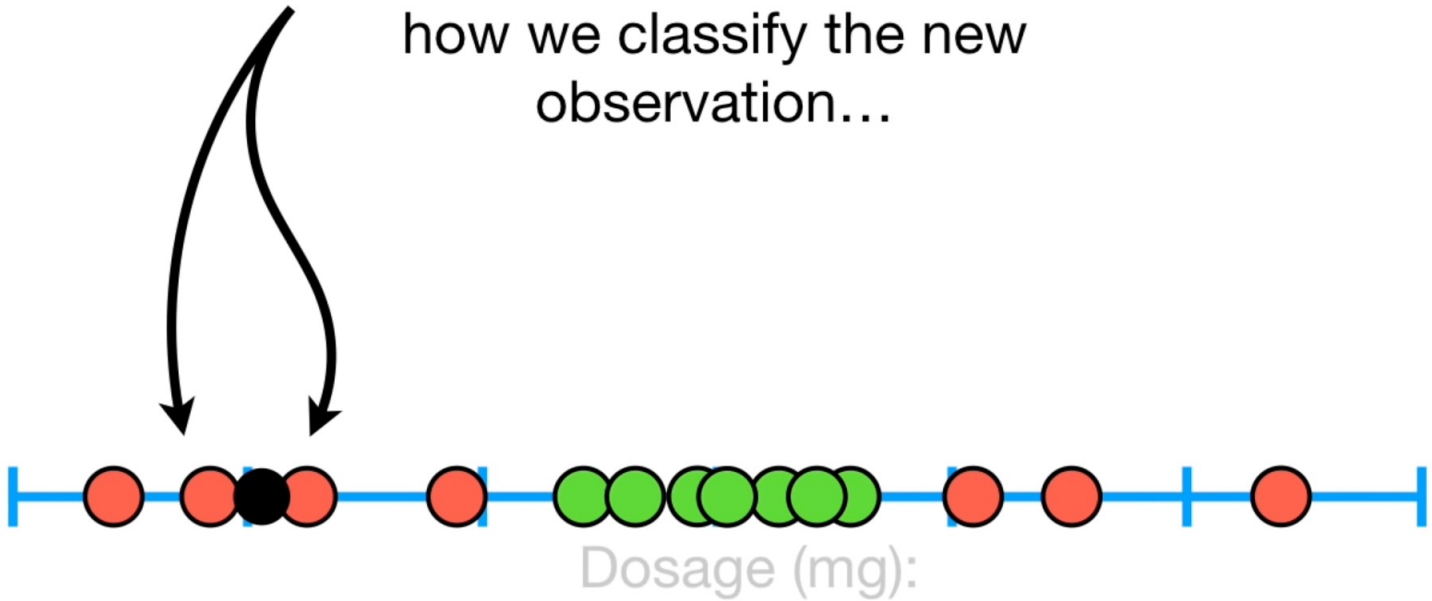

$(a \times b + r)^d$
a and **b** refer to two different observations in the dataset.
Dosage (mg):

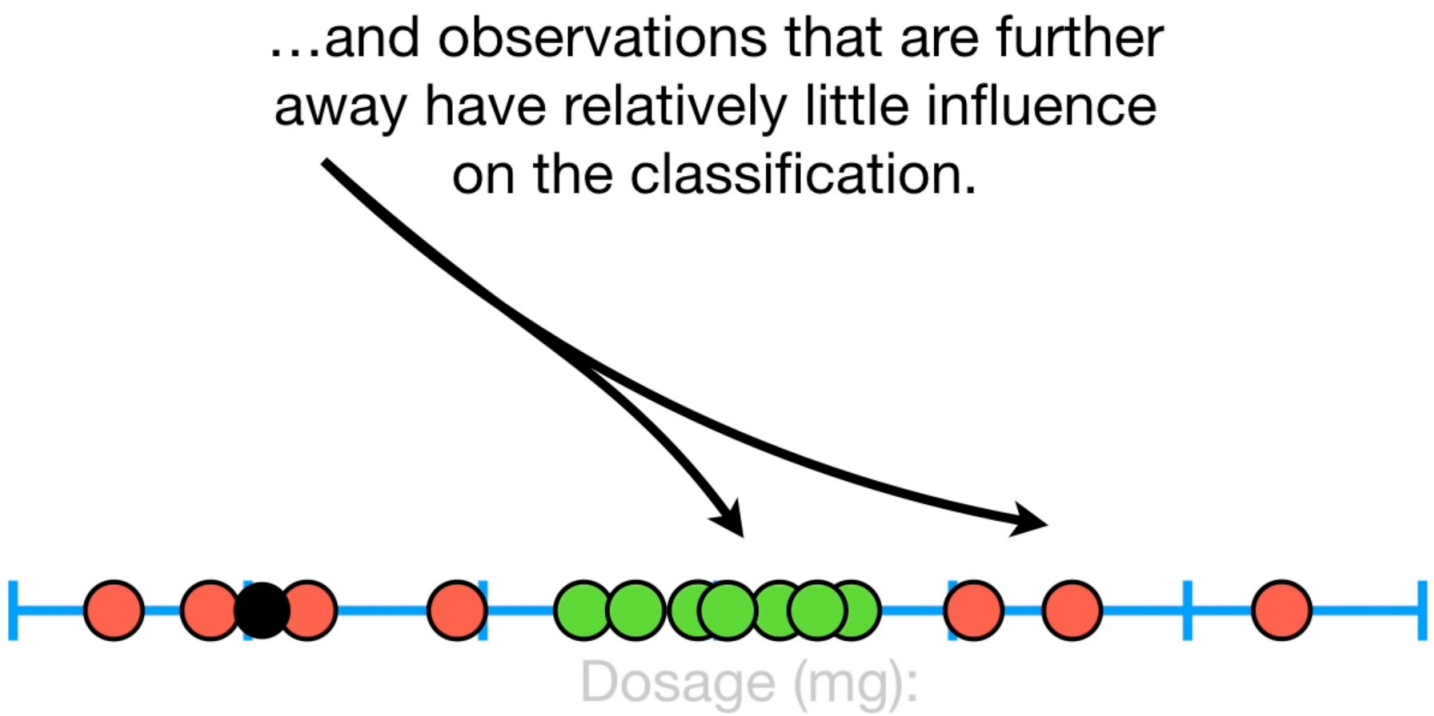

$(a \times b + r)^d$
…*r* determines the coefficient of the polynomial…
Dosage (mg):

Another very commonly used **Kernel** is the **Radial Kernel**, also known as the **Radial Basis Function (RBF) Kernel**.

…the **Radial Kernel** behaves like a
Weighted Nearest Neighbor model.

In other words, the closest observations (aka the nearest neighbors) have a lot of influence on how we classify the new observation…




…and observations that are further
away have relatively little influence
on the classification.
Dosage (mg):

…how the **Radial Kernel** calculates high-dimensional relationships…