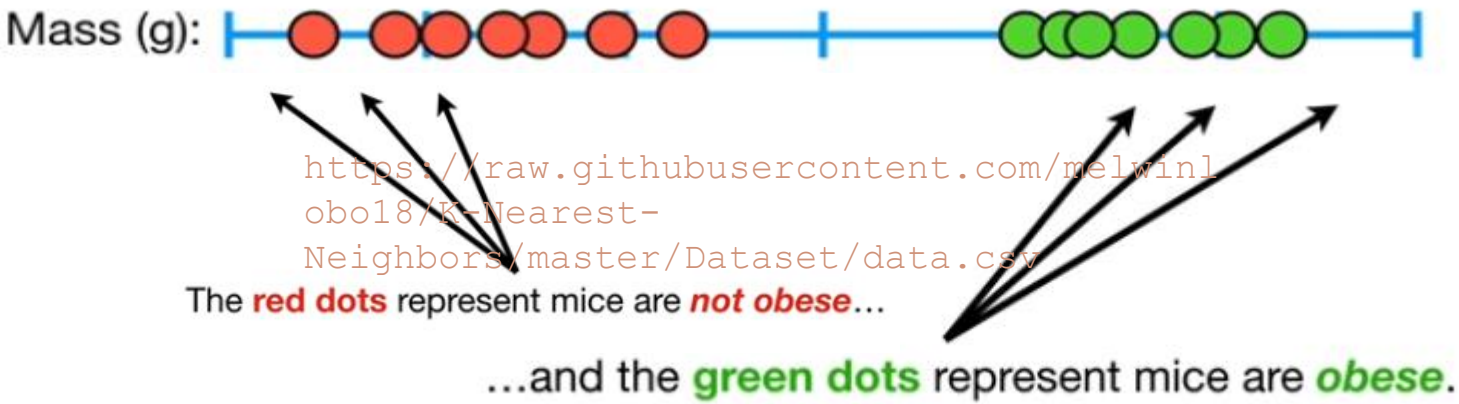
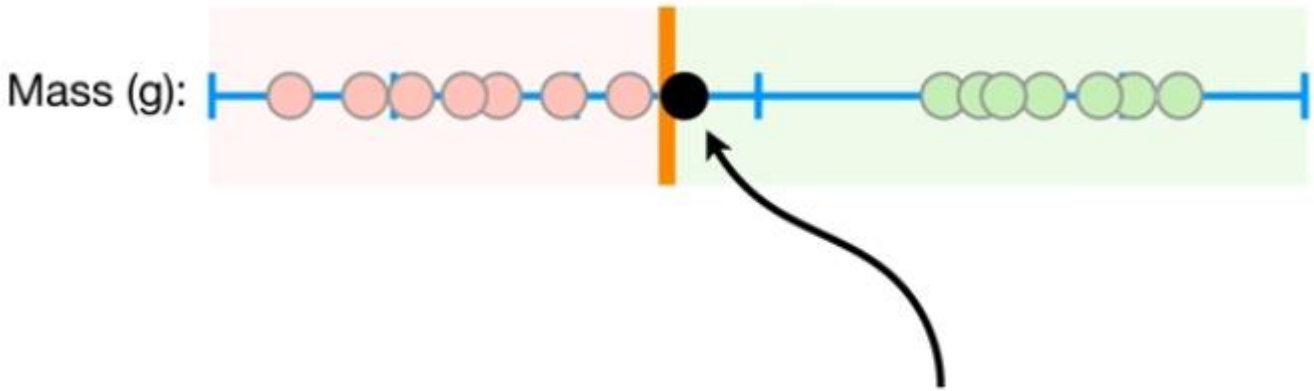
Support Vector Machines

Amirhossein Abaskohi
University of Tehran ACM Summer School 2021

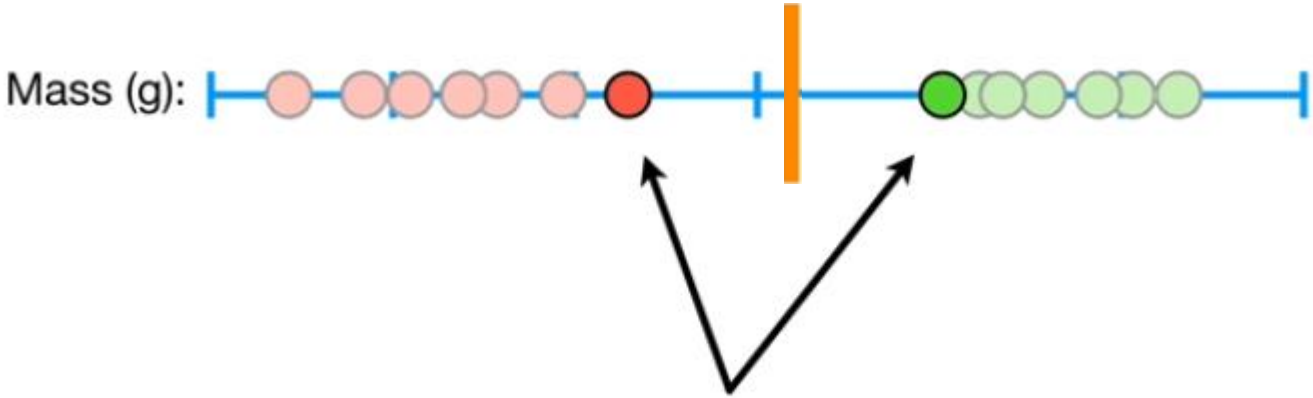


https://raw.githubusercontent.com/melwinlobo18/K-Nearest-Neighbors/master/Dataset/data.csv



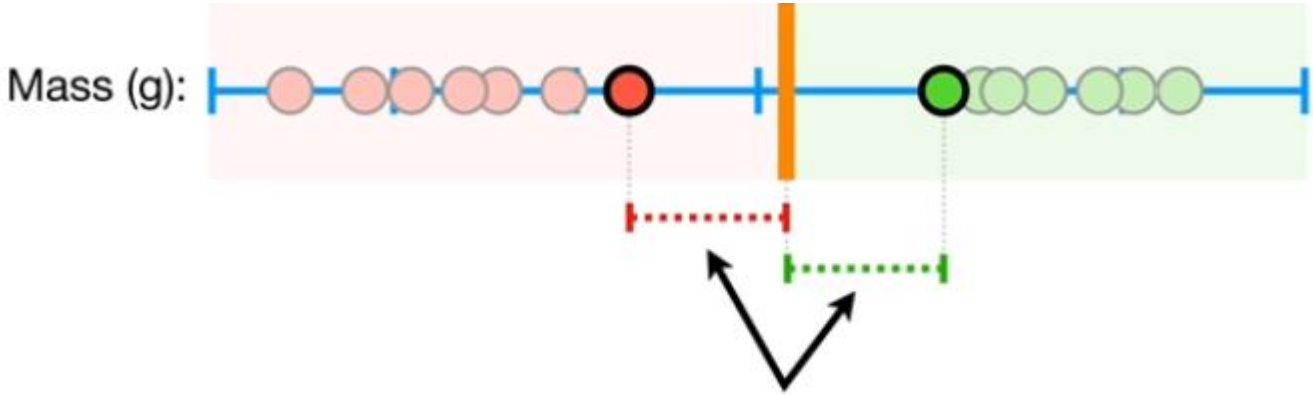
However, what if get a new observation here?

So this threshold is pretty lame.



…we can focus on the observations on the edges of each cluster…

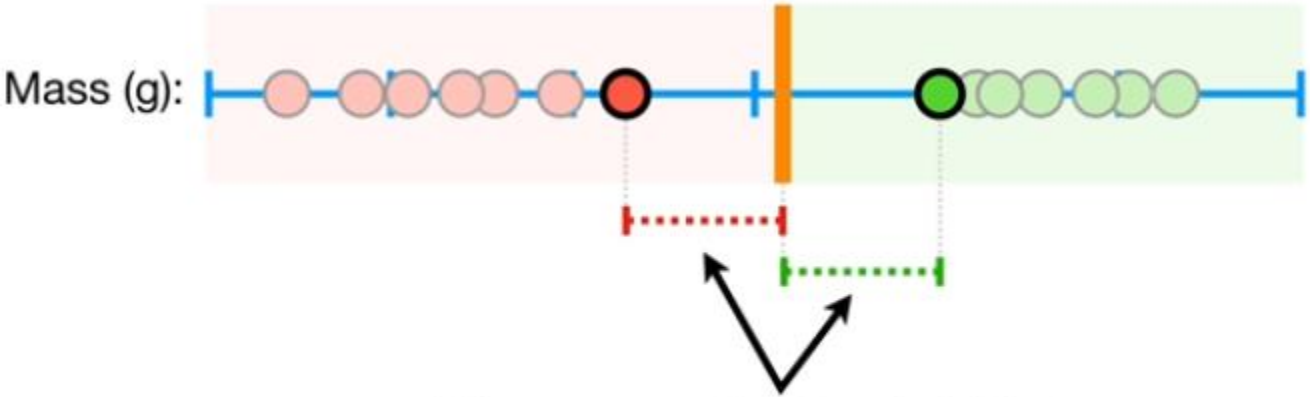
…and use the midpoint between them as the threshold.



The shortest distance between the observations and the threshold is called the **margin**.

Since we put the threshold halfway between these two observations…

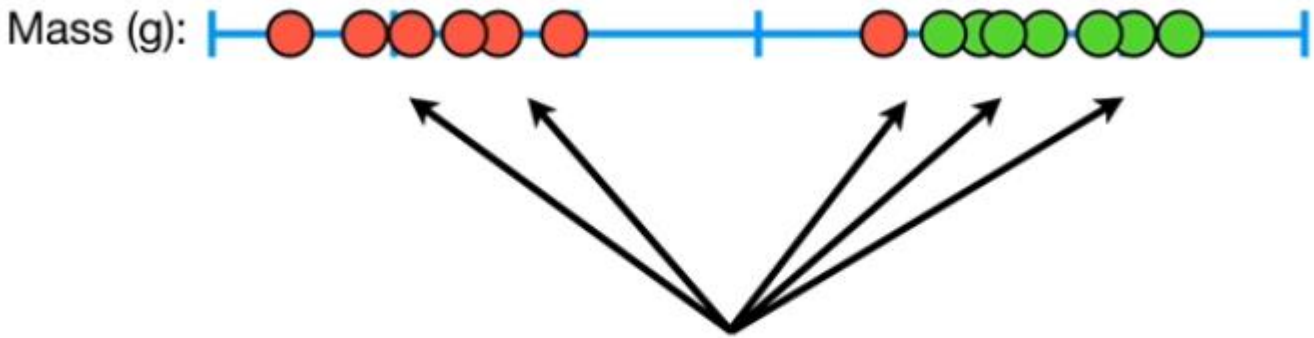
…the distances between the observations and the threshold are the same and both reflect the **margin**.



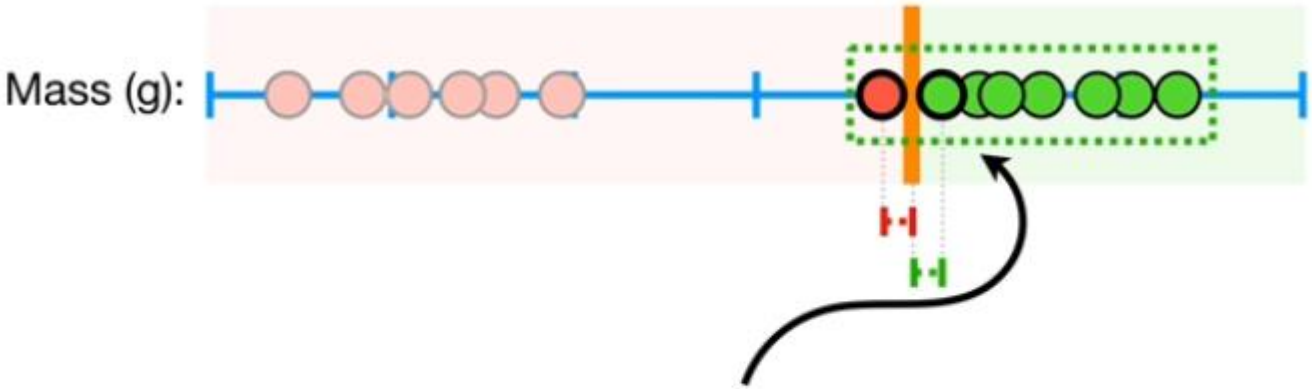
When we use the threshold that gives us the largest **margin** to make classifications…

…we are using a **Maximal Margin Classifier**.

Maximal Margin Classifiers
seem pretty cool…

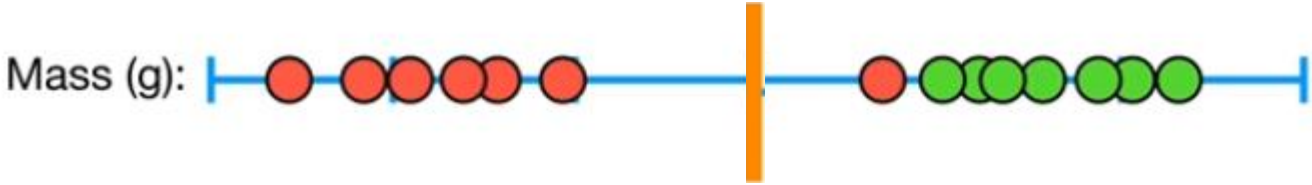


…but what if our training data
looked like this….

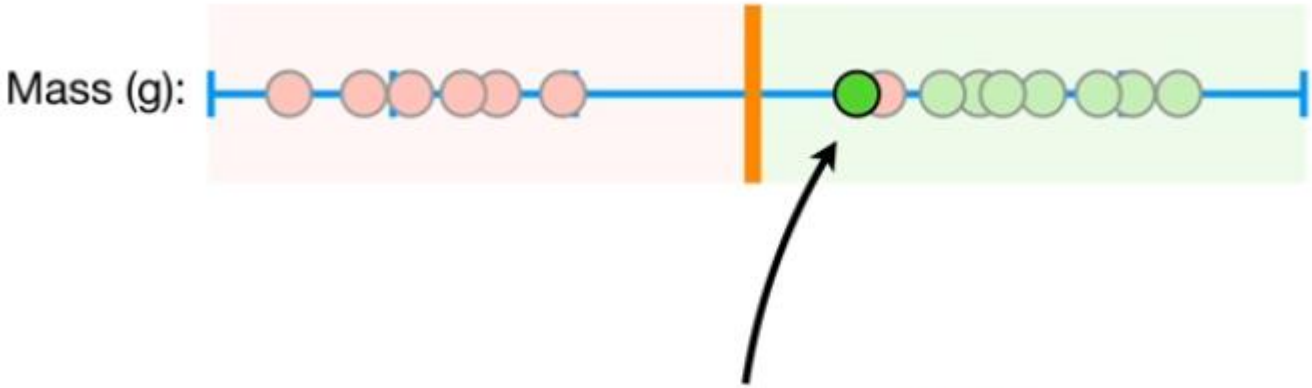


In this case, the **Maximum Margin Classifier** would be super close to the ***obese*** observations…

…and really far from the majority of the observations that are **not obese**.



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



…we will classify it as ***obese***…

…and that makes sense because it is closer to most of the ***obese*** observations.

Choosing a threshold that allows misclassifications is an example of the **Bias/Variance Tradeoff** that plagues all of machine learning.

What is bias-variance trade off?

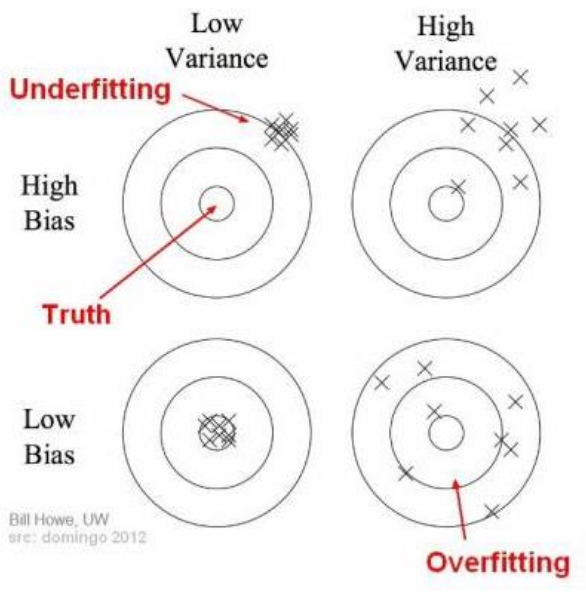
What is bias?

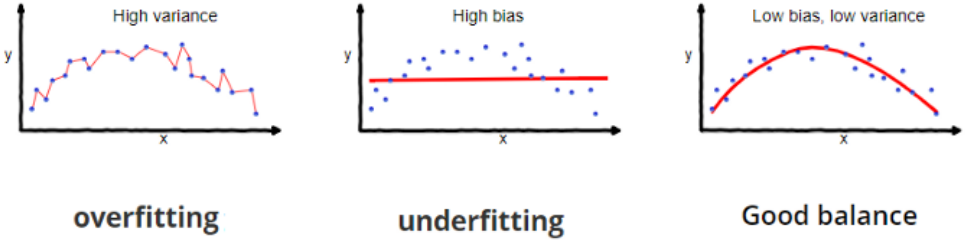
Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.

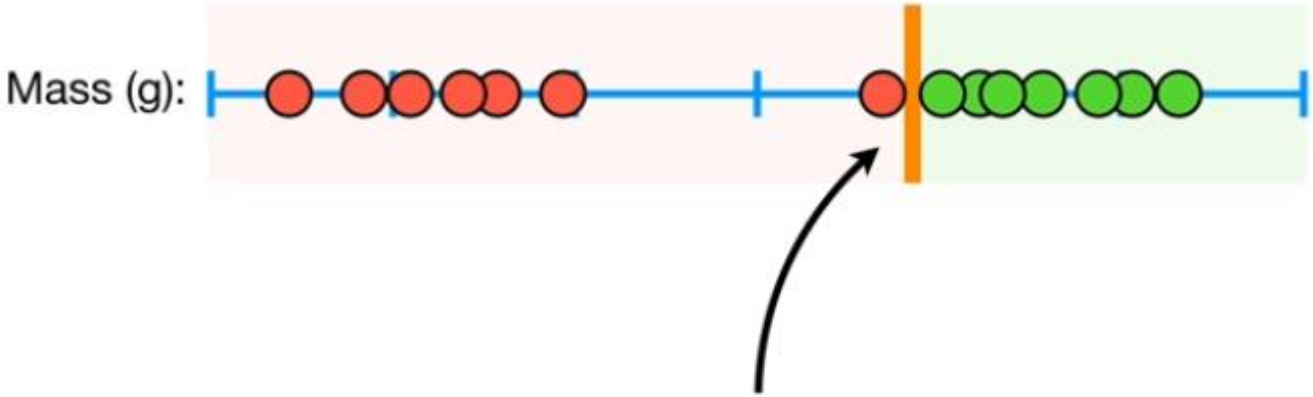
What is variance?

Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn’t seen before. As a result, such models perform very well on training data but has high error rates on test data.

What is bias-variance trade off?

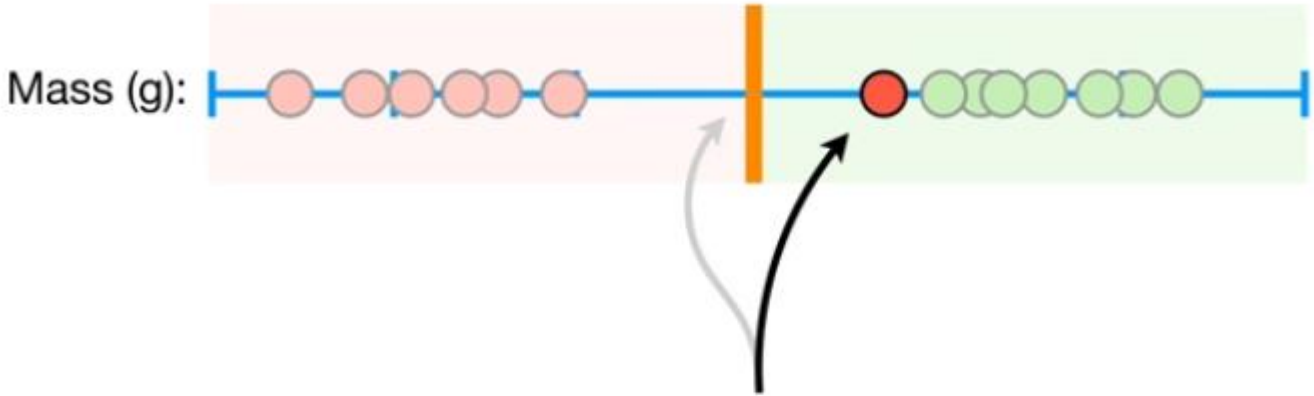






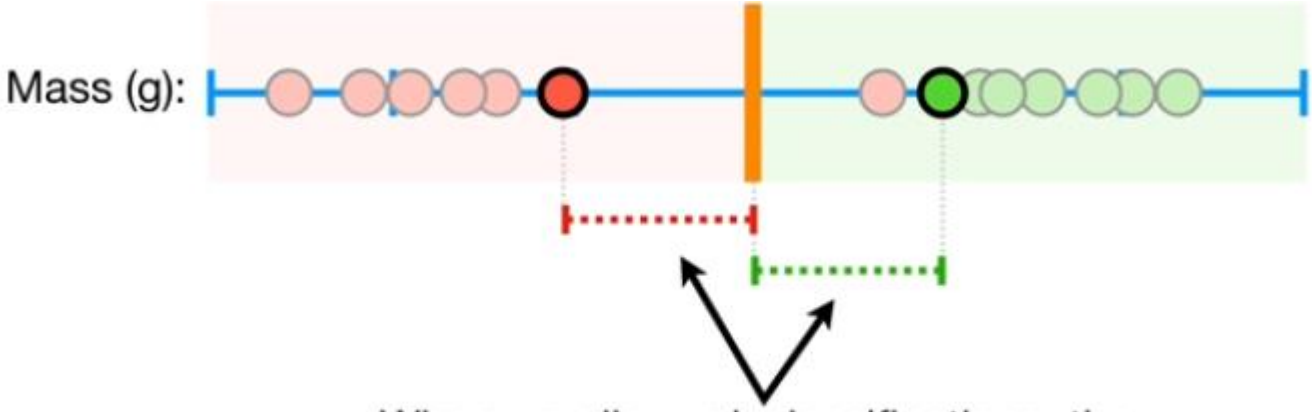
In other words, before we allowed misclassifications, we picked a threshold that was very sensitive to the training data (low bias)…

…and it performed poorly when we got new data (high variance).

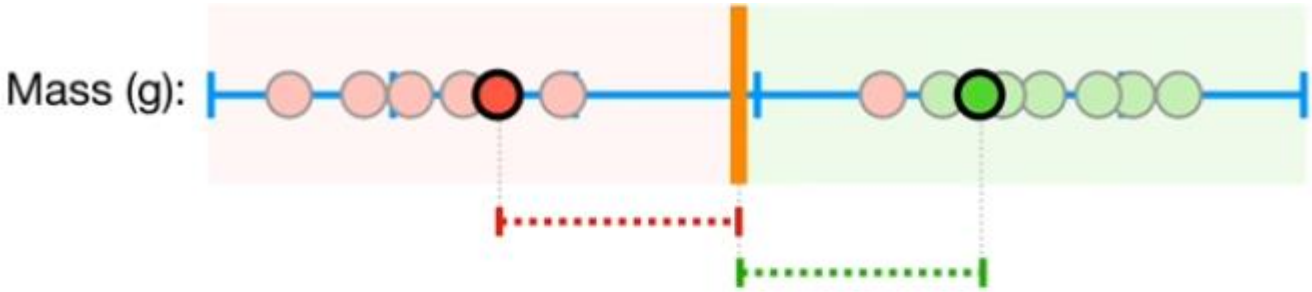


In contrast, when we picked a threshold that was less sensitive to the training data and allowed misclassifications (higher bias)…

…it performed better when we got new data (low variance).



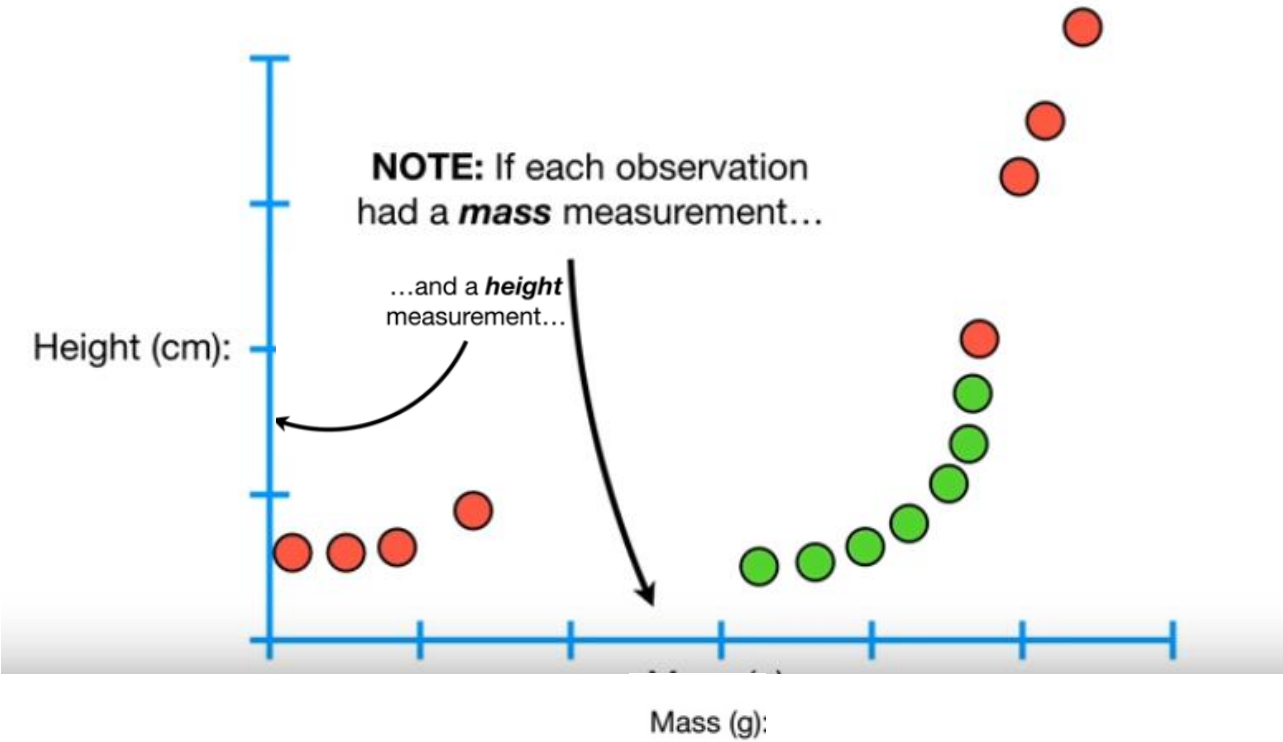
When we allow misclassifications, the distance between the observations and the threshold is called a **Soft Margin**.

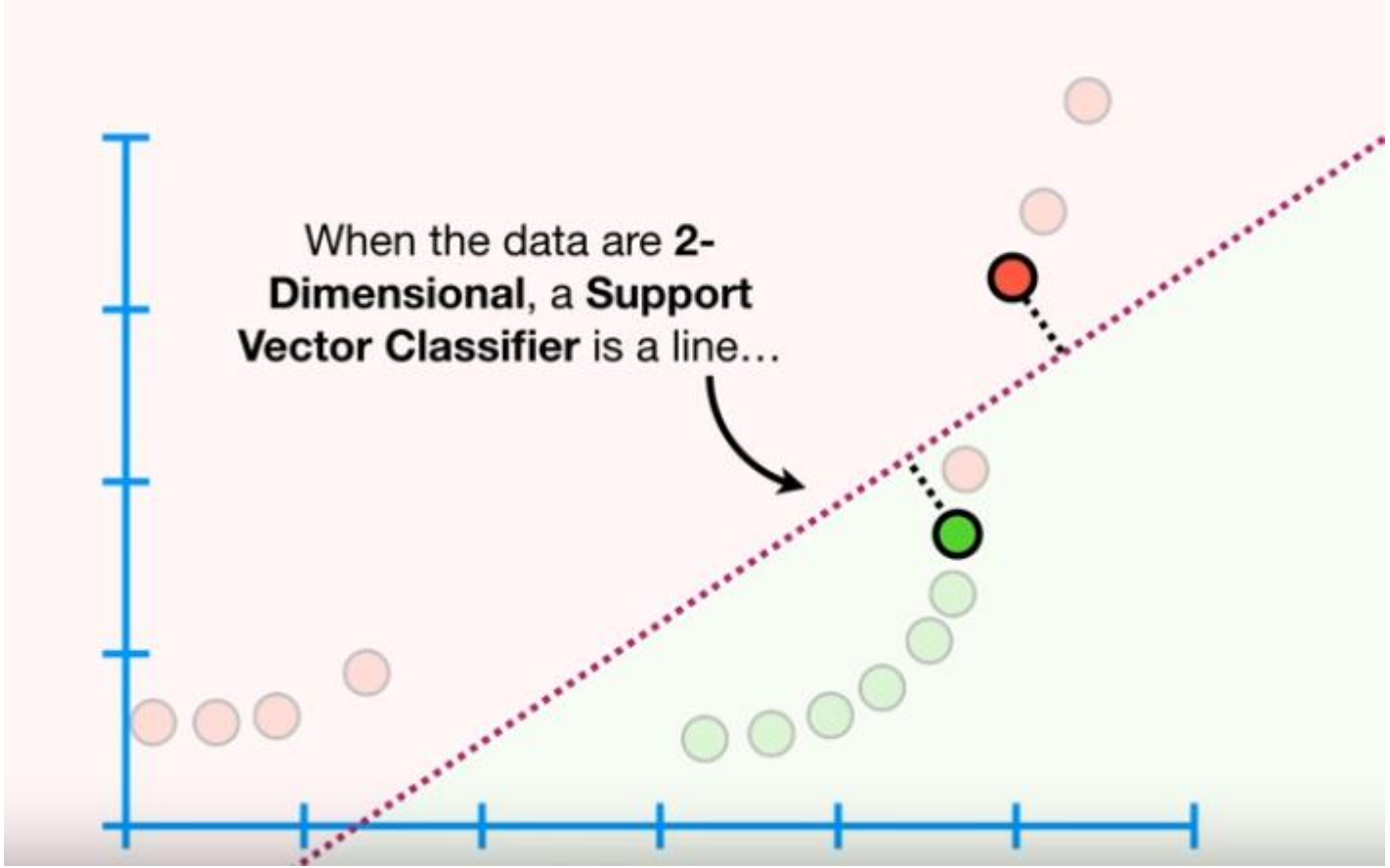


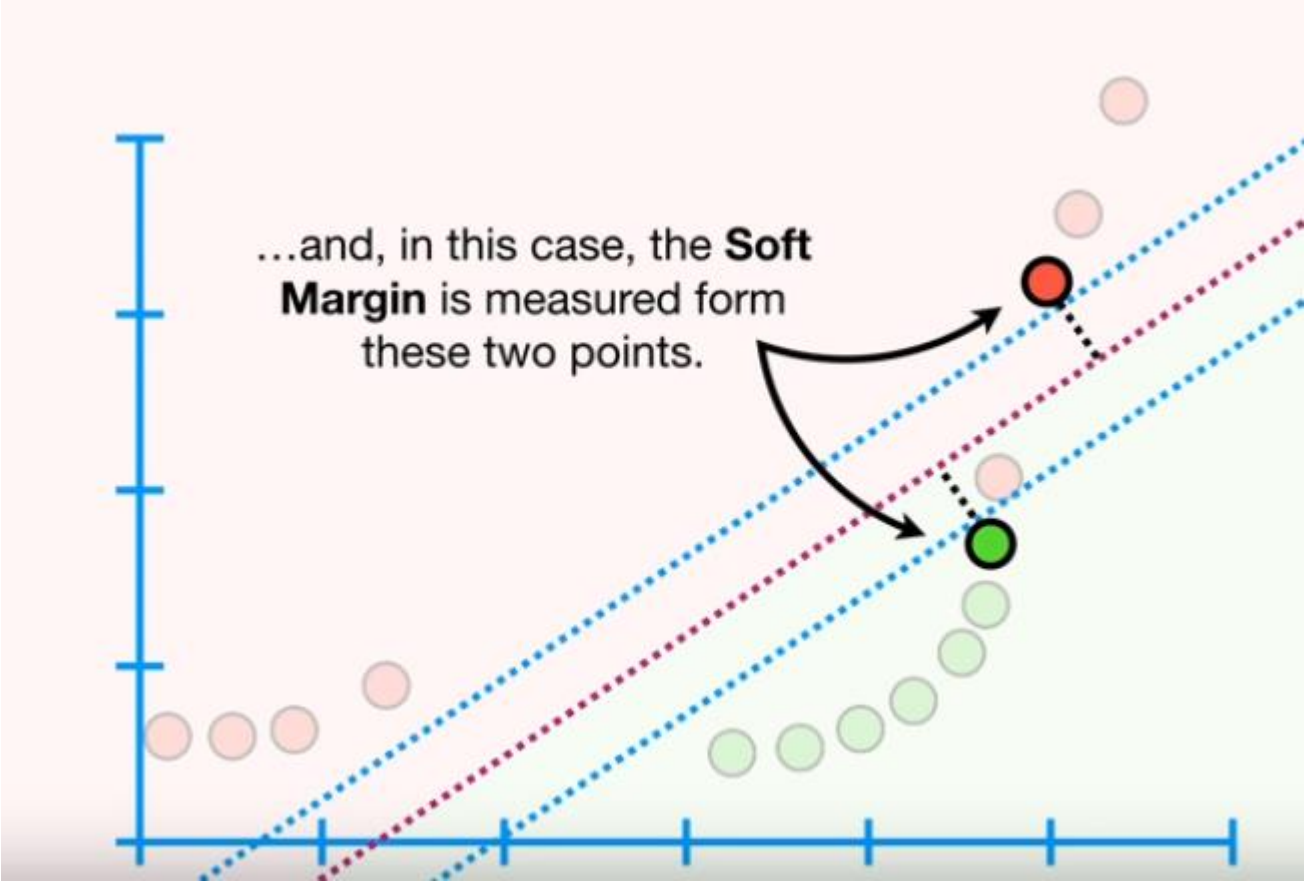
When we use a **Soft Margin** to determine the location of a threshold…

…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**.


NOTE: If each observation had a *mass* measurement…
…and a *height* measurement…
Height (cm):
Mass (g):

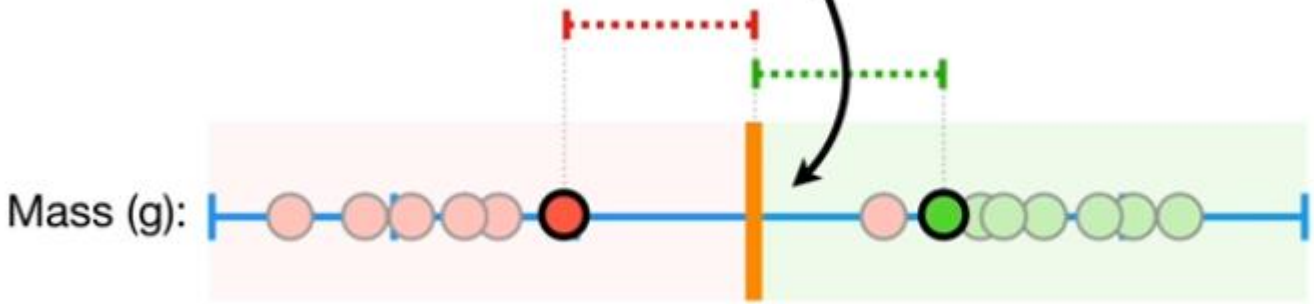

When the data are **2-Dimensional**, a **Support Vector Classifier** is a line…


…and, in this case, the **Soft Margin** is measured form these two points.

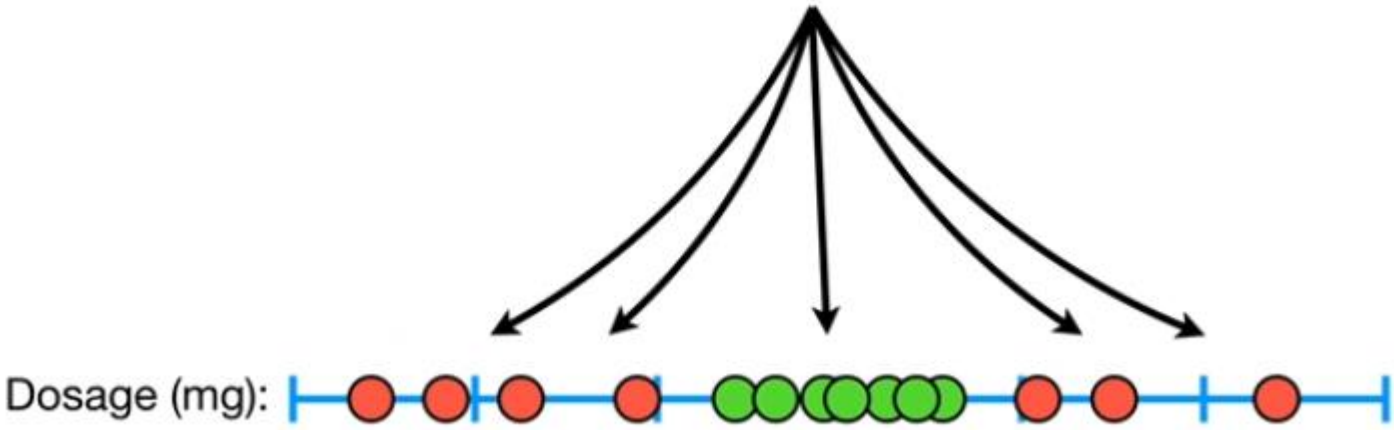
Support Vector Classifiers seem pretty cool because they can handle…

…outliers…

…and, because they allow misclassifications, they can handle overlapping classifications…



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

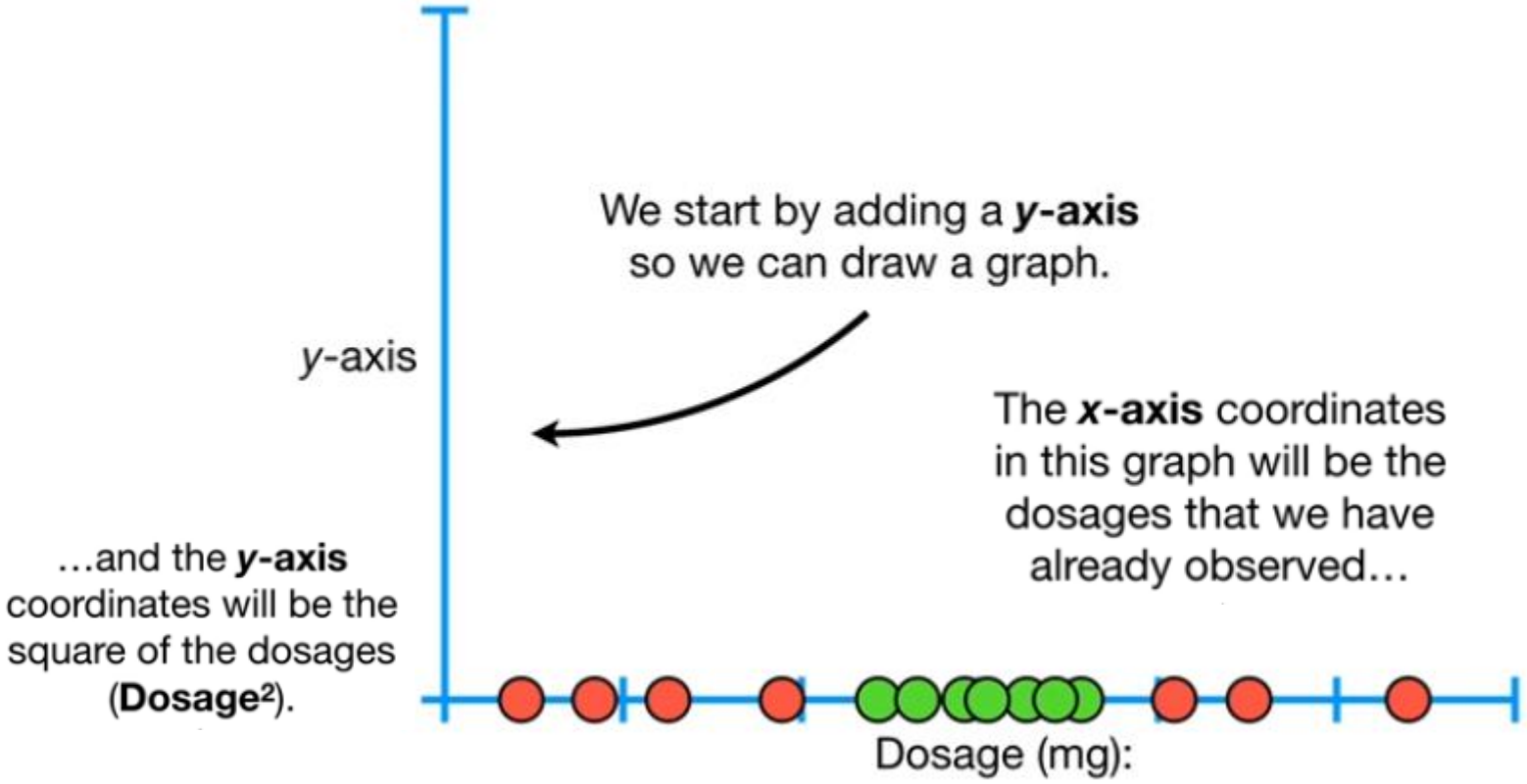


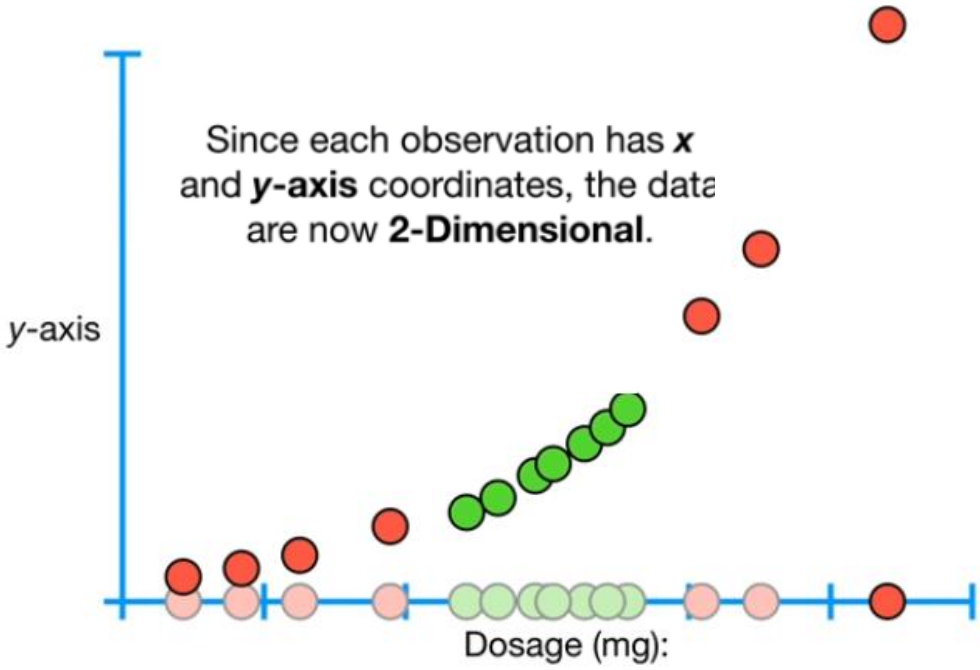
Now, no matter where we put the classifier, we will make a lot of misclassifications.

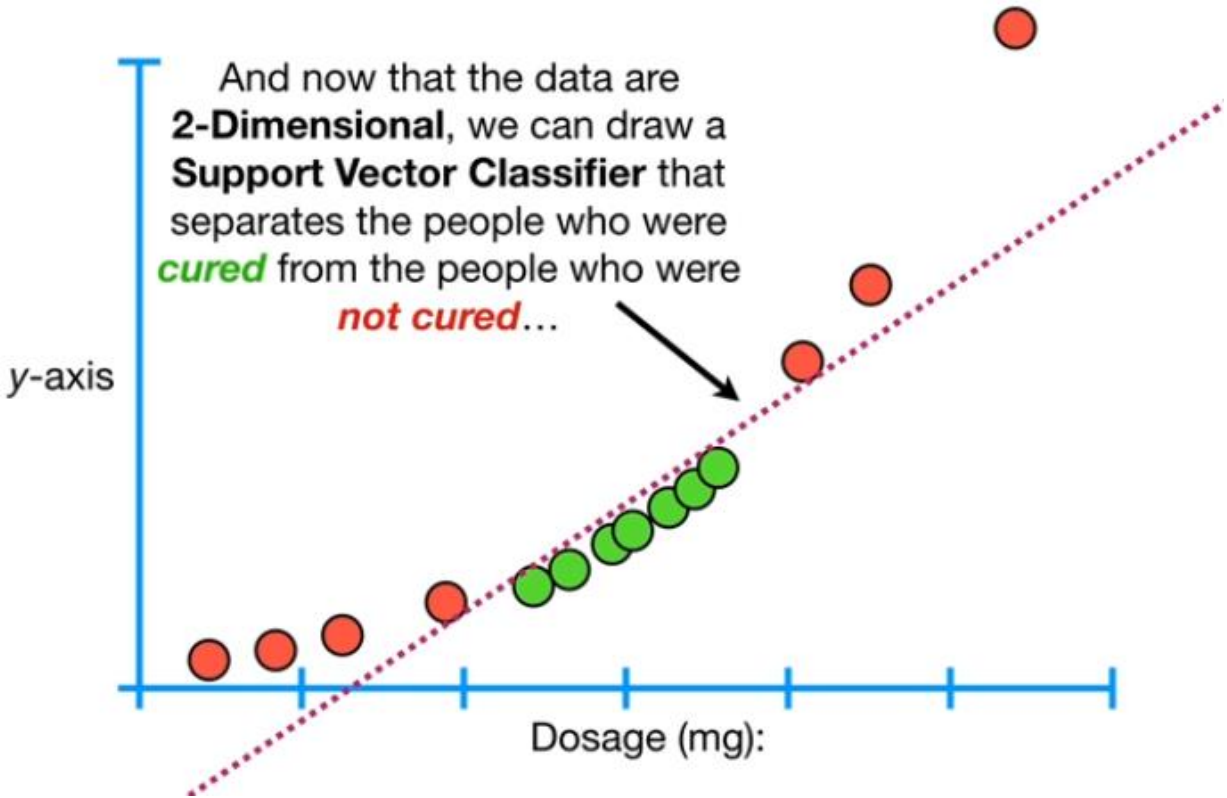
So **Support Vector Classifiers** are are only semi-cool, since they don't perform well with this type of data.

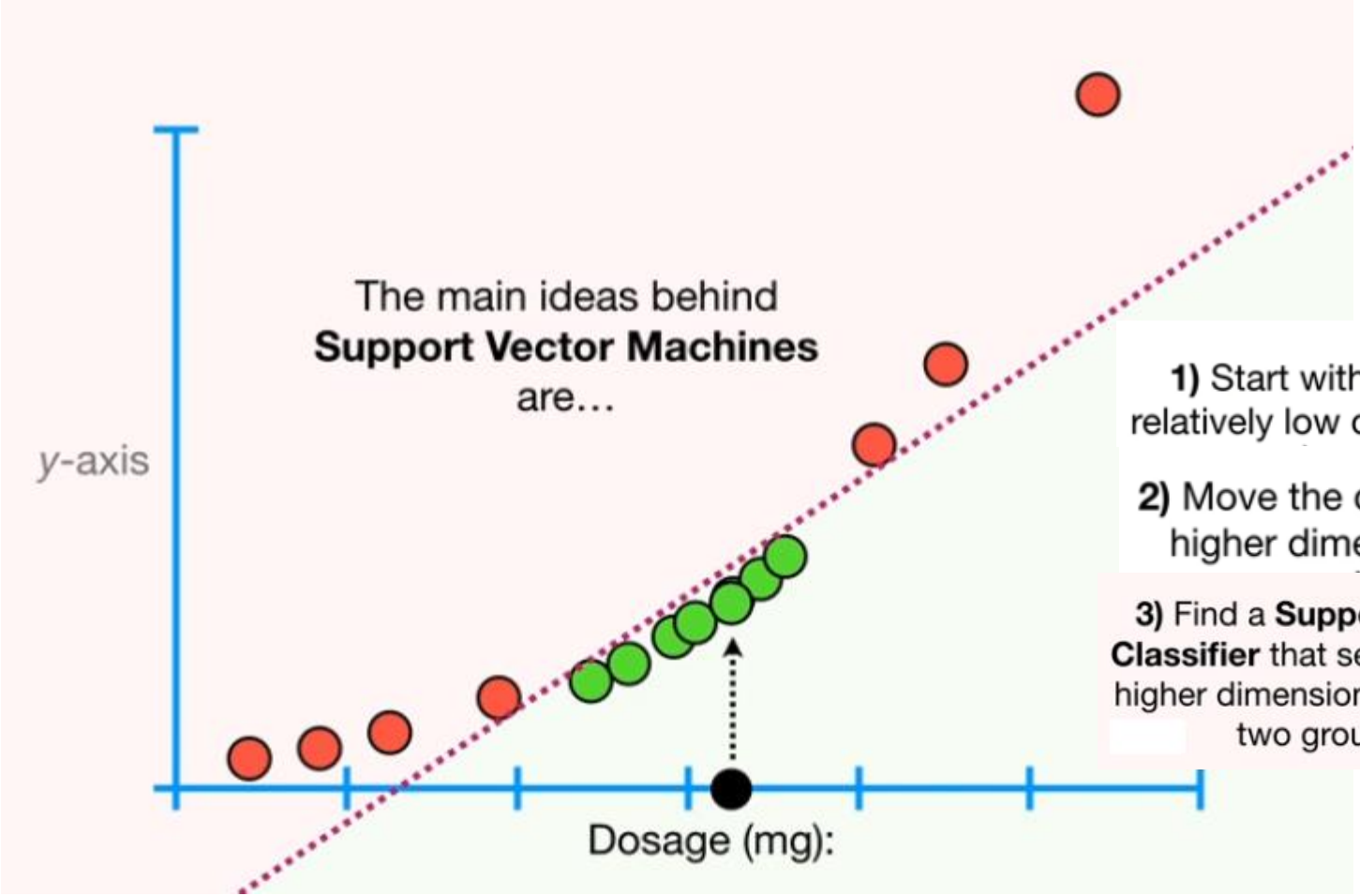
Since **Maximal Margin Classifiers** and **Support Vector Classifiers** can't handle this data, it's high time we talked about…

Support Vector Machines!!!


We start by adding a **y-axis** so we can draw a graph.
y-axis
The **x-axis** coordinates in this graph will be the dosages that we have already observed…
…and the **y-axis** coordinates will be the square of the dosages (**Dosage²**).
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):



1) Start with data in a relatively low dimension…

2) Move the data into a higher dimension…

3) Find a **Support Vector Classifier** that separates the higher dimensional data into two groups.

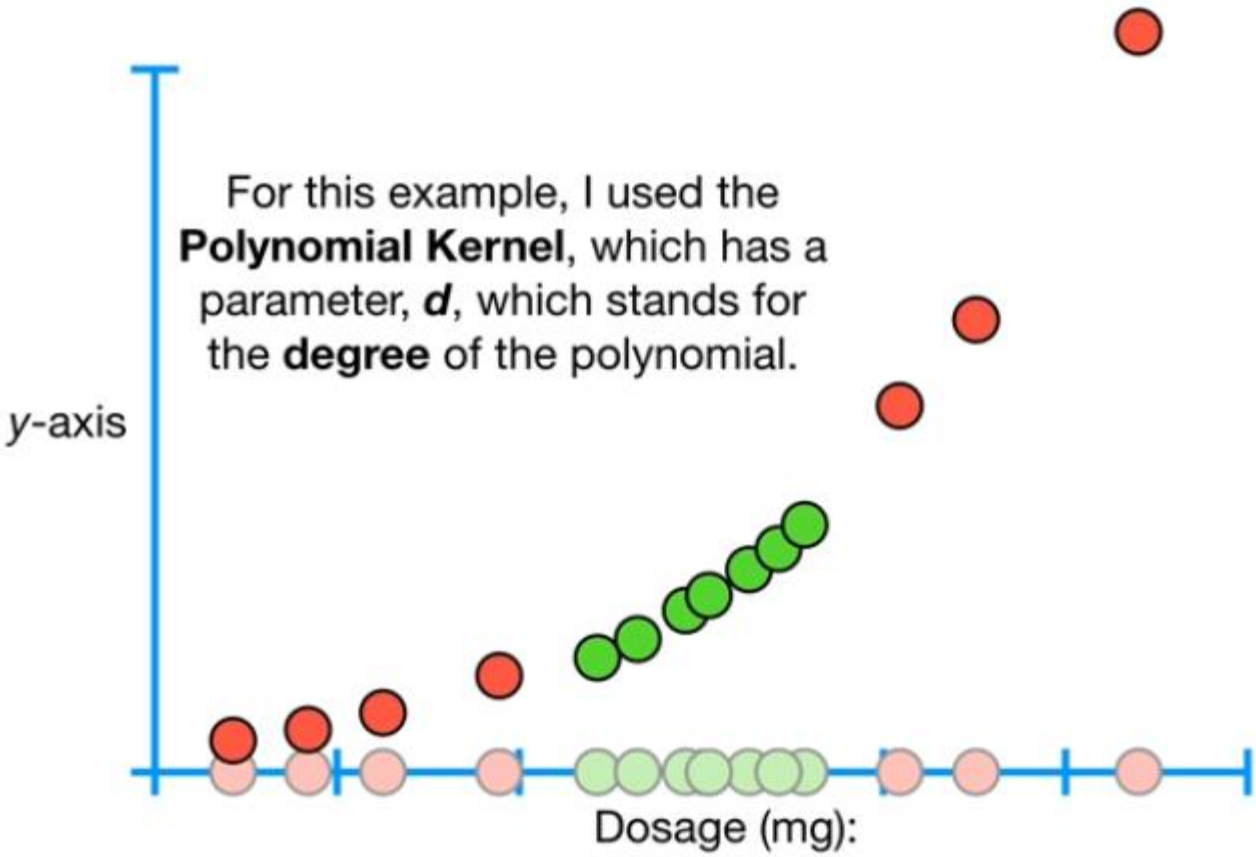
…you may be wondering why we decided to create ***y*-axis** coordinates with **Dosage²**.

Why not **Dosage³**?

…or $\frac{\pi}{4} \times \sqrt{\textbf{Dosage}}$

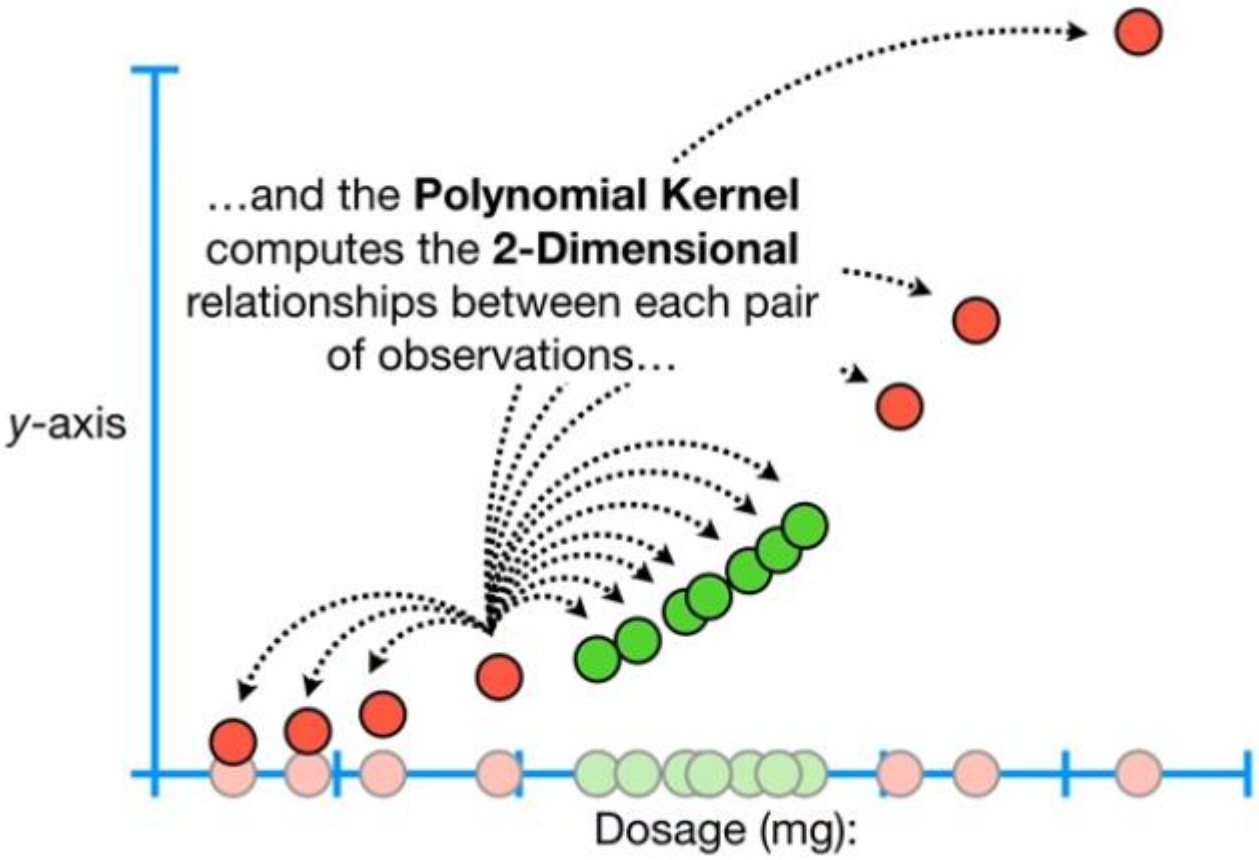
In other words, how do we decide how to transform the data?

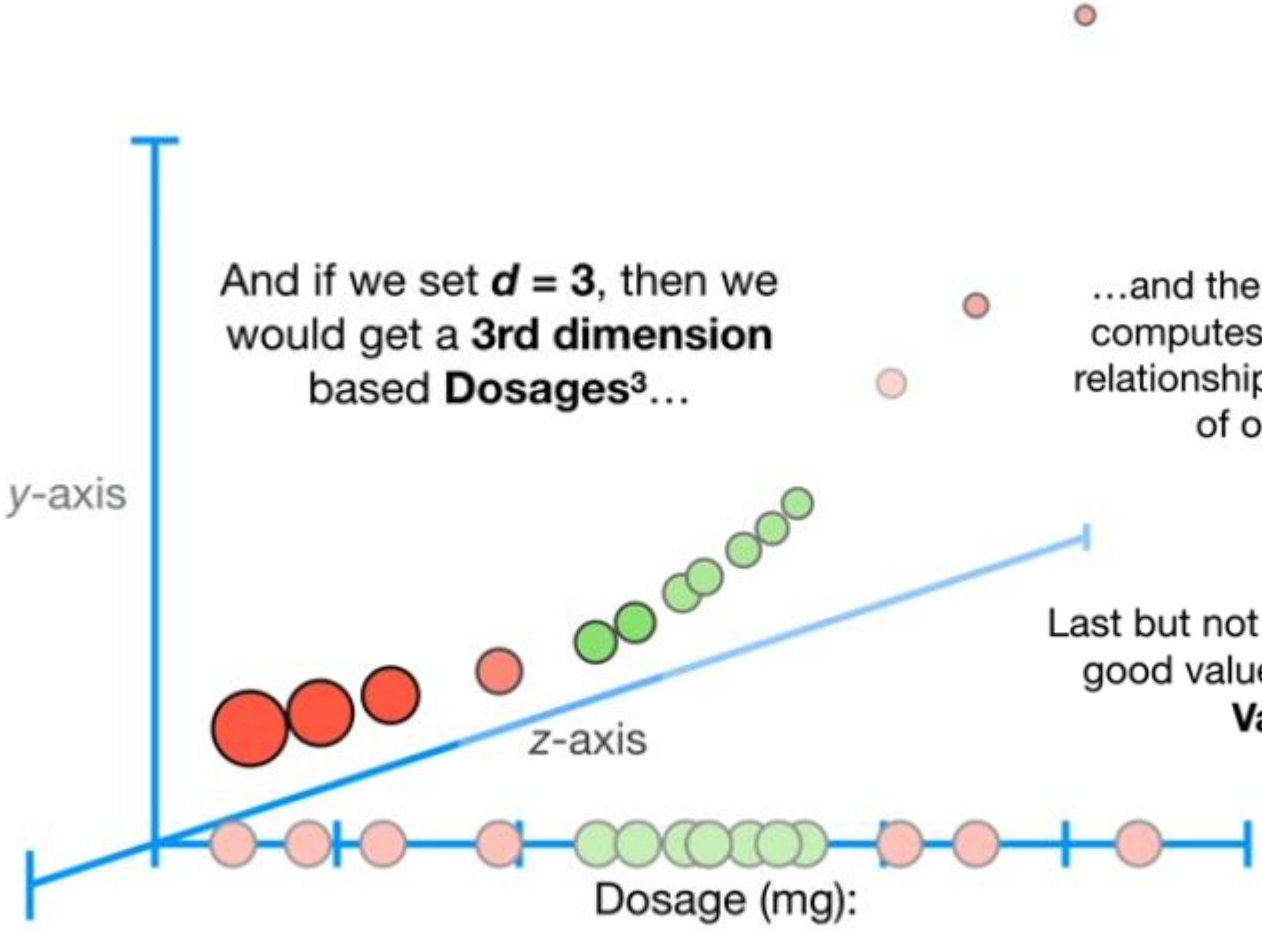
In order to make the mathematics possible, **Support Vector Machines** use something called **Kernel Functions** to *systematically* find **Support Vector Classifiers** in higher dimensions.


For this example, I used the **Polynomial Kernel**, which has a parameter, ***d***, which stands for the **degree** of the polynomial.
y-axis
Dosage (mg):

When ***d* = 1**, the **Polynomial Kernel** computes the relationships between each pair of observations in **1-Dimension**…

…and these relationships are used to find a **Support Vector Classifier**.

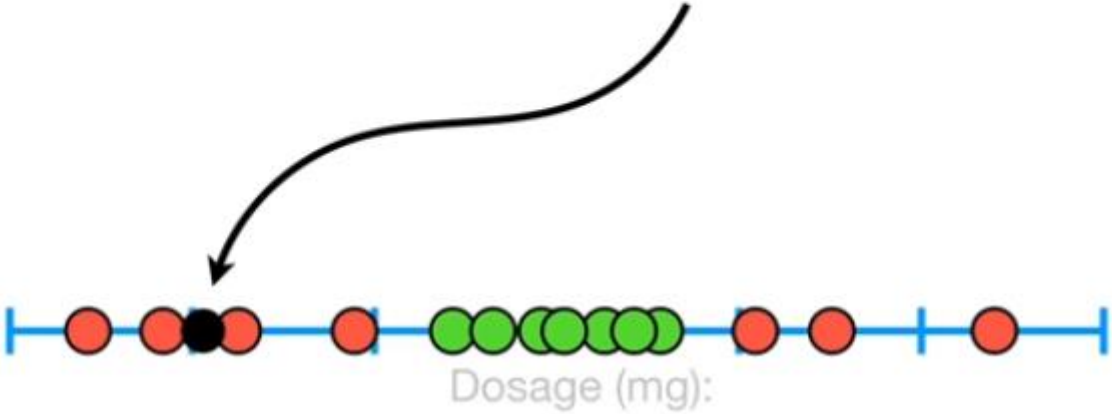

…and the **Polynomial Kernel** computes the **2-Dimensional** relationships between each pair of observations…
y-axis
Dosage (mg):


And if we set **d = 3**, then we would get a **3rd dimension** based **Dosages³**…
…and the **Polynomial Kernel** computes the **3-Dimensional** relationships between each pair of observations…
y-axis
z-axis
Last but not least, we can find a good value for ***d*** with **Cross Validation**.
Dosage (mg):

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

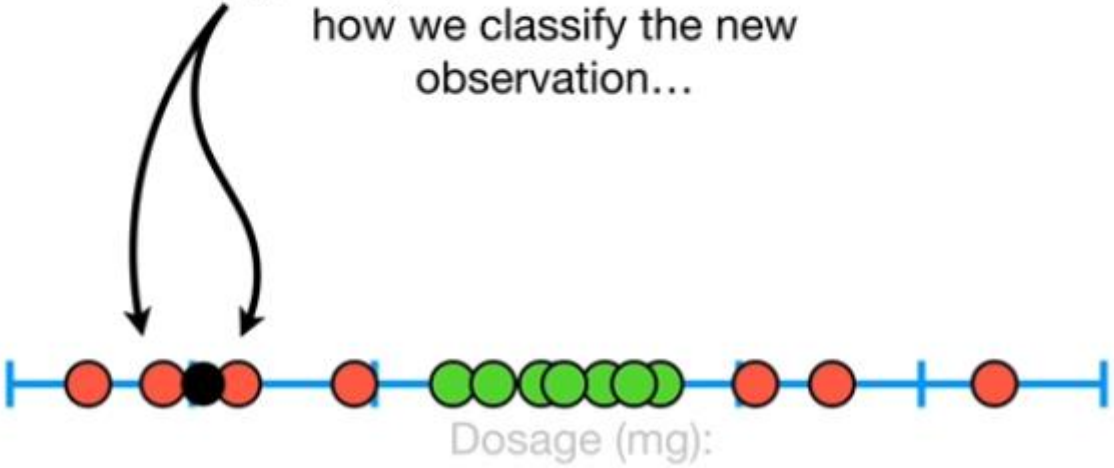
Unfortunately, the **Radial Kernel** finds **Support Vector Classifiers** in *infinite dimensions*, so I can't give you an example of what it does exactly.

However, when using it on a new observation like this…

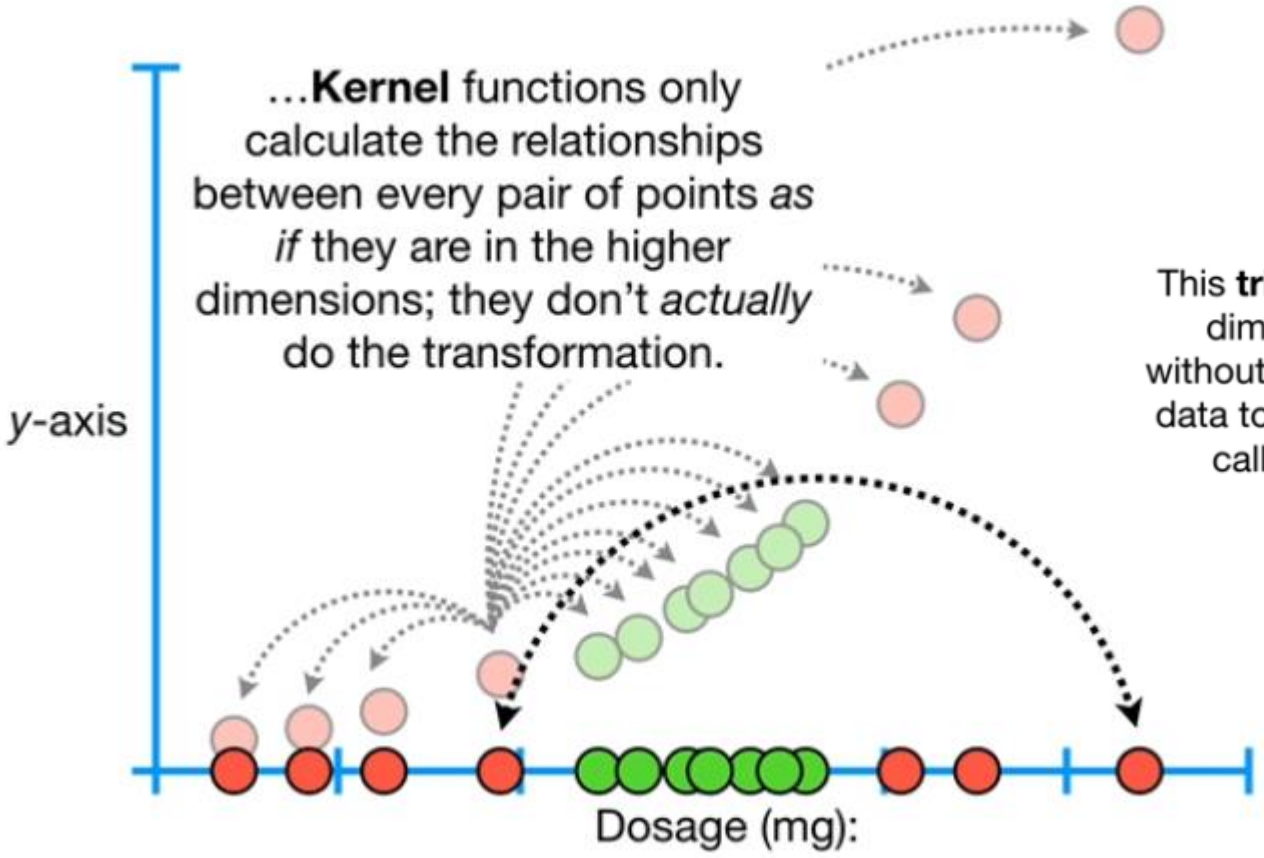


…the **Radial Kernel** behaves like a **Weighted Nearest Neighbor** model.

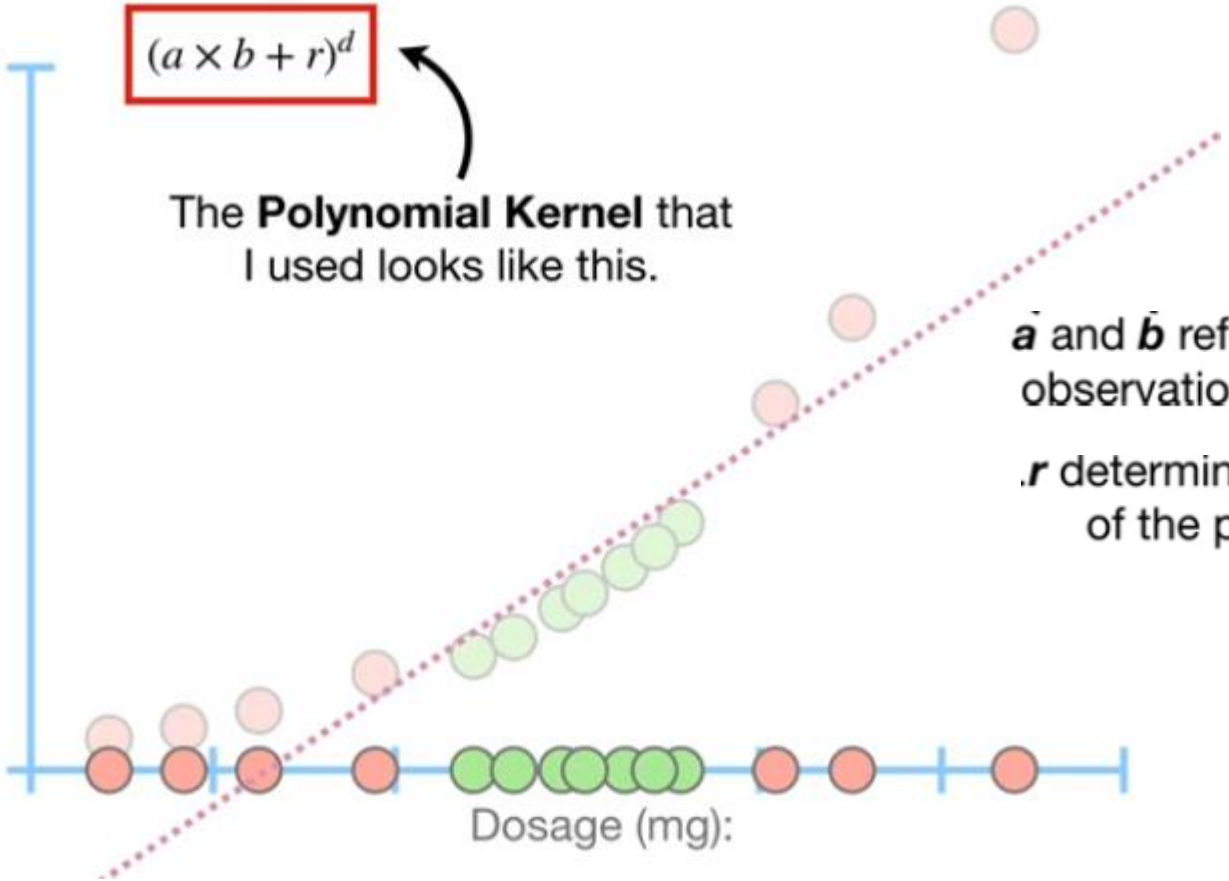
In other words, the closest observations (aka the nearest neighbors) have a lot of influence on how we classify the new observation…



…and observations that are further away have relatively little influence on the classification.



This **trick**, calculating the high-dimensional relationships without actually transforming the data to the higher dimension, is called **The Kernel Trick**.



a and ***b*** refer to two different observations in the dataset.

r determines the coefficient of the polynomial…

$$
\begin{aligned}
(a \times b + \frac{1}{2})^2 &= (a \times b + \frac{1}{2})(a \times b + \frac{1}{2}) \\
&= ab + a^2b^2 + \frac{1}{4} \\
&= (a, a^2, \frac{1}{2}) \cdot (b, b^2, \frac{1}{2})
\end{aligned}
$$

The **Dot Product** gives us the high-dimensional coordinates for the data.

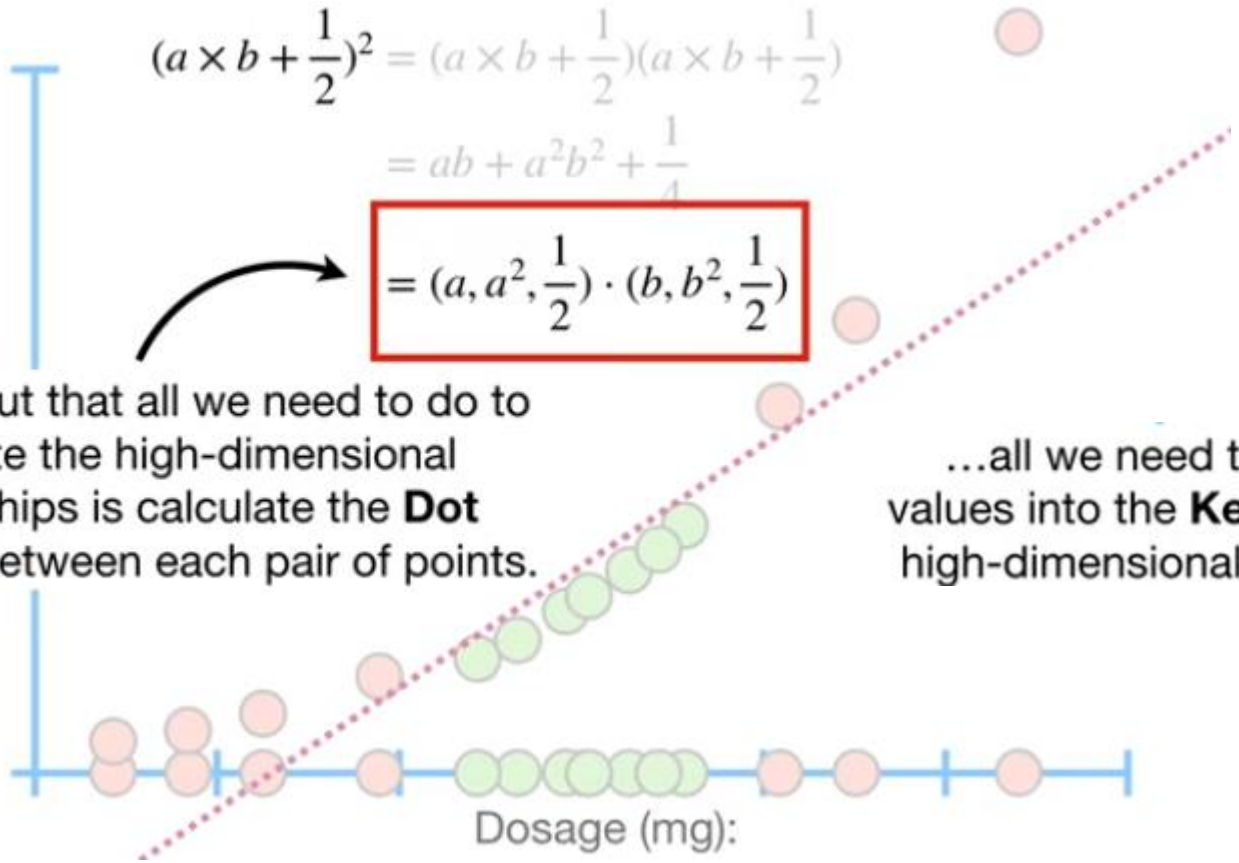
$$(a \times b + \frac{1}{2})^2 = (a \times b + \frac{1}{2})(a \times b + \frac{1}{2})$$

$$= ab + a^2b^2 + \frac{1}{4}$$

$$\boxed{= (a, a^2, \frac{1}{2}) \cdot (b, b^2, \frac{1}{2})}$$

…it turns out that all we need to do to calculate the high-dimensional relationships is calculate the **Dot Products** between each pair of points.

…all we need to do is plug values into the **Kernel** to get the high-dimensional relationships.



One way to deal with overlapping data is to use a **Support Vector Machine** with a **Radial Kernel**

$$e^{-\gamma(a-b)^2}$$

Because the **Radial Kernel** finds **Support Vector Classifiers** in infinite dimensions, it's not possible to visualize what it does.

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

Now let's talk about how the **Radial Kernel** determines how much influence each observation in the **Training Dataset** has on classifying new observations.

$$e^{-\gamma(a-b)^2}$$

Just like with the **Polynomial Kernel**, ***a*** and ***b*** refer to two different **Dosage** measurements.

$$e^{-\gamma(a-b)^2}$$

γ (gamma), which is determined by **Cross Validation**, scales the squared distance, and thus, it scales the influence.

The difference between the measurements is then squared, giving us the squared distance between the two observations.

Thus, the amount of influence one observation has on another is a function of the squared distance.

NOTE: Just like with the **Polynomial Kernel**, when we plug values into the **Radial Kernel**, we get the high-dimensional relationship.

$$e^{-\gamma(a-b)^2} = \text{high-dimensional relationship}$$

Thus, **0.11** is the high-dimensional relationship between these two observations that are relatively close to each other…

…and **A Number Very Close to Zero** is the high-dimensional relationship between these two observations that are relatively far from each other.

$$e^{-\frac{1}{2}(a-b)^2} = e^{-\frac{1}{2}(a^2+b^2-2ab)} = e^{-\frac{1}{2}(a^2+b^2)}\boxed{e^{ab}}$$

Now let's create the **Taylor Series Expansion** of this last term.

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots + \frac{f^{\infty}(a)}{\infty!}(x-a)^{\infty}$$

$$e^x = e^a + \frac{e^a}{1!}(x-a) + \frac{e^a}{2!}(x-a)^2 + \frac{e^a}{3!}(x-a)^3 + \cdots + \frac{e^a}{\infty!}(x-a)^{\infty}$$

$$e^x = e^0 + \frac{e^0}{1!}(x-0) + \frac{e^0}{2!}(x-0)^2 + \frac{e^0}{3!}(x-0)^3 + \cdots + \frac{e^0}{\infty!}(x-0)^{\infty}$$

$$e^{ab} = 1 + \frac{1}{1!}ab + \frac{1}{2!}(ab)^2 + \frac{1}{3!}(ab)^3 + \cdots + \frac{1}{\infty!}(ab)^\infty$$

…we got a **Dot Product** with coordinates for an infinite number of dimensions.

$$a^0b^0 + a^1b^1 + a^2b^2 + \cdots + a^\infty b^\infty \boxed{= (1,\, a,\, a^2, \ldots,\, a^\infty) \cdot (1,\, b^1,\, b^2, \ldots,\, b^\infty)}$$

$$e^{ab} = (1, \sqrt{\frac{1}{1!}}a, \sqrt{\frac{1}{2!}}a^2, \sqrt{\frac{1}{3!}}a^3, \ldots, \sqrt{\frac{1}{\infty!}}a^\infty) \cdot (1, \sqrt{\frac{1}{1!}}b, \sqrt{\frac{1}{2!}}b^2, \sqrt{\frac{1}{3!}}b^3, \ldots, \sqrt{\frac{1}{\infty!}}b^\infty)$$

$$e^{-\frac{1}{2}(a-b)^2} = (s, s\sqrt{\frac{1}{1!}}a, s\sqrt{\frac{1}{2!}}a^2, \ldots, s\sqrt{\frac{1}{\infty!}}a^\infty) \cdot (s, s\sqrt{\frac{1}{1!}}b, s\sqrt{\frac{1}{2!}}b^2, \ldots, s\sqrt{\frac{1}{\infty!}}b^\infty)$$

$$s = \sqrt{e^{-\frac{1}{2}(a^2+b^2)}}$$