

Support Vector Machines I

Dr. Chelsea Parlett-Pelleriti

Outline

- Maximal Margin Classifiers
- Support Vector Classifiers
- Support Vector Machines: Kernels and The Kernel Trick
- Common Kernels
 - Polynomial
 - Radial Basis Function

Hyperplanes

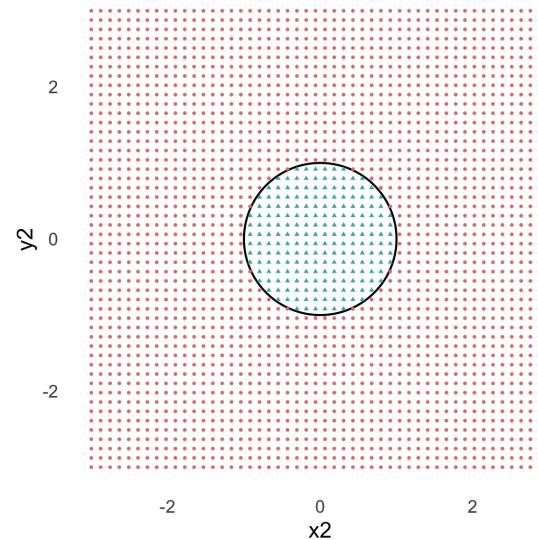
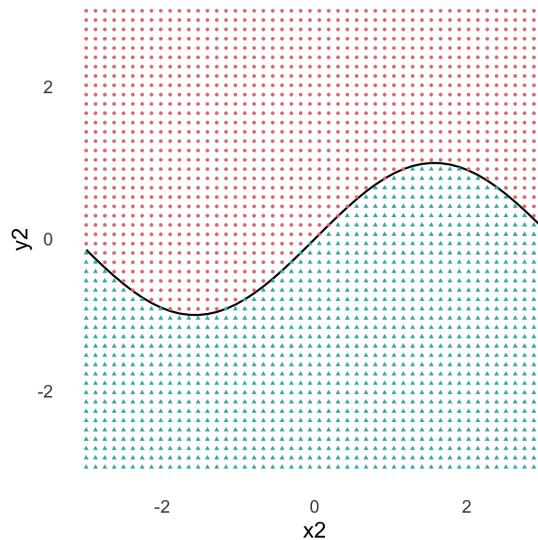
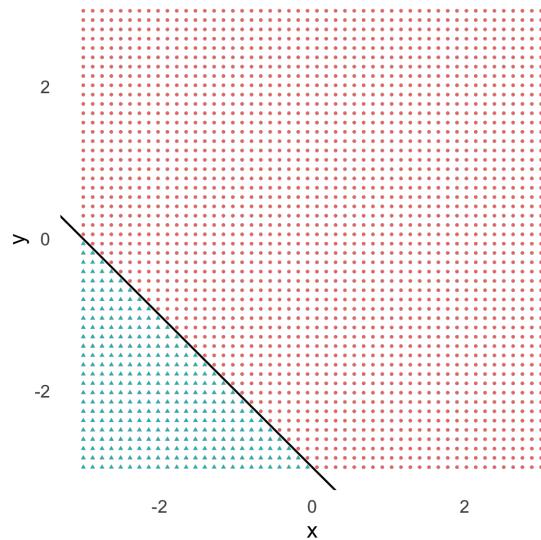


Flat: the hyperplane is not curved, it increases/decreases constantly in each direction

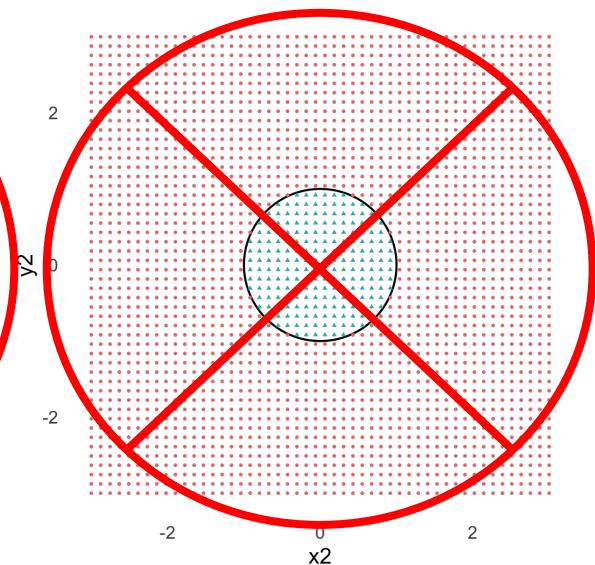
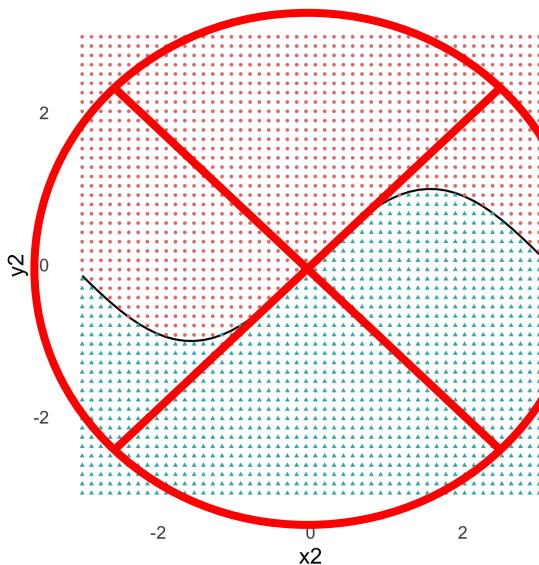
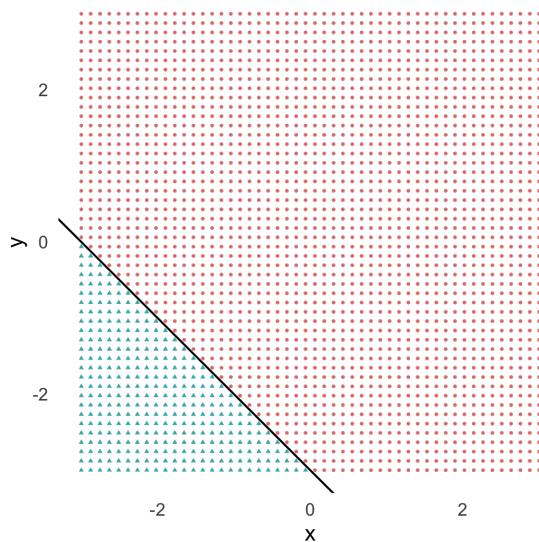
Affine: the hyperplane doesn't need to pass through the origin

Subspace: a subset of vectors in a larger vector space (closure under addition and scalar multiplication; here, a hyperplane with $n-1$ dimensions)

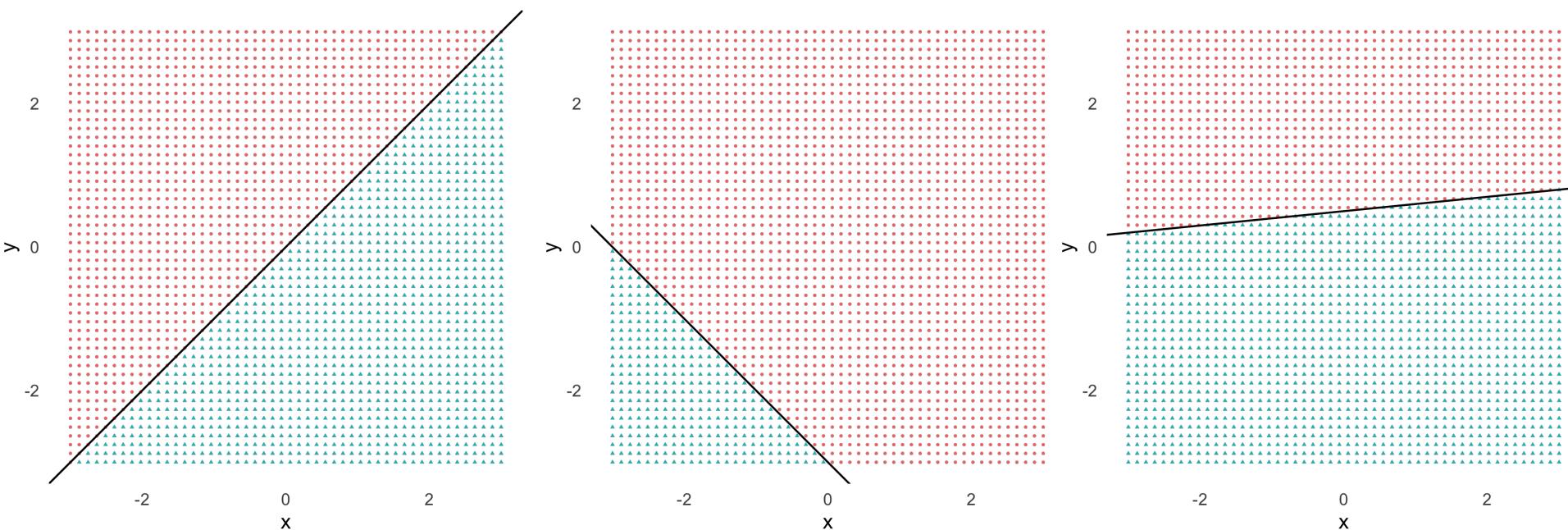
Quick Check: Which of these are flat affine subspaces?



Quick Check: Which of these are flat affine subspaces?

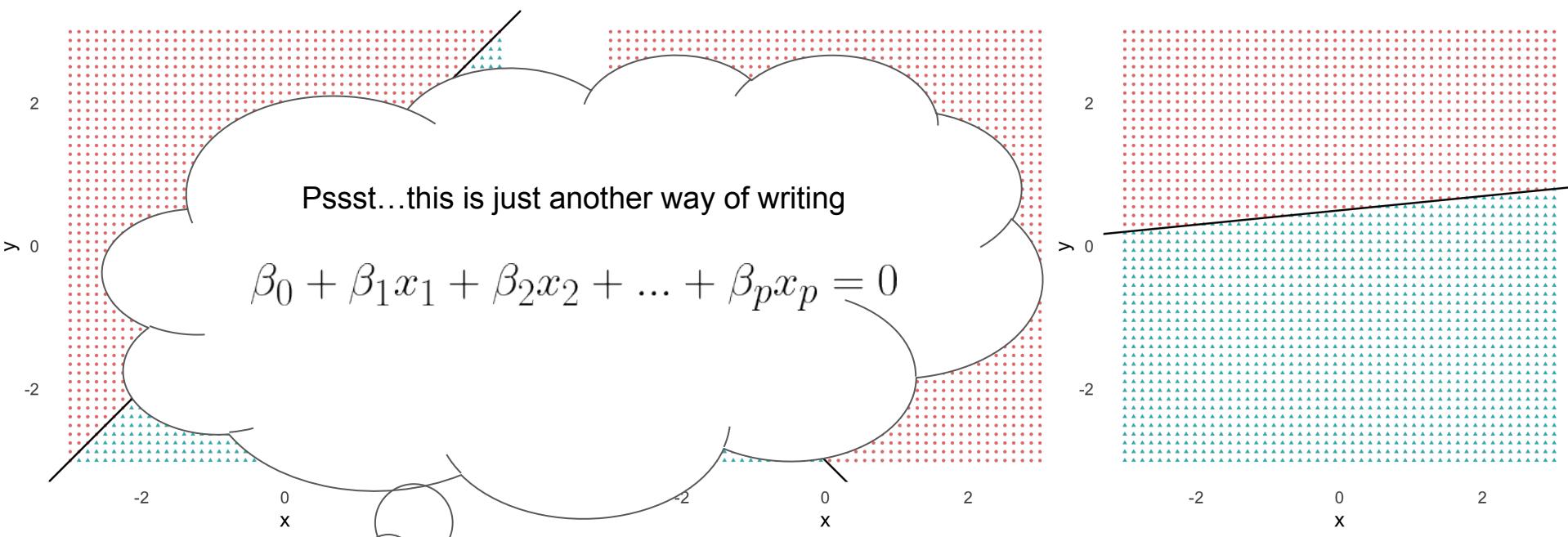


Hyperplanes Divide Spaces in Half



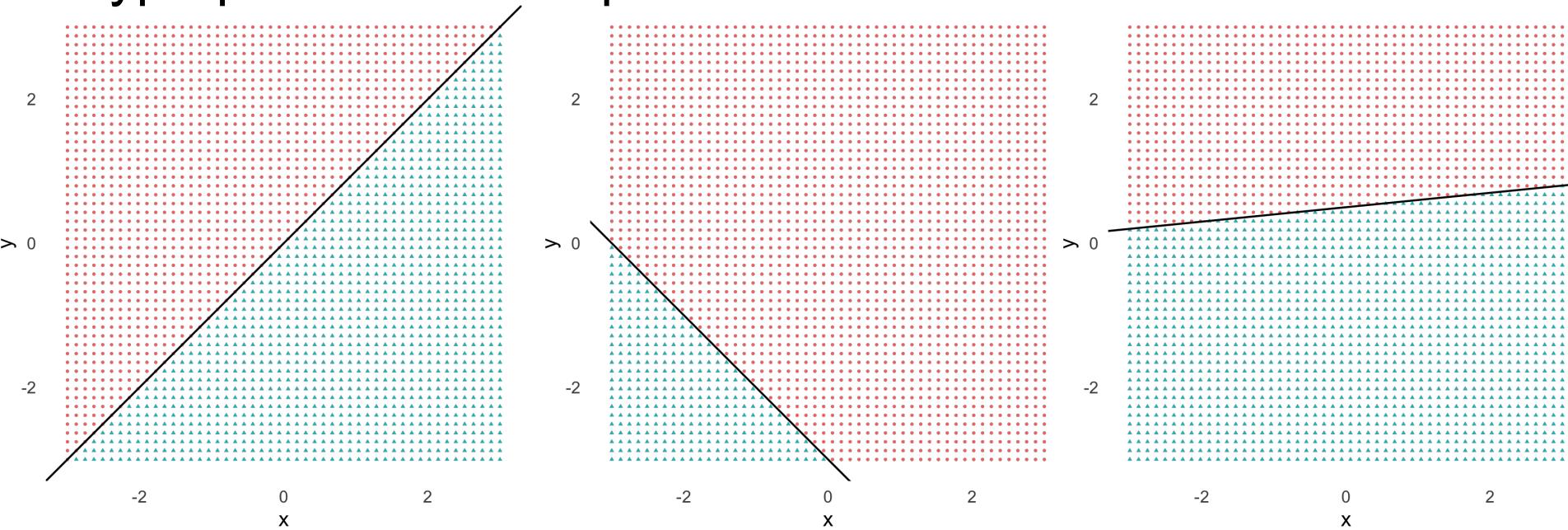
$$\mathbf{w}^T \mathbf{x} + b = 0$$

Hyperplanes Divide Spaces in Half



$$\mathbf{w}^T x + b = 0$$

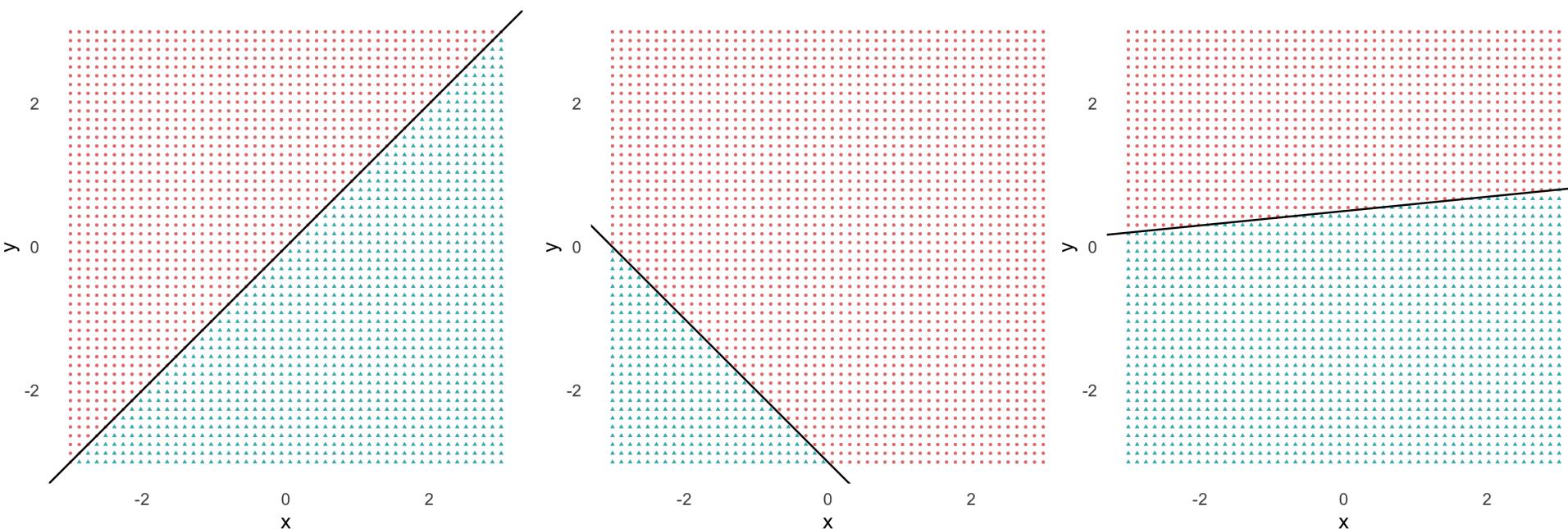
Hyperplanes Divide Spaces in Half



$$\mathbf{w}^T \mathbf{x}_n + b > 0 \text{ if } t_n = 1$$

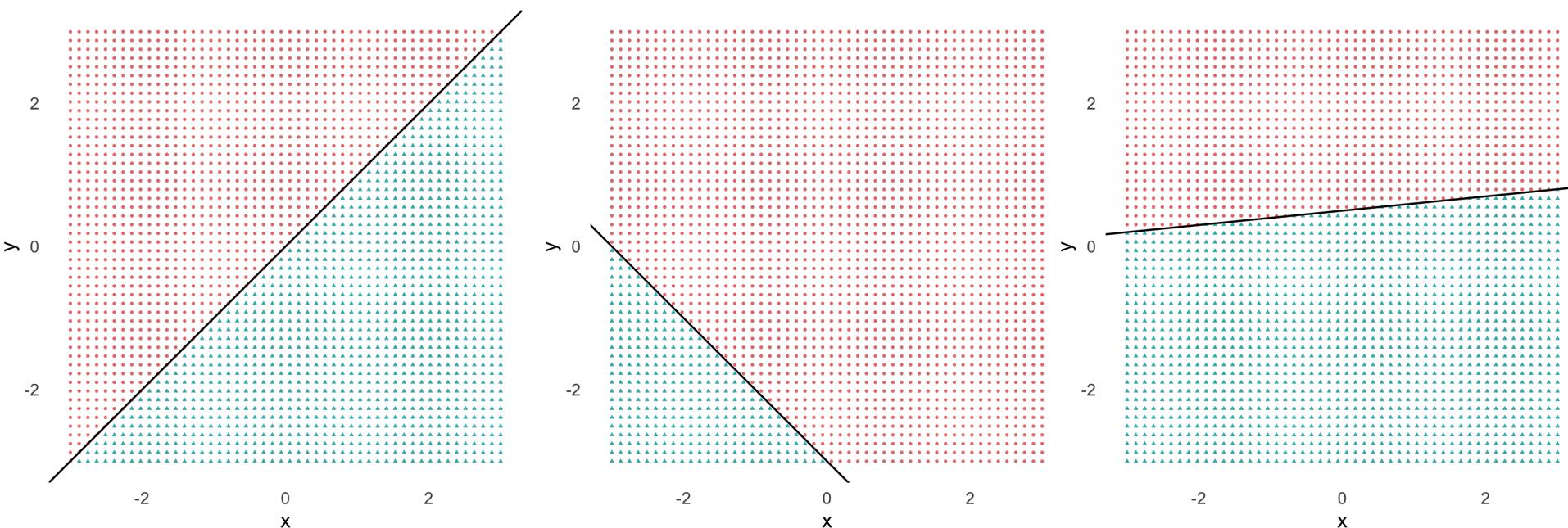
$$\mathbf{w}^T \mathbf{x}_n + b < 0 \text{ if } t_n = -1$$

Hyperplanes Divide Spaces in Half



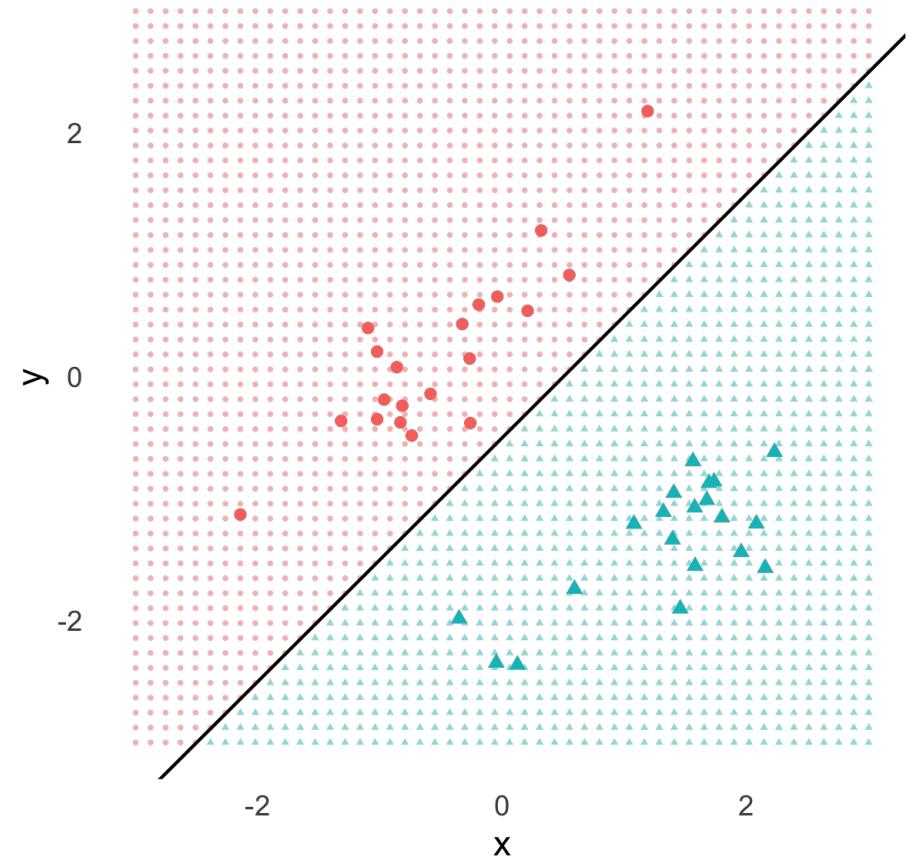
$$t_n(\mathbf{w}^T \mathbf{x}_n + b) > 0$$

Hyperplanes Divide Spaces in Half

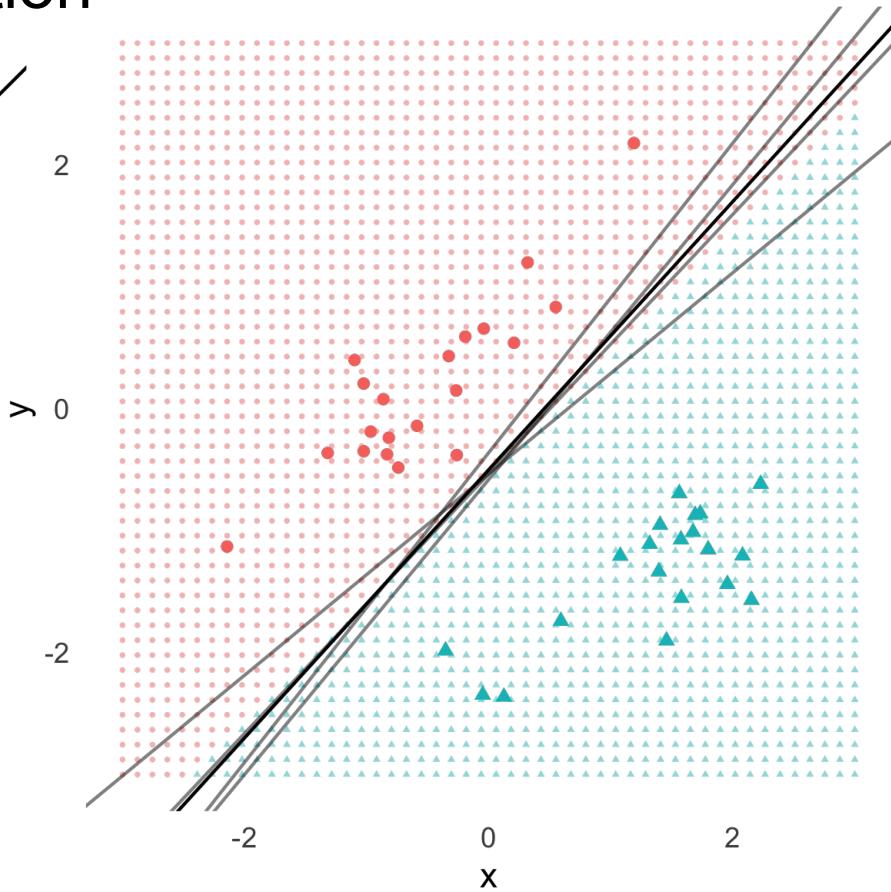
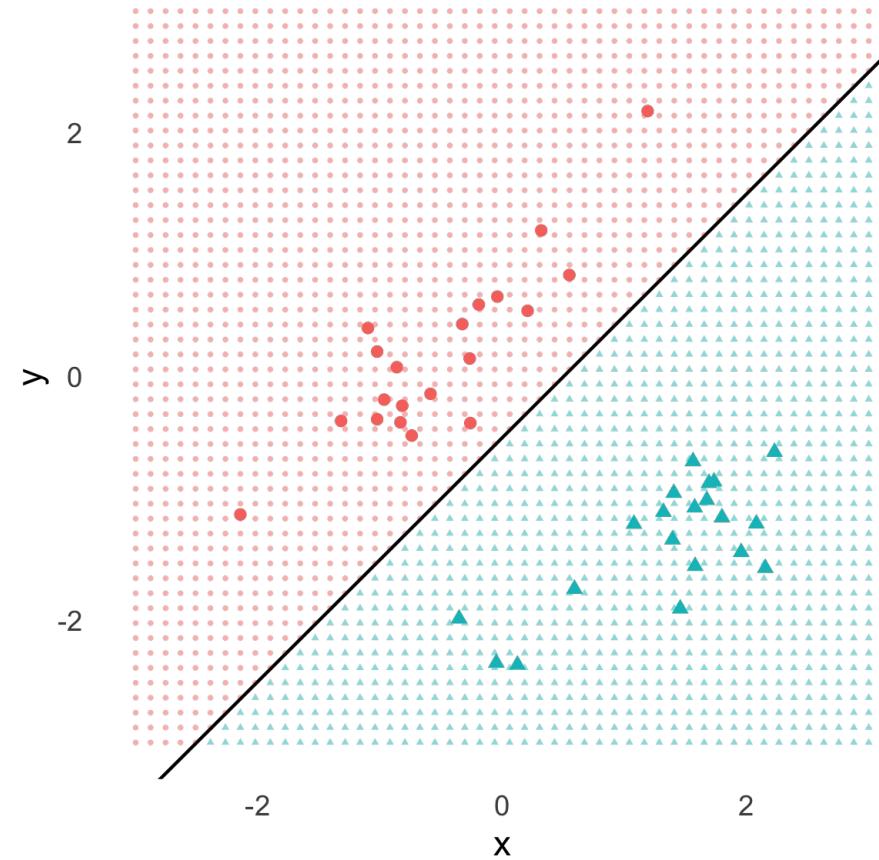


We can use these hyperplanes to classify data!

Hyperplanes for Classification

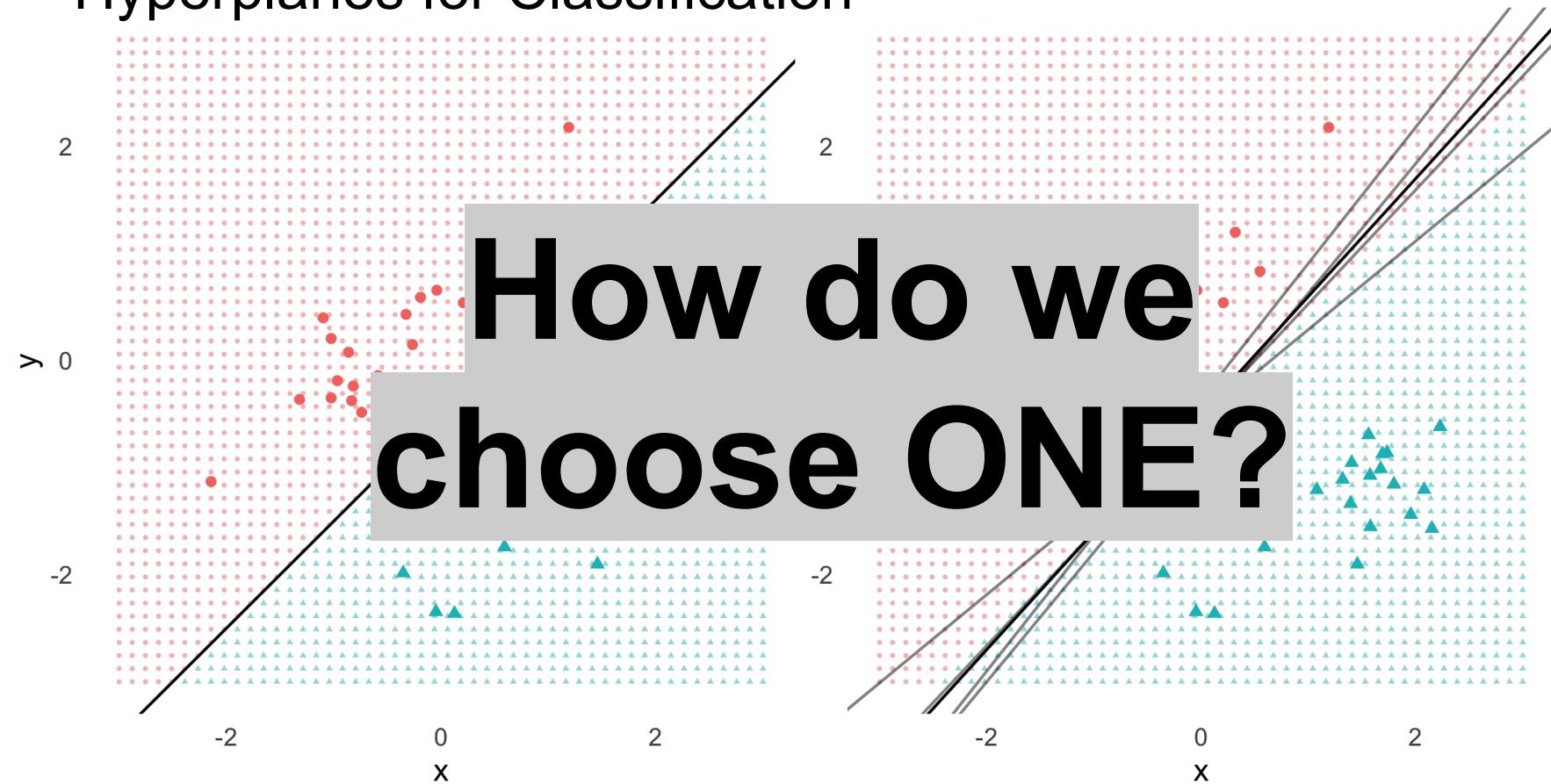


Hyperplanes for Classification



Hyperplanes for Classification

How do we
choose ONE?

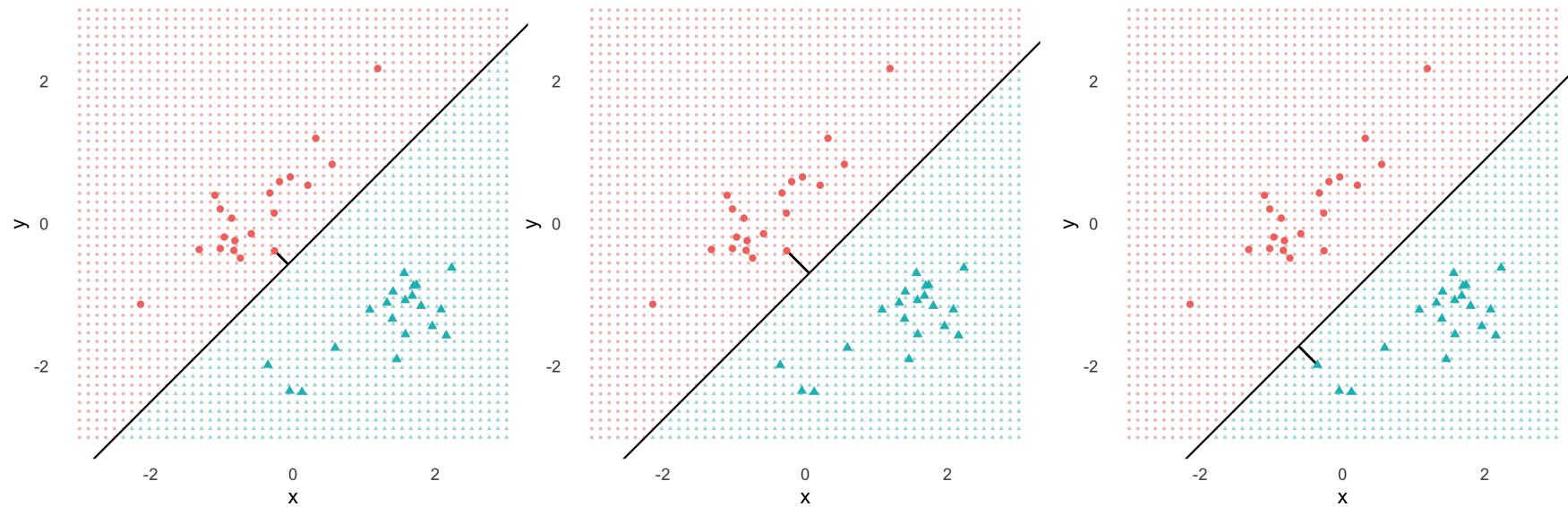


Maximal Margin Classifier

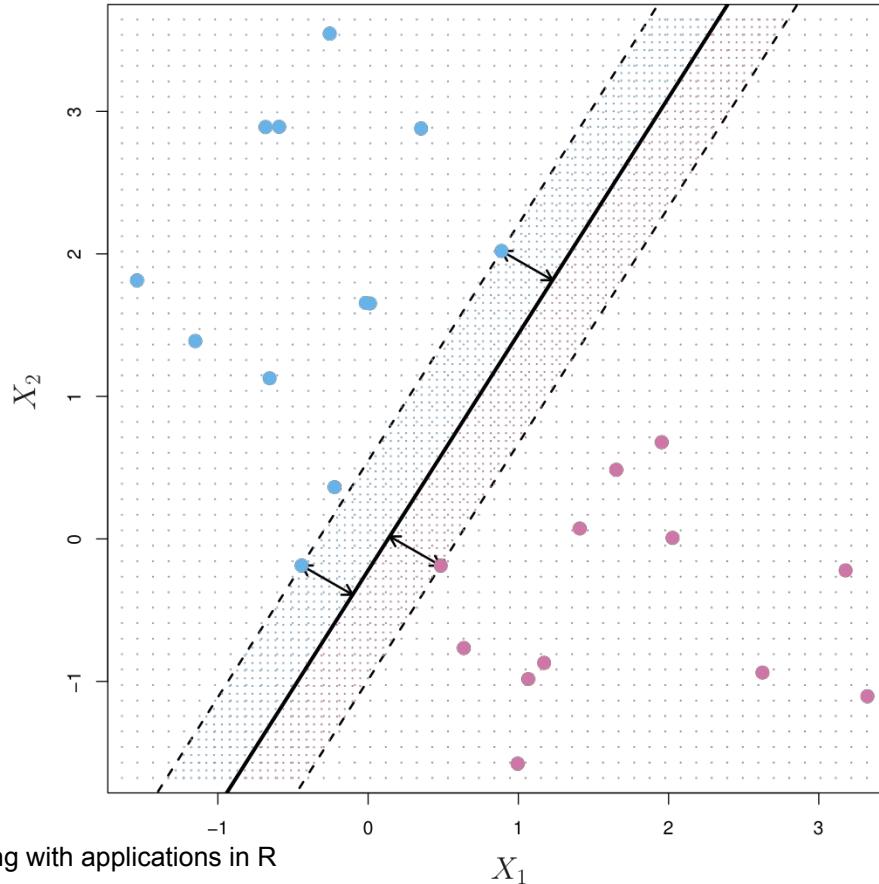
Maximal Margin Classifier

Problem: If a separating hyperplane exists, there will be ∞

Solution: Choose the one that's *furthest* from the training examples



Maximal Margin Classifier



$$\underbrace{\frac{t_n (w^T x_n + b)}{\|w\|}}_{\text{distance btw } x_n \text{ and line}}$$

Maximal Margin Classifier Math

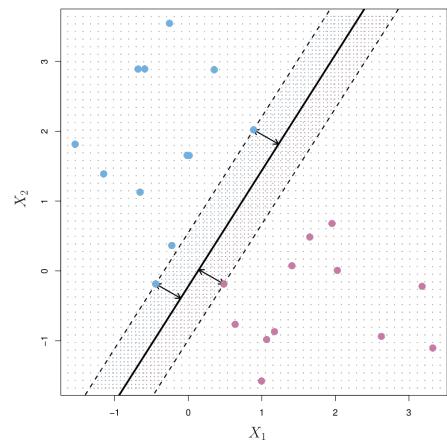
$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n (\mathbf{w}^T \mathbf{x}_n + b) \right] \right\}$$

maximize the minimum distance btw hyperplane and point

subject to: $t_n (\mathbf{w}^T \mathbf{x}_n + b) \geq 1$

Easier to minimize

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$



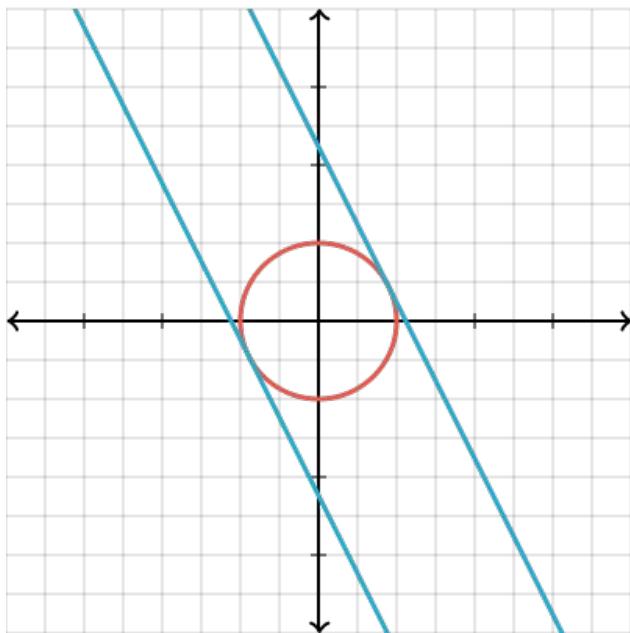
$$f(x, y) = 2x + y$$

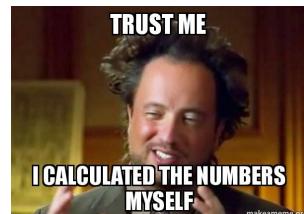
$$g(x, y) = x^2 + y^2 = 1$$

Constrained Optimization

Optimize $f(x, y, z)$ subject to $g(x, y, z) = k$

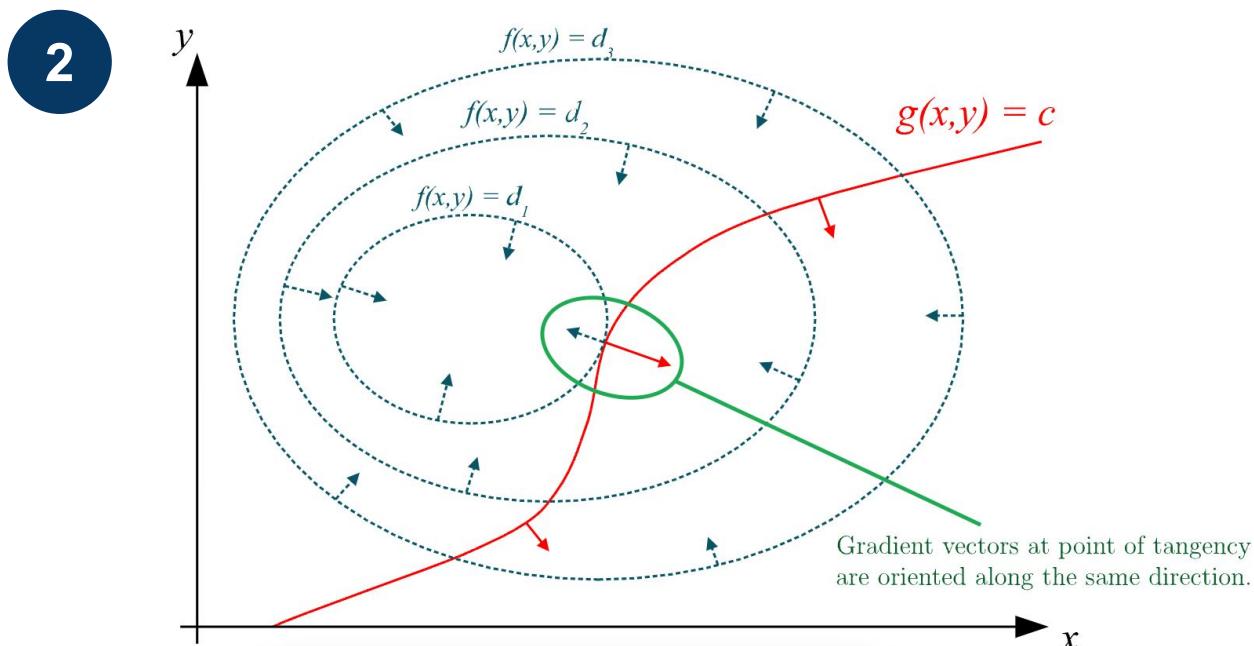
1





Constrained Optimization

Optimize $f(x, y, z)$ subject to $g(x, y, z) = k$



Constrained Optimization

Optimize $f(x, y, z)$ subject to $g(x, y, z) = k$

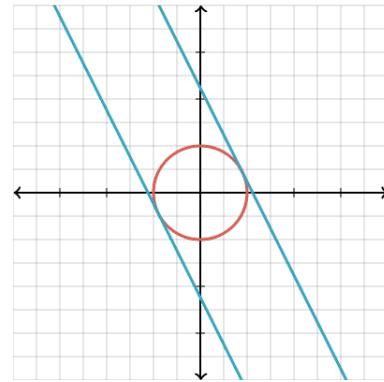
3

$$\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$$

Constrained Optimization

Optimize $f(x, y, z)$ subject to $g(x, y, z) = k$

3 $\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$



$$f(x, y) = 2x + y$$

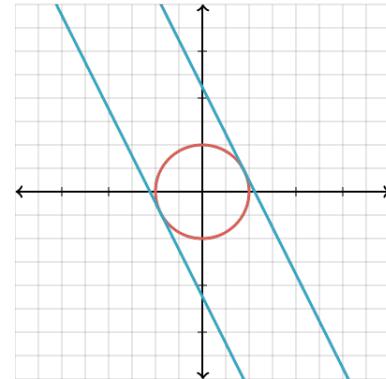
$$g(x, y) = x^2 + y^2 = 1$$

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} 2x + y \\ \frac{\partial}{\partial y} 2x + y \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\nabla g(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} x^2 + y^2 \\ \frac{\partial}{\partial y} x^2 + y^2 \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

Constrained Optimization

Optimize $f(x, y, z)$ subject to $g(x, y, z) = k$



3 $\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$

$$f(x, y) = 2x + y$$
$$g(x, y) = x^2 + y^2 = 1$$

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} 2x + y \\ \frac{\partial}{\partial y} 2x + y \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

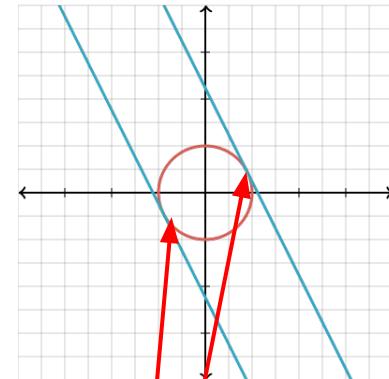
$$\nabla g(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} x^2 + y^2 \\ \frac{\partial}{\partial y} x^2 + y^2 \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

$$\boxed{\begin{bmatrix} 2 \\ 1 \end{bmatrix}} = \lambda_0 \boxed{\begin{bmatrix} 2x_0 \\ 2y_0 \end{bmatrix}}$$

Constrained Optimization

Optimize $f(x, y, z)$ subject to $g(x, y, z) = k$

3 $\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$



$$f(x, y) = 2x + y$$

$$g(x, y) = x^2 + y^2 = 1$$

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} 2x + y \\ \frac{\partial}{\partial y} 2x + y \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\nabla g(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} x^2 + y^2 \\ \frac{\partial}{\partial y} x^2 + y^2 \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

$$\boxed{\begin{bmatrix} 2 \\ 1 \end{bmatrix}} = \lambda_0 \boxed{\begin{bmatrix} 2x_0 \\ 2y_0 \end{bmatrix}}$$

Constrained Optimization

$$f(x, y) = 2x + y$$

$$g(x, y) = x^2 + y^2 = 1$$

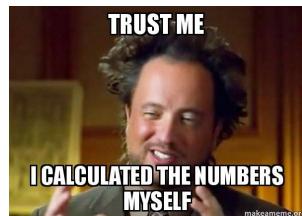
Optimize $f(x, y, z)$ subject to $g(x, y, z) = k$

4

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda(g(x, y) - c)$$

$$\mathcal{L}(x, y, \lambda) = 2x + y - \lambda(x^2 + y^2 - 1)$$

For example



Constrained Optimization

Optimize $f(x, y, z)$ subject to $g(x, y, z) = k$

4

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda(g(x, y) - c)$$

$$\mathcal{L}_\lambda(x, y, \lambda) = \frac{\partial \mathcal{L}}{\partial \lambda} = 0 - (g(x, y) - c) = -g(x, y) + c$$

$$\mathcal{L}_\lambda(x, y, \lambda) = -g(x, y) + c = 0 \text{ when } g(x, y) = c$$

Constrained Optimization

Optimize $f(x, y, z)$ subject to $g(x, y, z) = k$

4

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda(g(x, y) - c)$$

$$\mathcal{L}_x(x, y, \lambda) = 0$$

$$\frac{\partial}{\partial x}(f(x, y) - \lambda(g(x, y) - c)) = 0$$

$$f_x(x, y) - \lambda g_x(x, y) = 0$$

$$f_x(x, y) = \lambda g_x(x, y)$$

Constrained Optimization

Optimize $f(x, y, z)$ subject to $g(x, y, z) = k$

4

$$\nabla \mathcal{L} = \begin{bmatrix} \frac{\partial}{\partial x} (2x + y - \lambda(x^2 + y^2 - 1)) \\ \frac{\partial}{\partial y} (2x + y - \lambda(x^2 + y^2 - 1)) \\ \frac{\partial}{\partial \lambda} (2x + y - \lambda(x^2 + y^2 - 1)) \end{bmatrix} = \begin{bmatrix} 2 - 2\lambda x \\ 1 - 2\lambda y \\ -x^2 - y^2 + 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

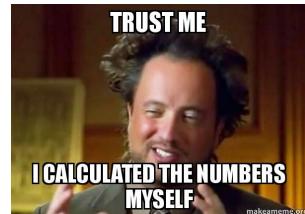
SVMs as Constrained Optimization

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

$$t_n(w^T x_n + b) \geq 1 \text{ for all } n$$

$$\mathcal{L}(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n \{t_n(w^T x_n + b) - 1\}$$

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$



SVMs as Constrained Optimization

$$t_n(w^T x_n + b) \geq 1 \text{ for all } n$$

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N \alpha_n \{t_n(w^T x_n + b) - 1\}$$

$$\boxed{\begin{aligned} \mathbf{w} &= \sum_{n=1}^N a_n t_n x_n \\ 0 &= \sum_{n=1}^N a_n t_n \end{aligned}}$$

$$L(\tilde{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^M \underbrace{a_n a_m}_{\text{lagrange multipliers}} \underbrace{t_n t_m}_{\text{targets}} \underbrace{x_n \cdot x_m}_{\text{dot product between two vectors}}$$

$a_n \geq 0, \text{ for } n = 1, \dots, N$

$$\sum_{n=1}^N a_n t_n = 0$$

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

$$t_n(w^T x_n + b) \geq 1 \text{ for all } n$$

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N \alpha_n \{t_n(w^T x_n + b) - 1\}$$

SVMs as Constrained Optimization

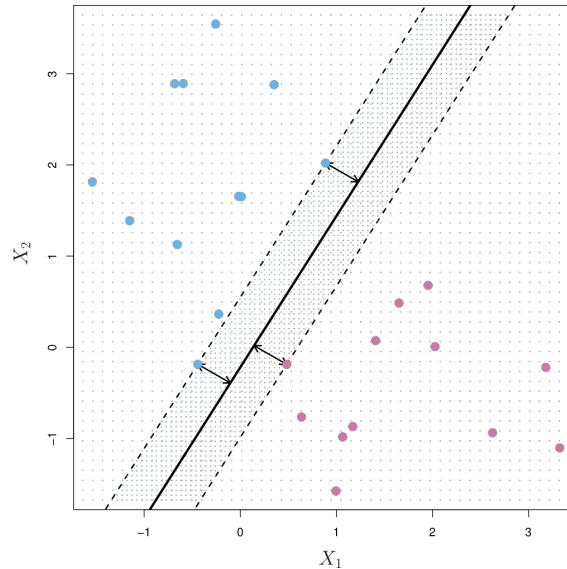
formula to classify new data point x

$$y(\mathbf{w}) = \sum_{n=1}^N a_n t_n \ x \cdot x_n + b$$

$$a_n \geq 0$$

$$t_n y(x_n) - 1 \geq 0$$

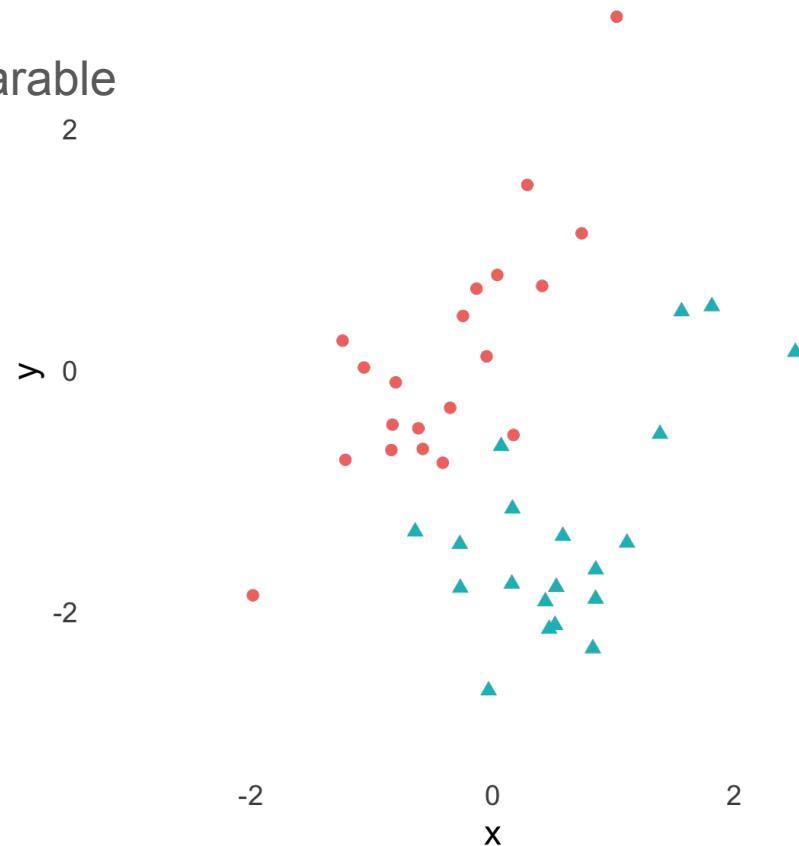
② $a_n \{t_n y(x_n) - 1\} = 0$



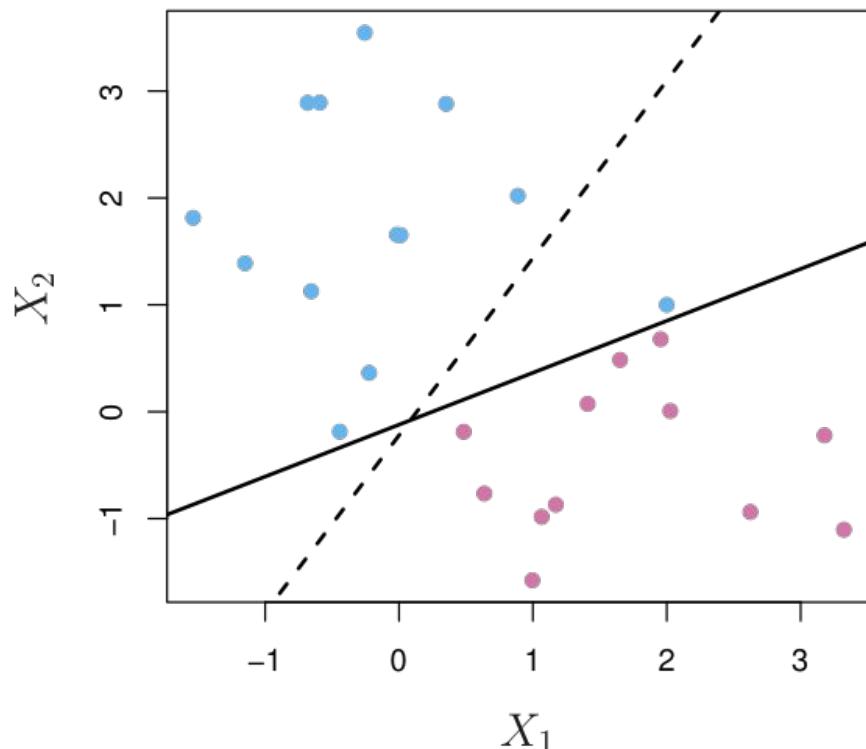
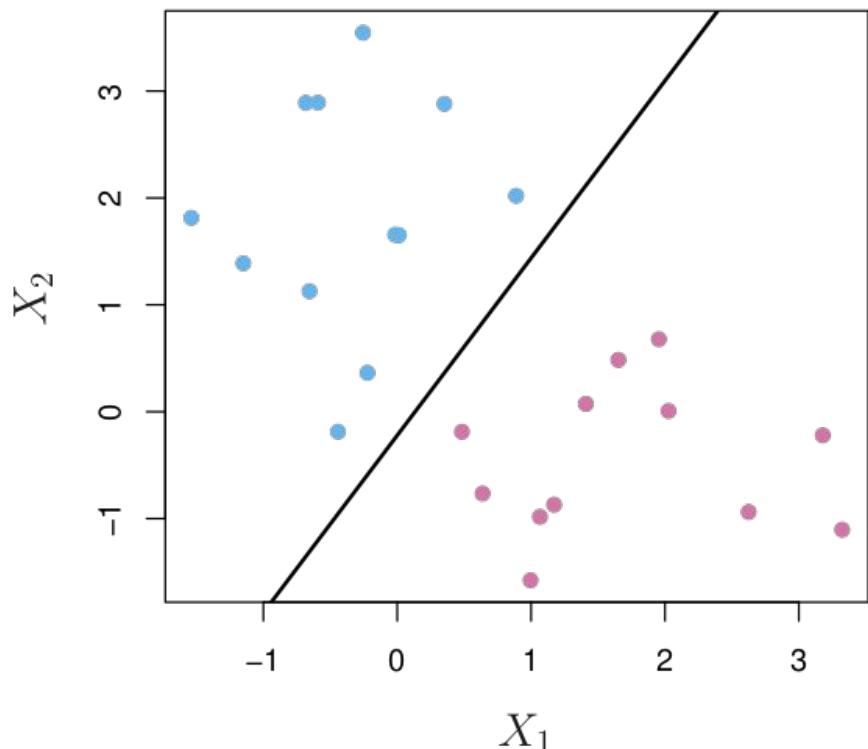
Support Vector Classifier

Support Vector Classifier

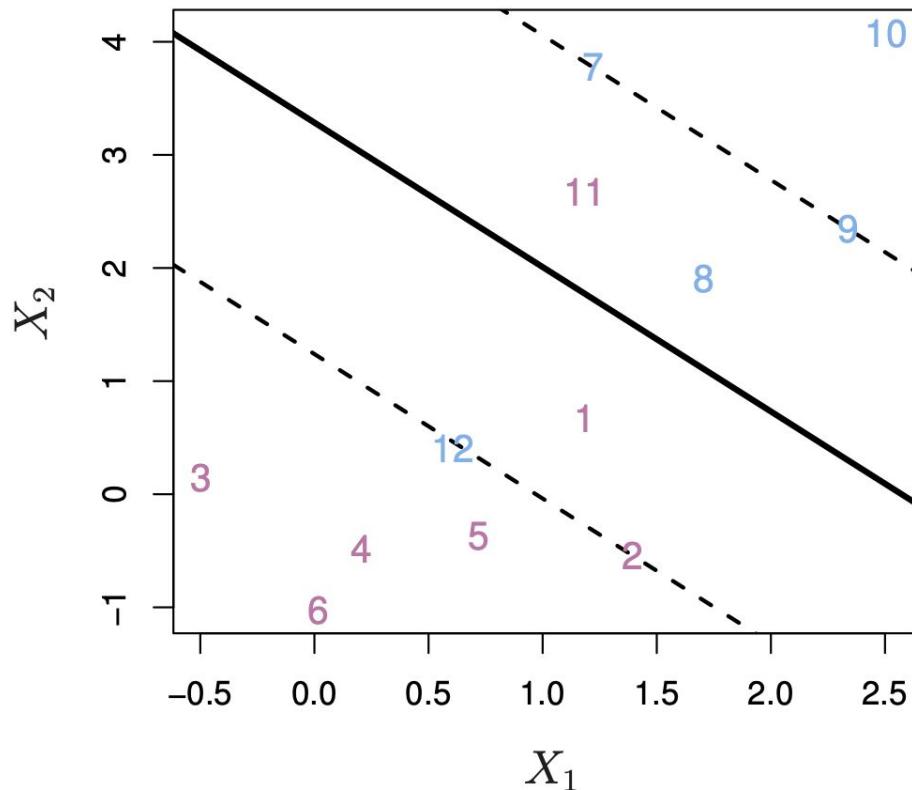
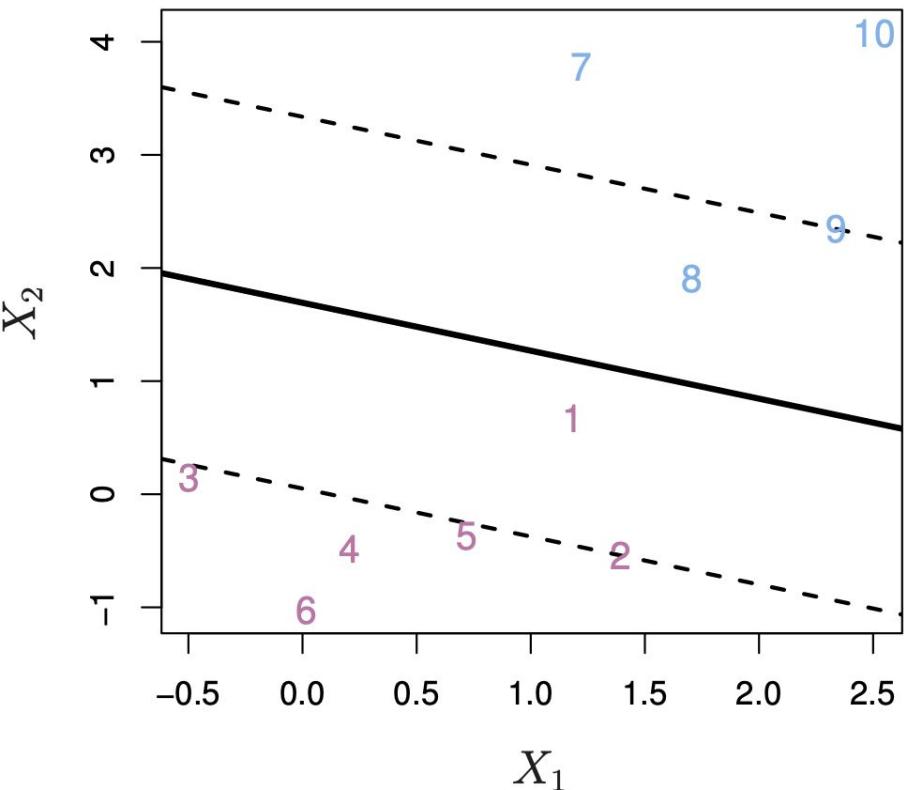
Problem: Non-Separable



Support Vector Classifier



Support Vector Classifier



Support Vector Classifier

Before: minimize $\frac{1}{2} \|\mathbf{w}\|^2$

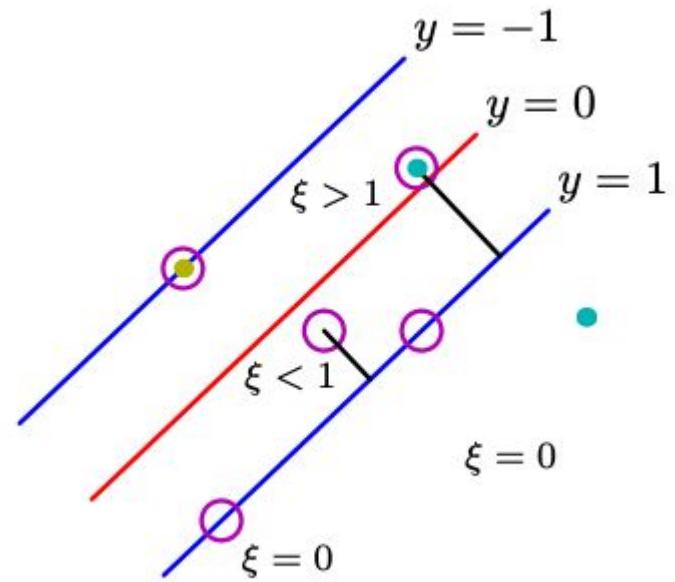
After: minimize $C \sum_{n=1}^N \xi_n + \frac{1}{2} \|\mathbf{w}\|^2$

Before:

$$t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1$$

After:

$$t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1 - \xi_n$$



Support Vector Classifier

Before: $\frac{1}{2} \|\mathbf{w}\|^2$

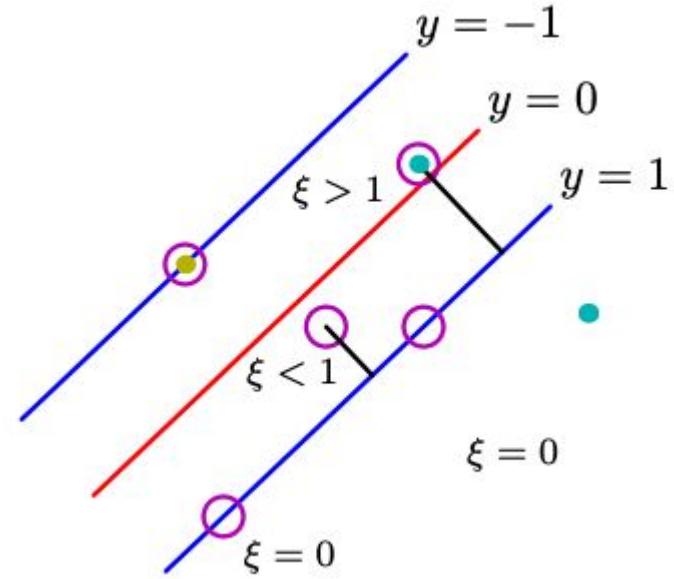
After: $C \sum_{n=1}^N \xi_n + \frac{1}{2} \|\mathbf{w}\|^2$

Before:

$$t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1$$

After:

$$t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1 - \xi_n$$

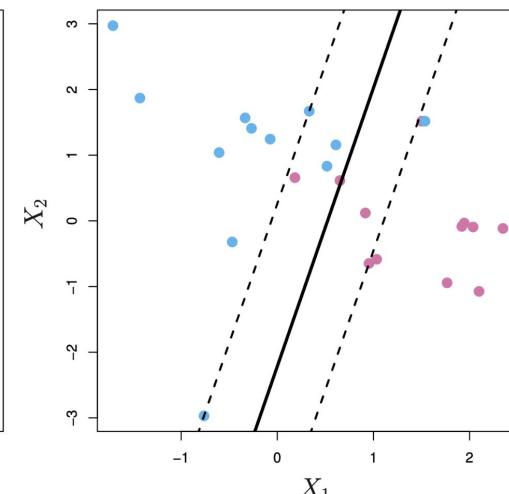
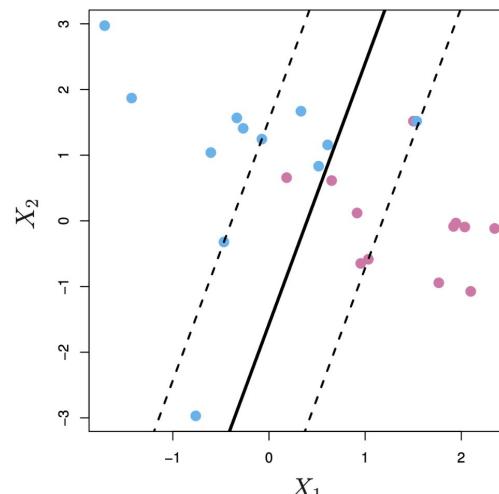
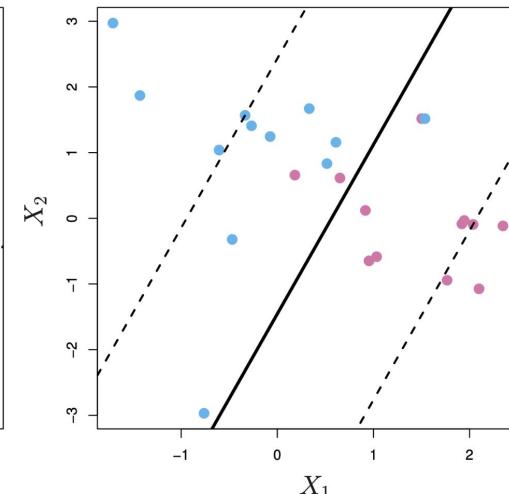
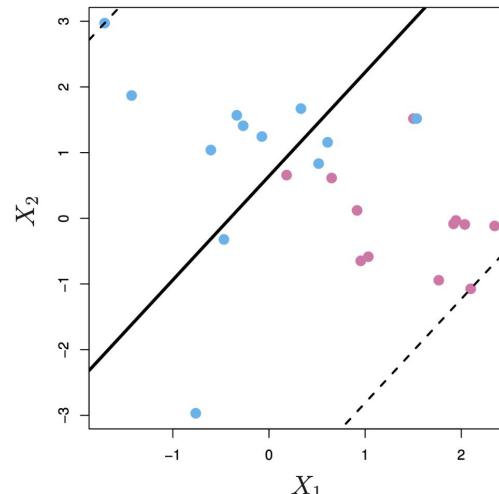


| | Regularization | Margins | Bias/Variance |
|---------|---------------------|----------|------------------------------|
| Small C | More regularization | wider | Prone to underfitting |
| Large C | Less regularization | narrower | Prone to overfitting |

Support Vector Classifier

C: Penalty on Error

| | Regularization | Margins | Bias/Variance |
|---------|---------------------|----------|------------------------------|
| Small C | More regularization | wider | Prone to underfitting |
| Large C | Less regularization | narrower | Prone to overfitting |



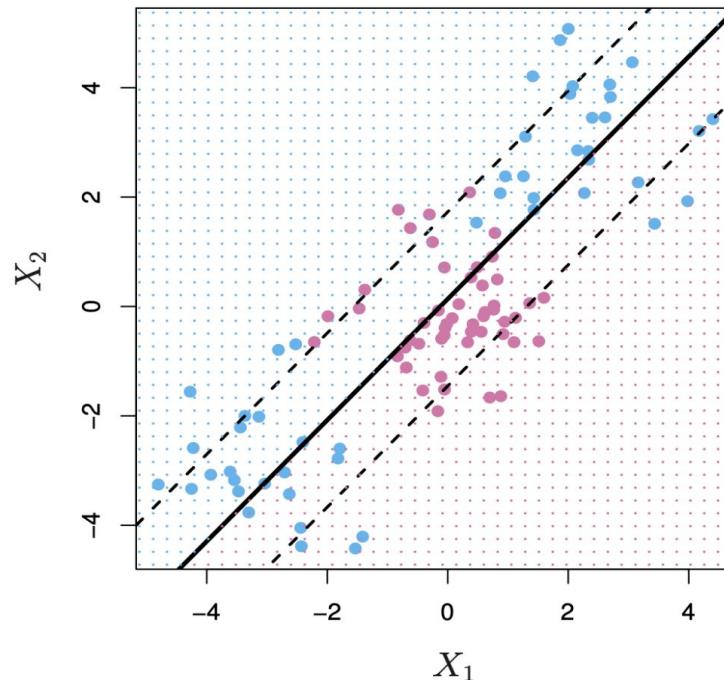
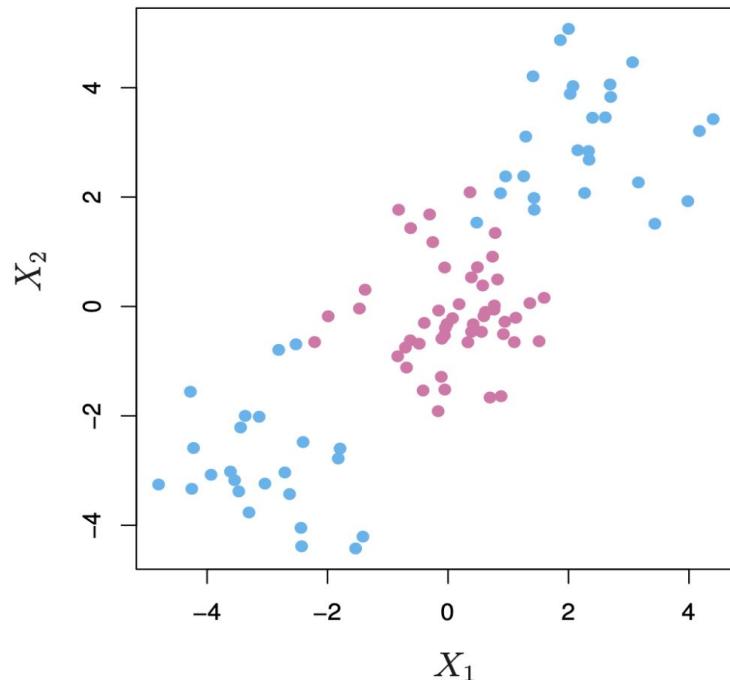
Support Vector Machines II

Dr. Chelsea Parlett-Pelleriti

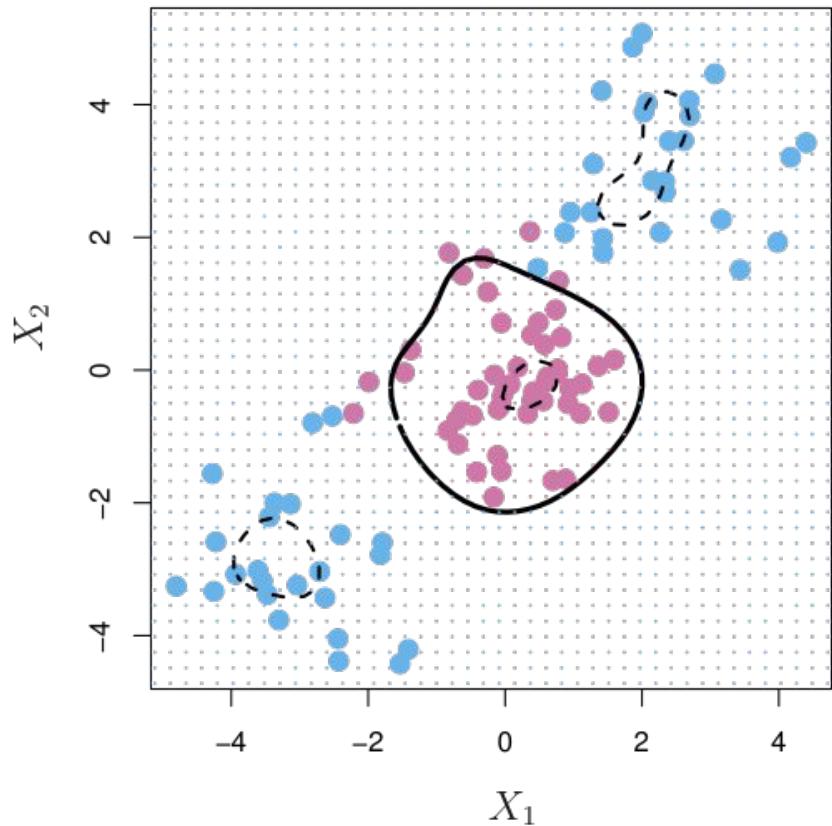
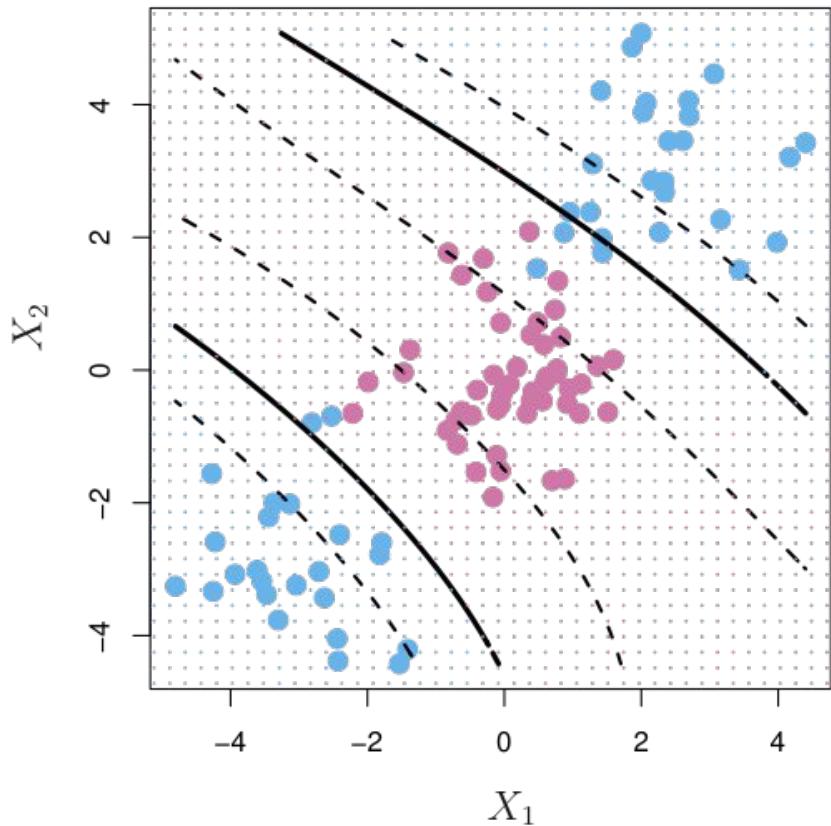
Support Vector Machines

Support Vector Machines

New problem: not linearly separable even with error

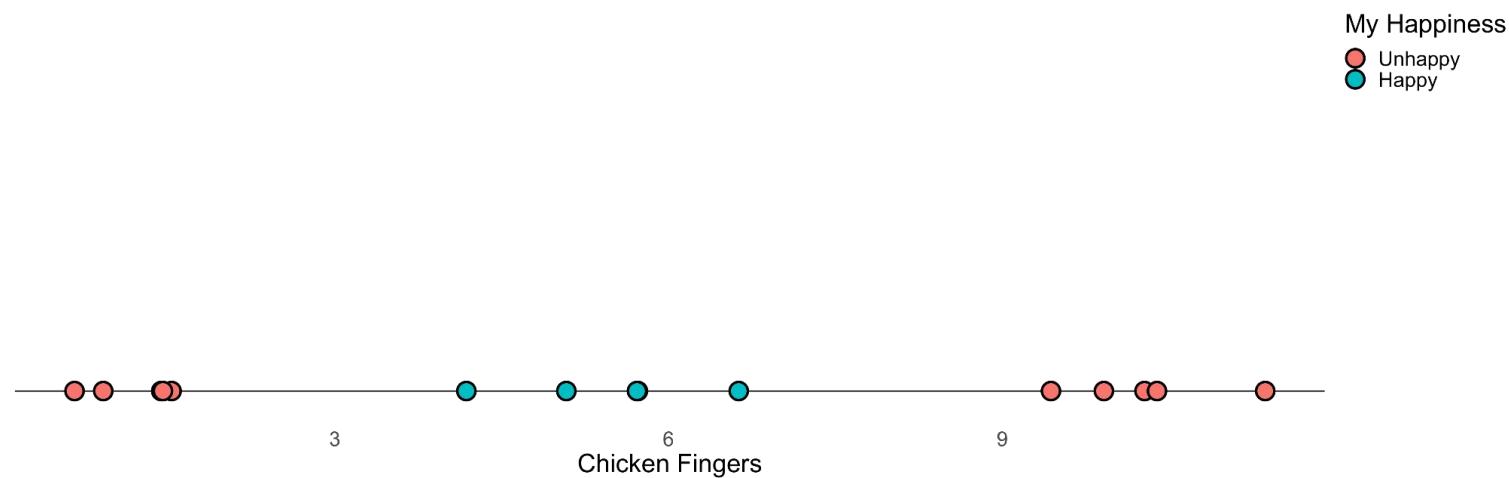


The Kernel Trick



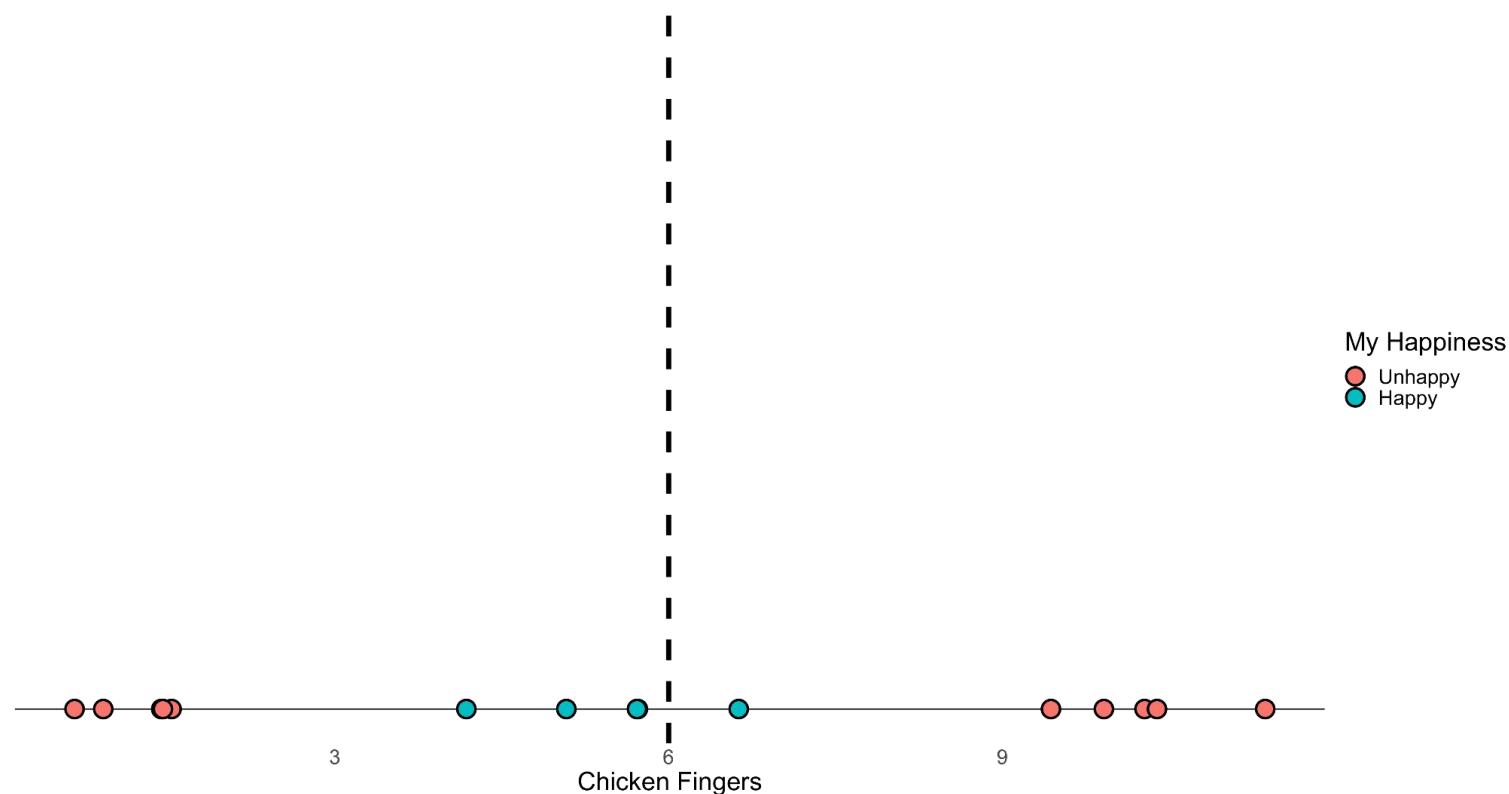
The Kernel Trick

Chelsea's Happiness When Eating Chicken Fingers



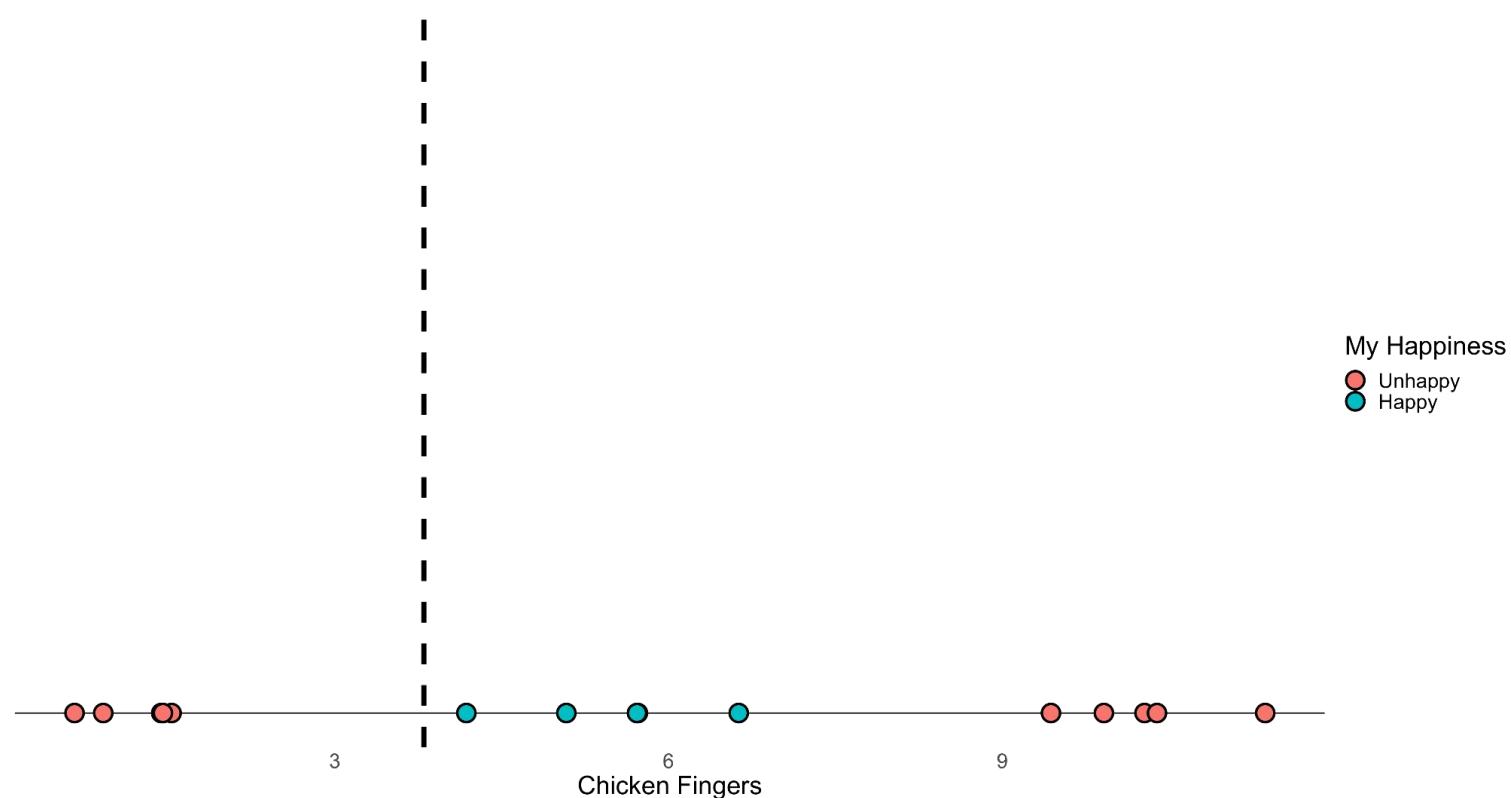
The Kernel Trick

Chelsea's Happiness When Eating Chicken Fingers



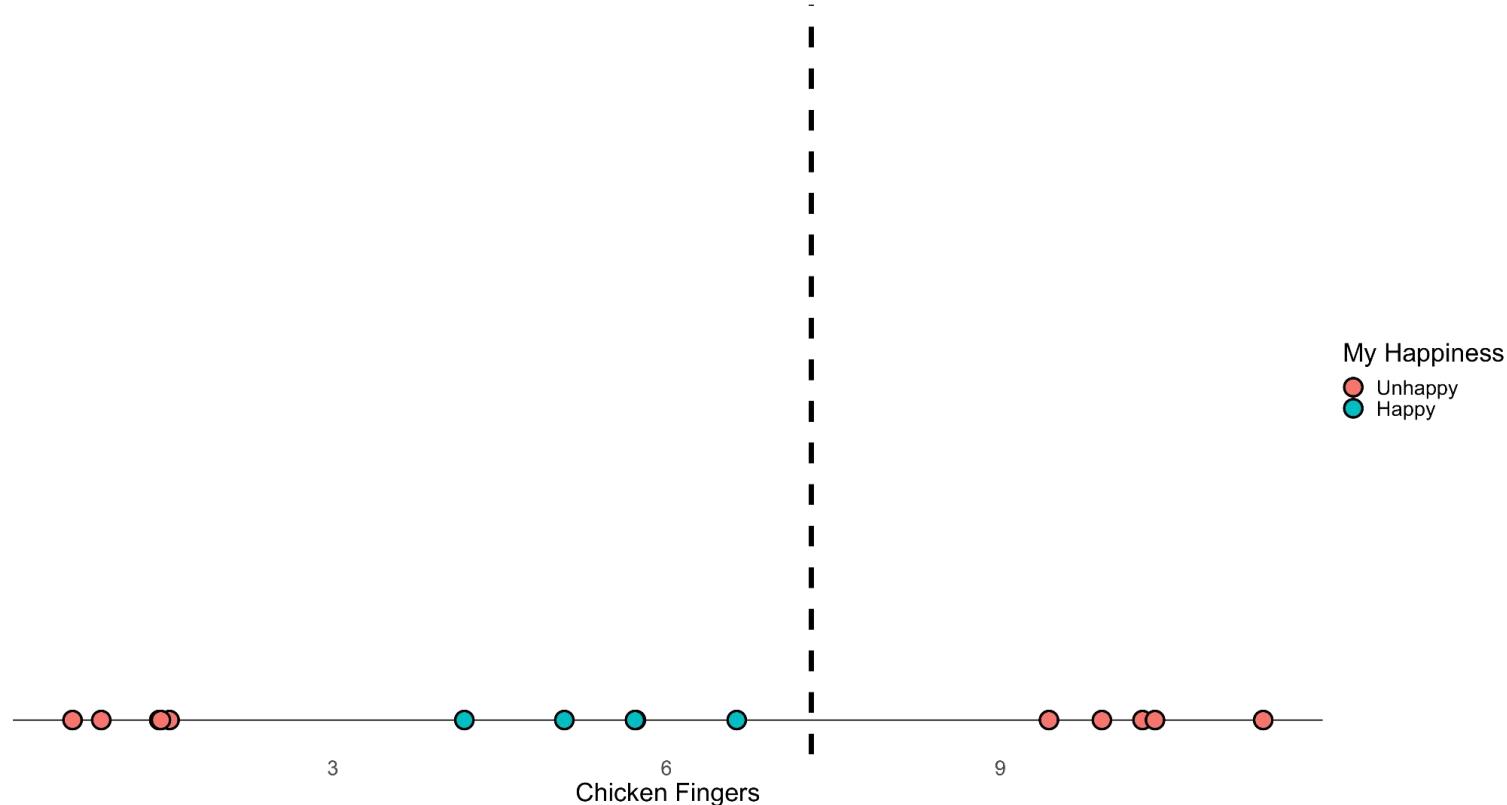
The Kernel Trick

Chelsea's Happiness When Eating Chicken Fingers



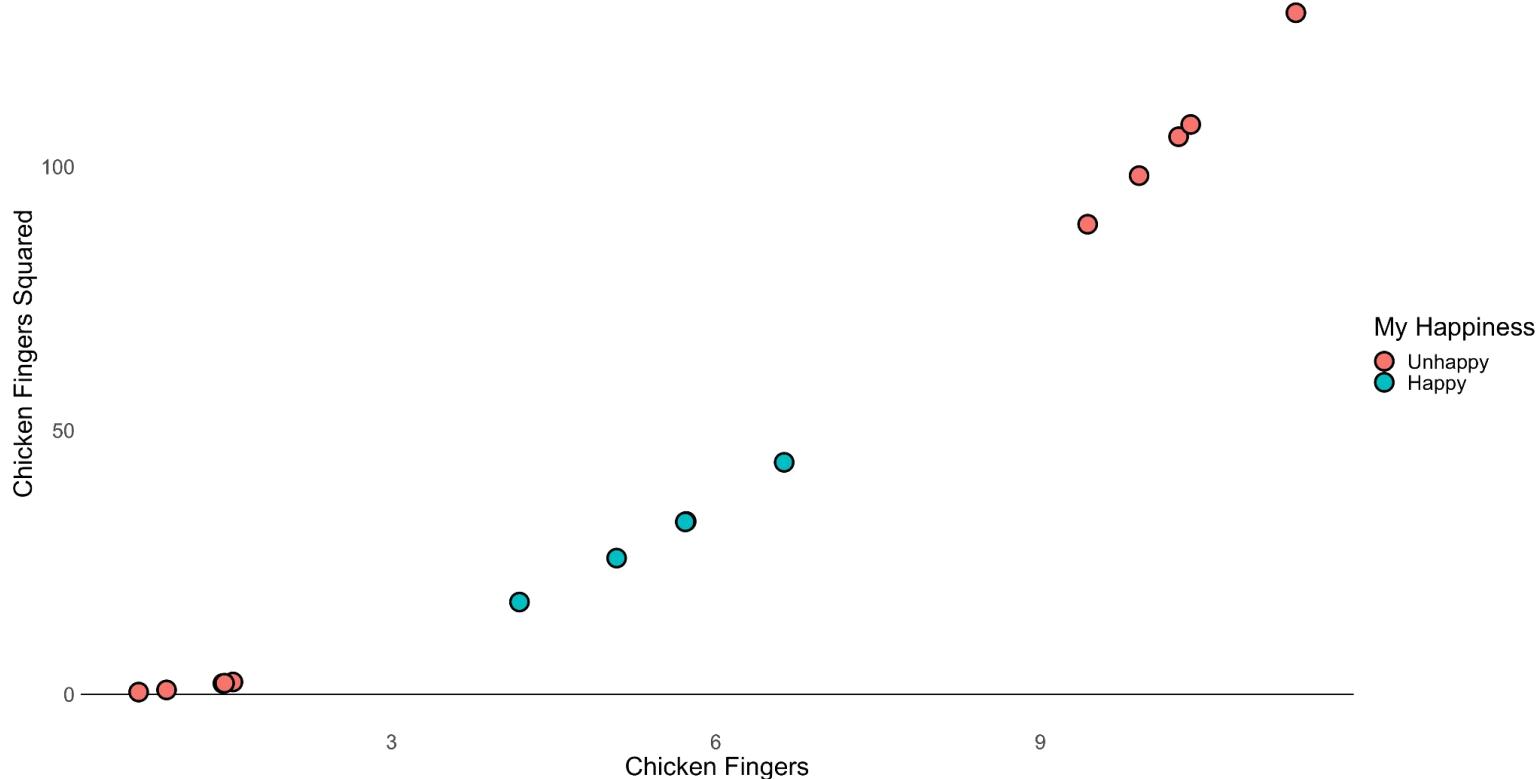
The Kernel Trick

Chelsea's Happiness When Eating Chicken Fingers



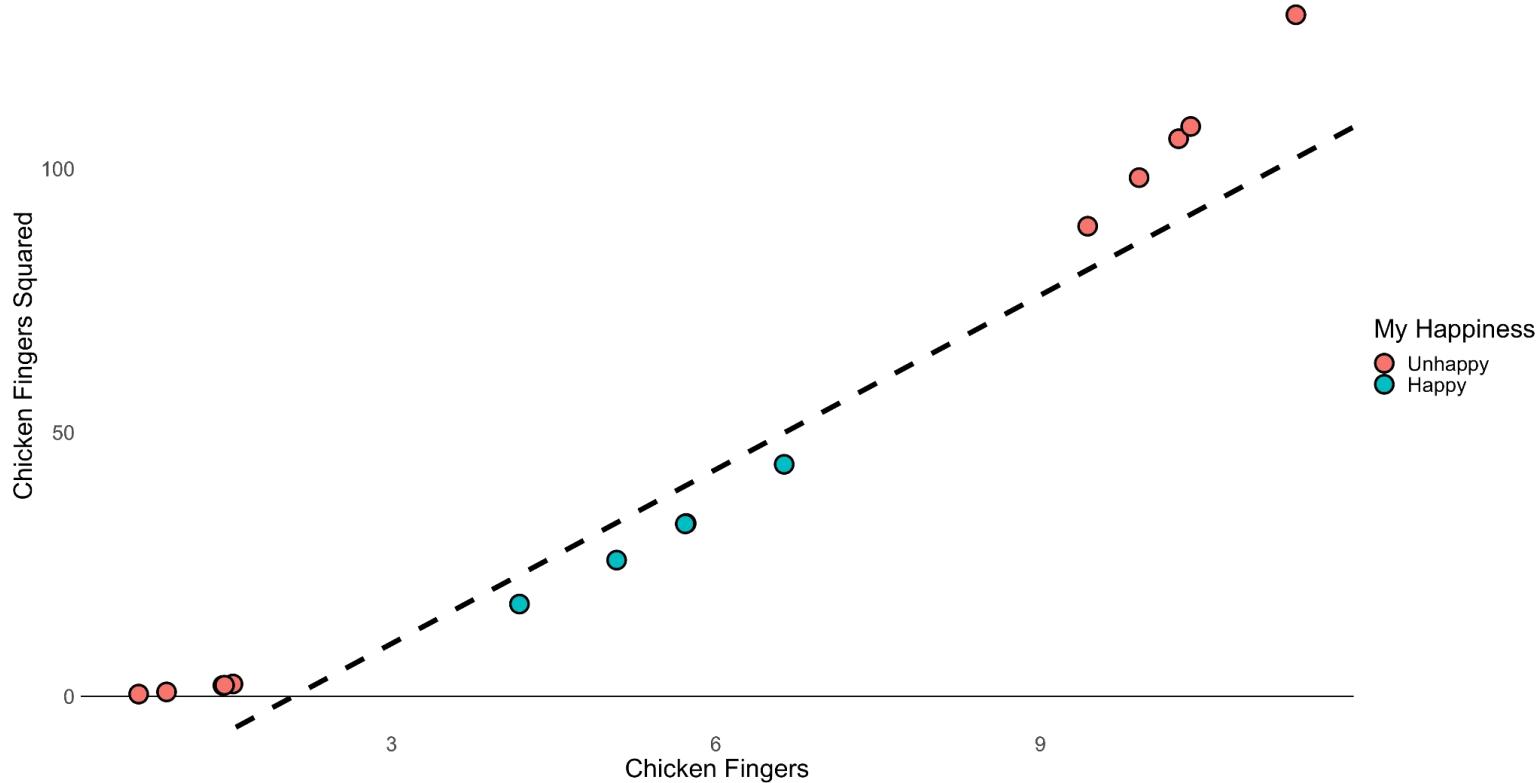
The Kernel Trick

Chelsea's Happiness When Eating Chicken Fingers



The Kernel Trick

Chelsea's Happiness When Eating Chicken Fingers



The Kernel Trick

What is a Kernel (in SVMs)?

A function that calculates the relationship between two vectors in multiple dimensions (without actually having to calculate the coordinates for those dimensions)

The Kernel Trick

$$\underbrace{K(x, y)}_{\text{kernel}} = (x * y + r)^d \quad \bullet \quad \begin{array}{l} \text{We used a Polynomial Kernel with } \mathbf{r} \\ \text{(coefficient) and } \mathbf{d} \text{ (degree)} \end{array}$$

The Kernel Trick

$$\underbrace{K(x, y)}_{\text{kernel}} = (x * y + r)^d$$

A point

Another
point

- We used a Polynomial Kernel with **r** (coefficient) and **d** (degree)
- Let's try $d = 2, r = \frac{1}{2}$

The Kernel Trick

$$\underbrace{K(x, y)}_{\text{kernel}} = (x * y + r)^d$$

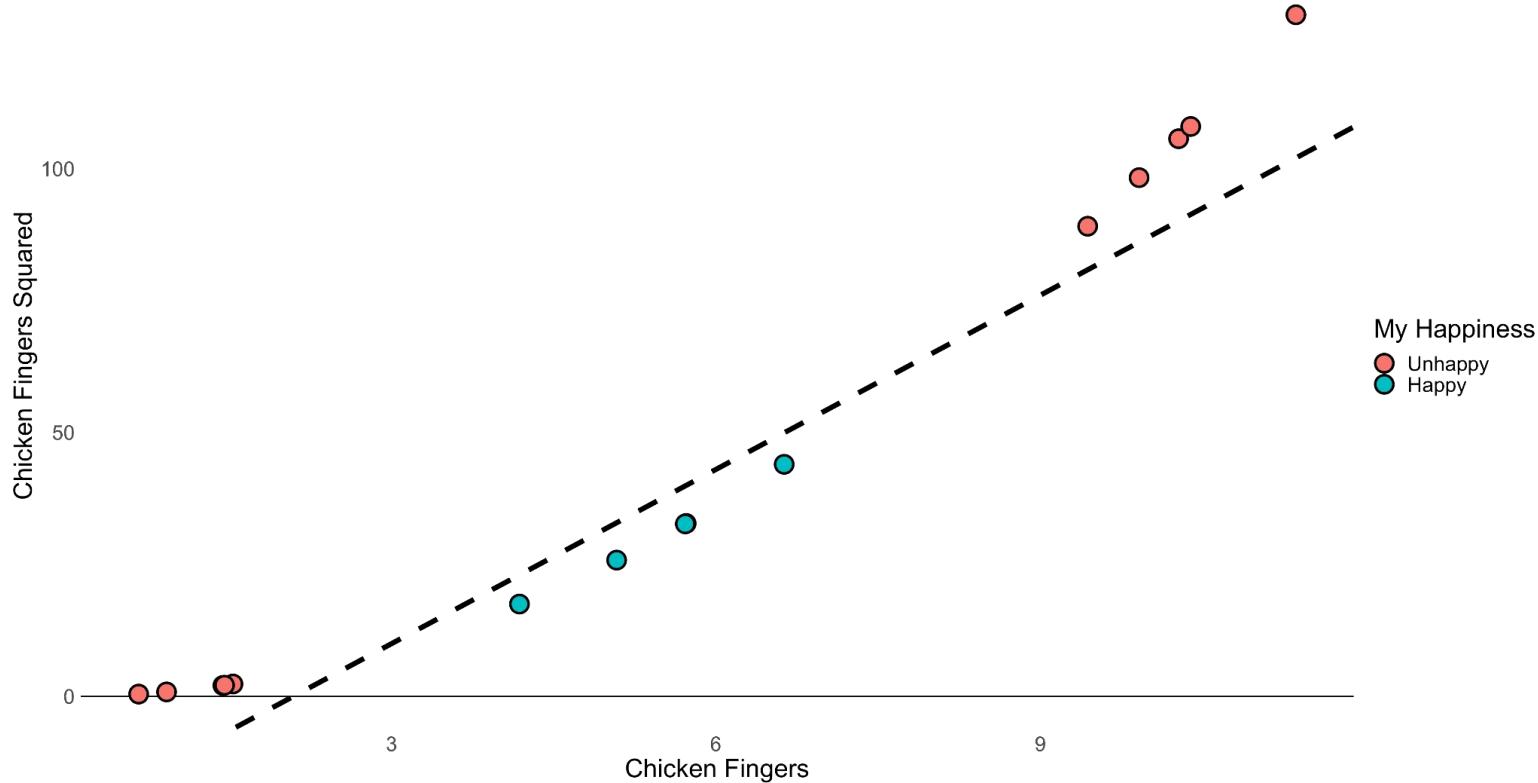
- We used a Polynomial Kernel with **r** (coefficient) and **d** (degree)
- Let's try $d = 2, r = \frac{1}{2}$

Polynomial Kernel

$$\underbrace{K(x, y)}_{\text{kernel}} = (x * y + r)^d$$

The Kernel Trick

Chelsea's Happiness When Eating Chicken Fingers

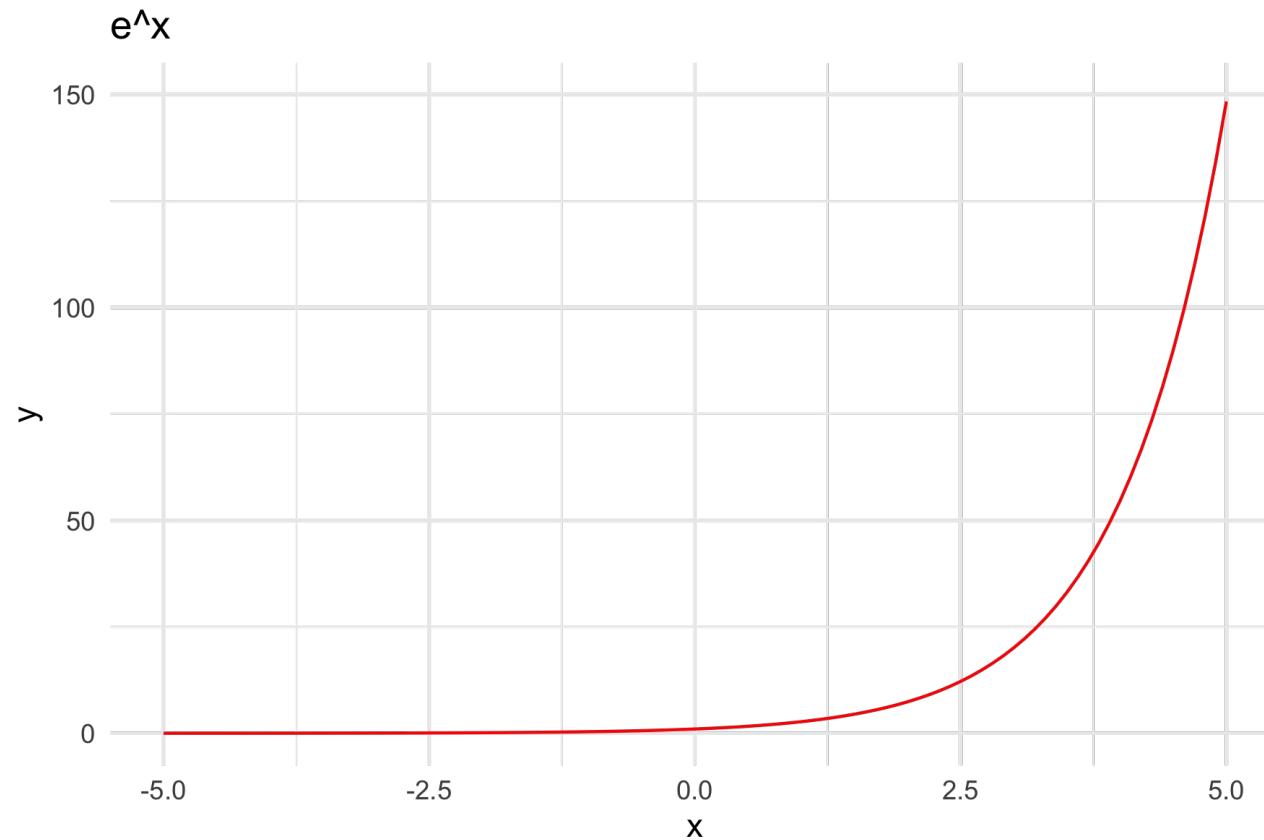


Radial Kernel (RBF)

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

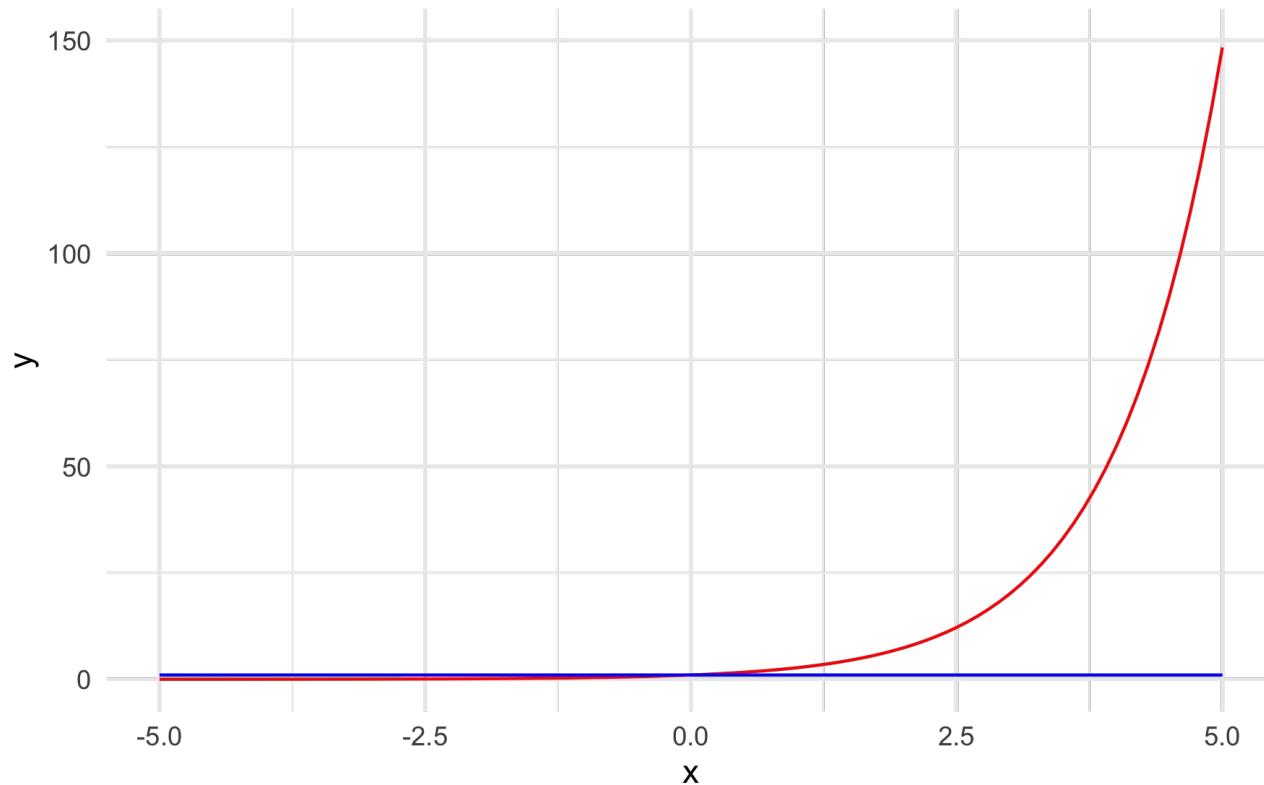
- Projects to *infinite dimensional* space
- Similar to a nearest neighbors classifier (points that are far away have less influence)

Taylor Series (visually)



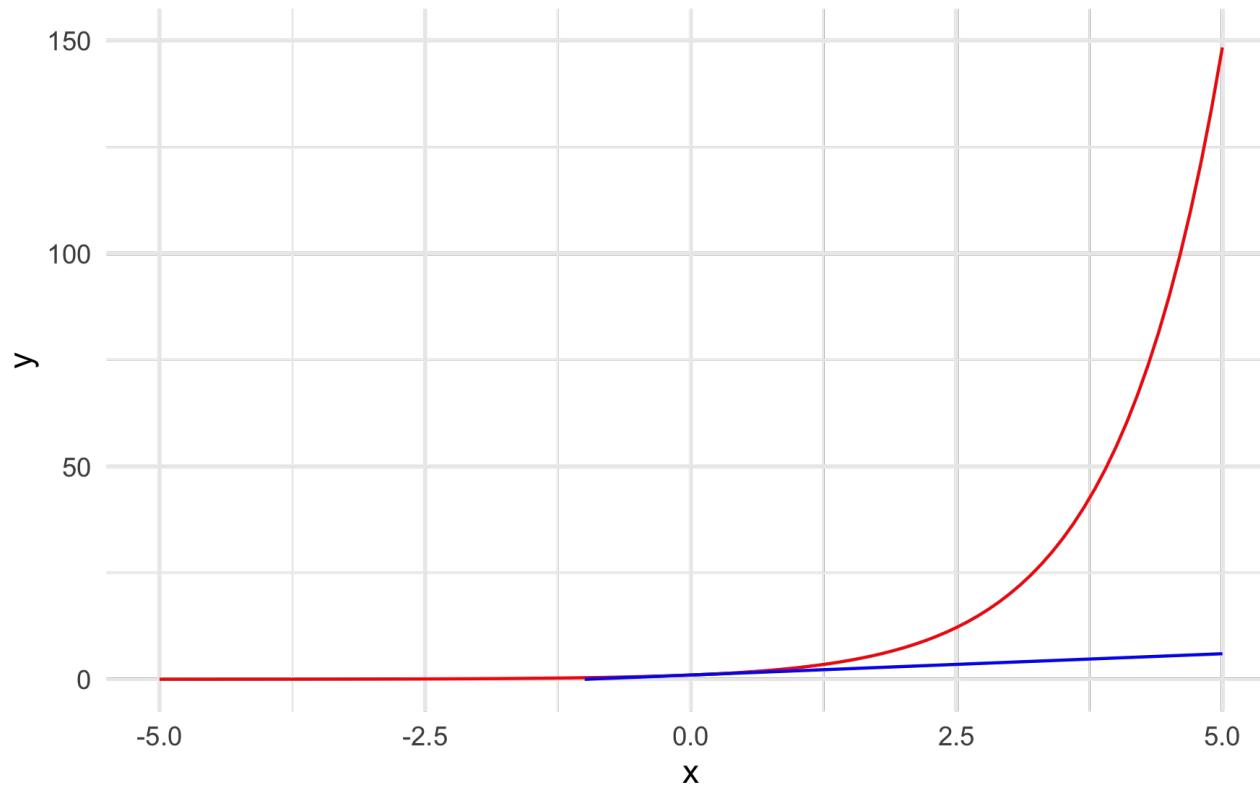
Taylor Series (visually)

e^x approximated by a 0-degree Taylor Polynomial



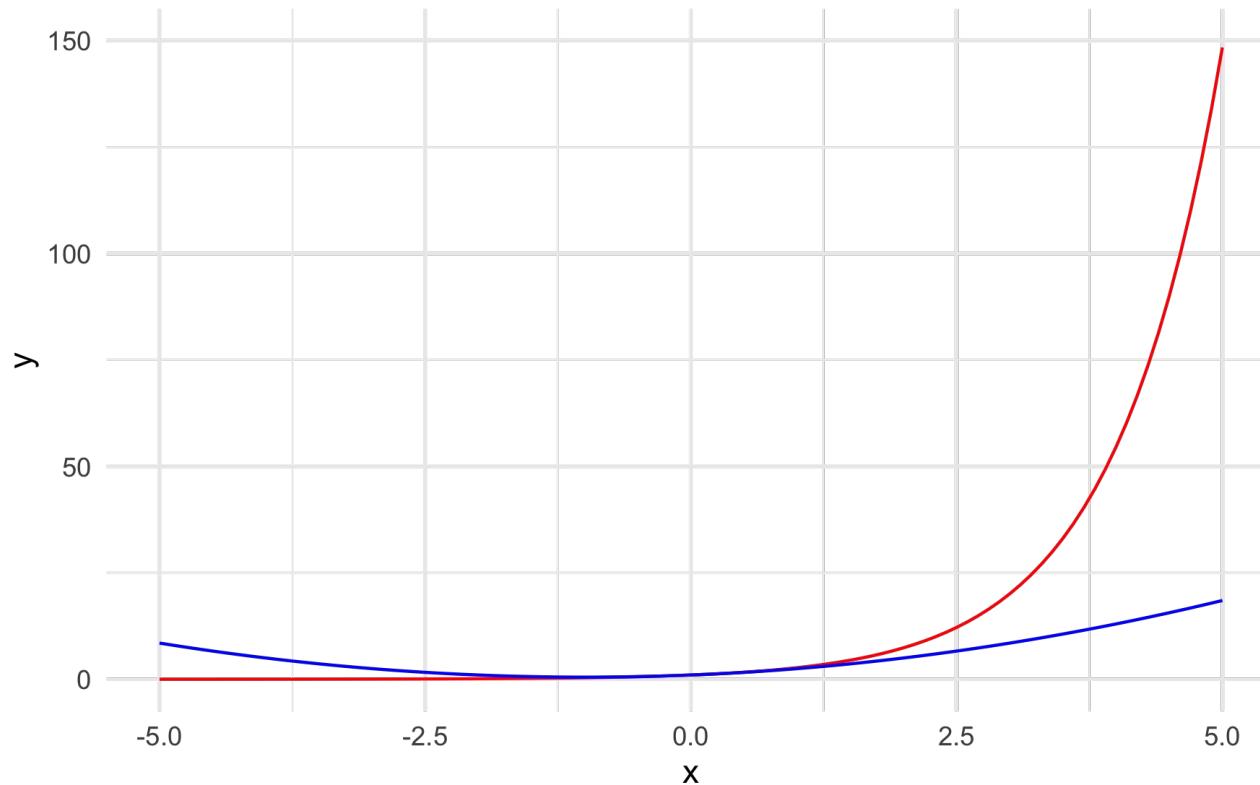
Taylor Series (visually)

e^x approximated by a 1-degree Taylor Polynomial



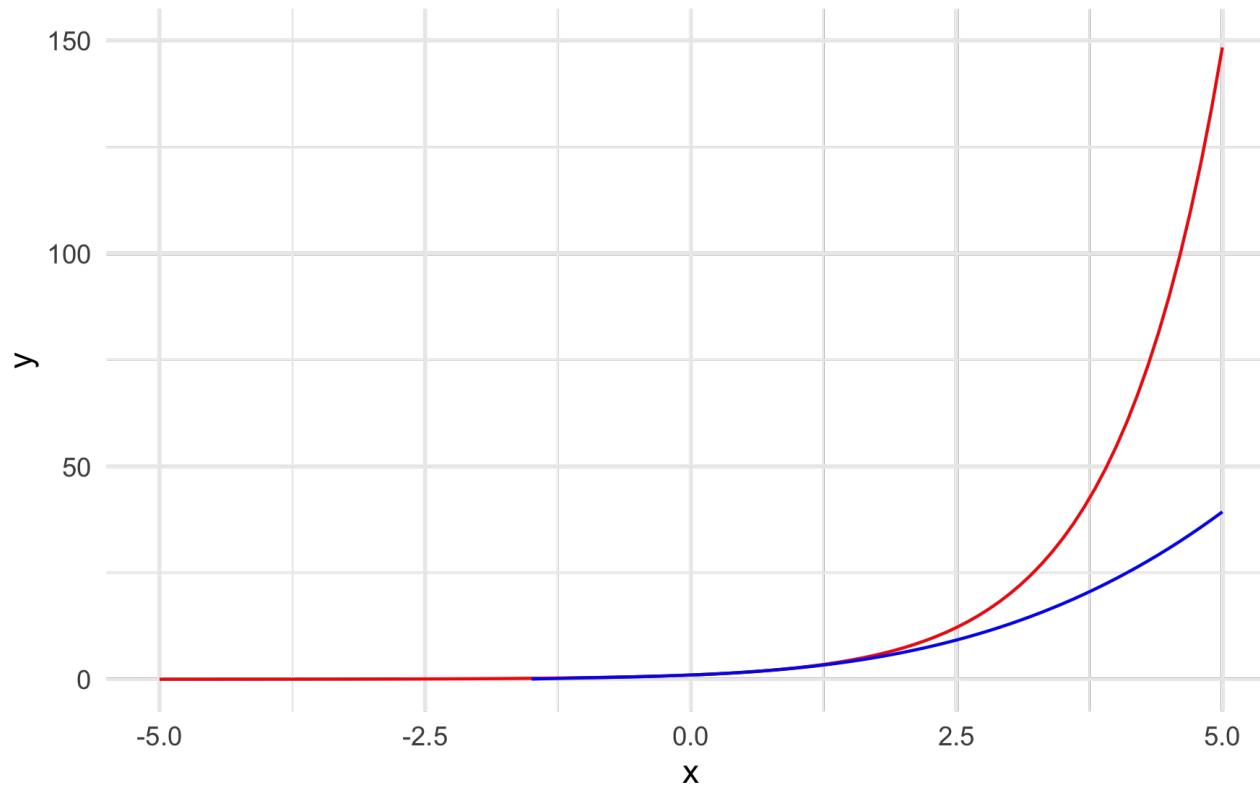
Taylor Series (visually)

e^x approximated by a 2-degree Taylor Polynomial



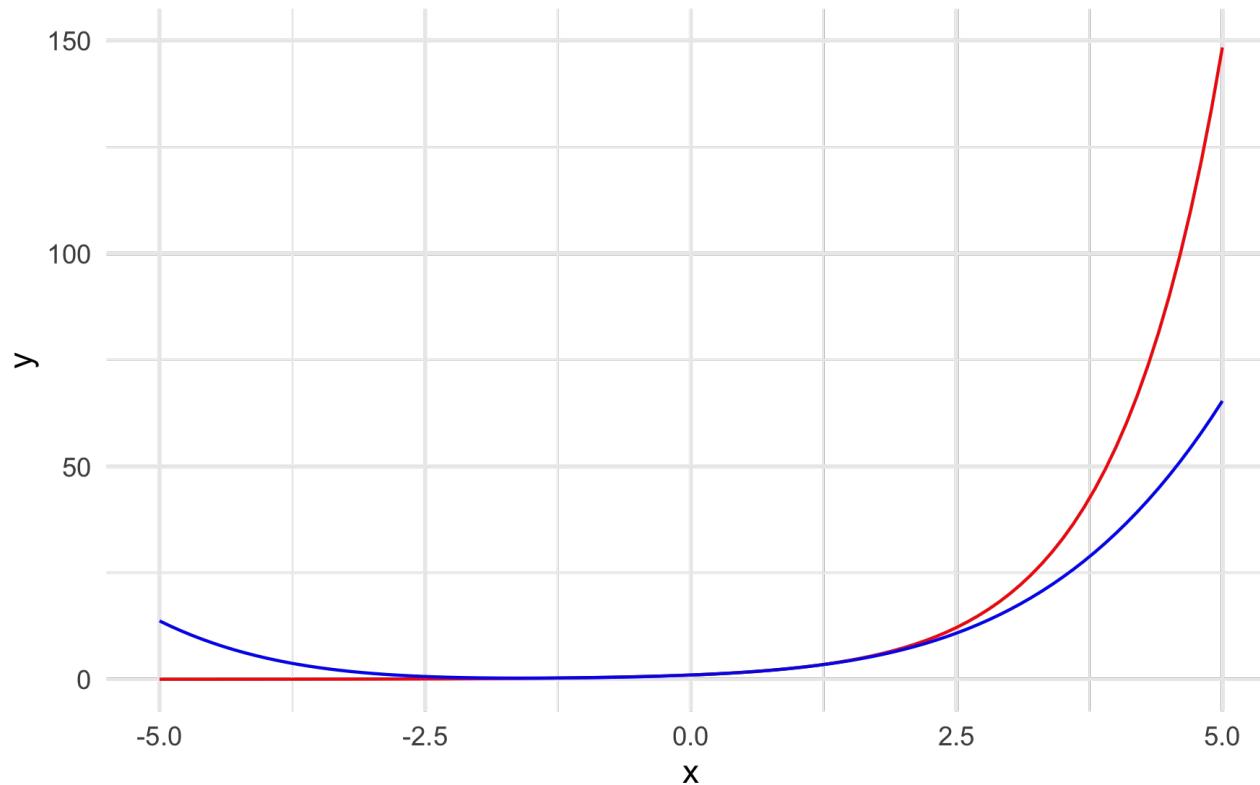
Taylor Series (visually)

e^x approximated by a 3-degree Taylor Polynomial



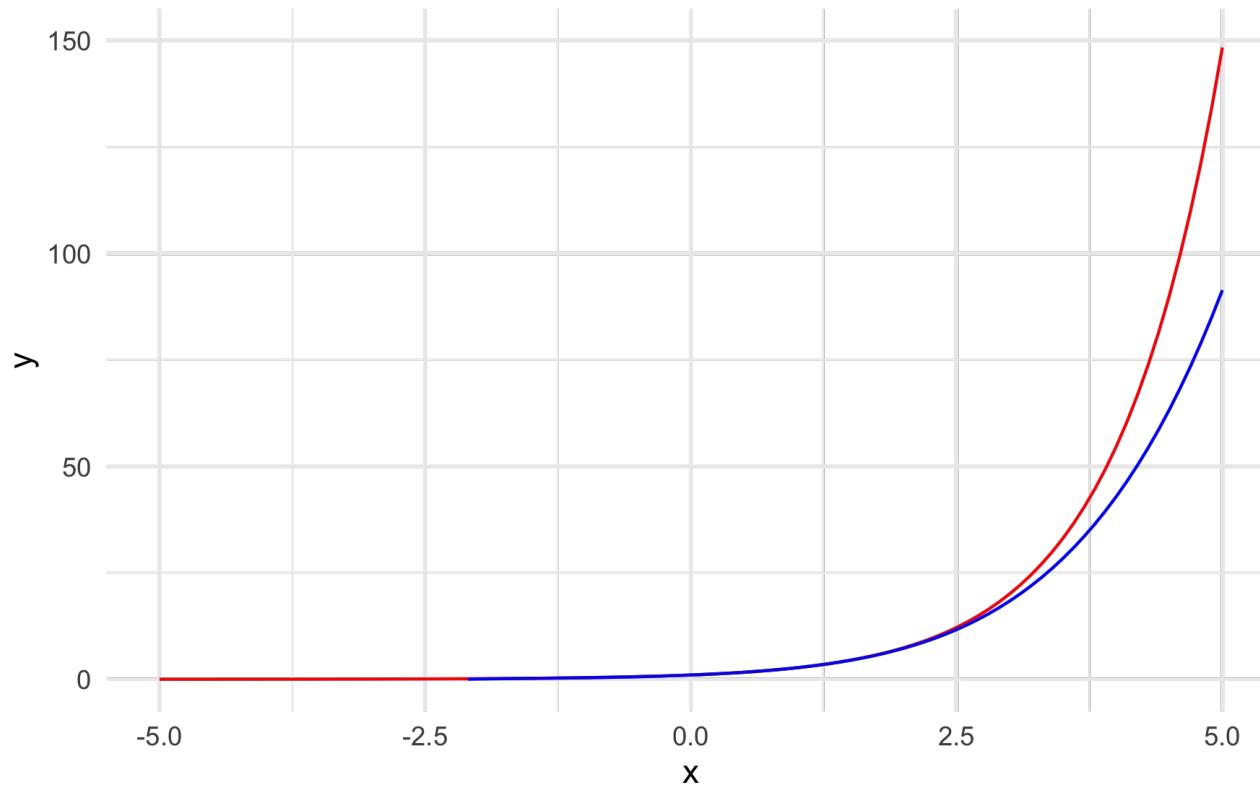
Taylor Series (visually)

e^x approximated by a 4-degree Taylor Polynomial



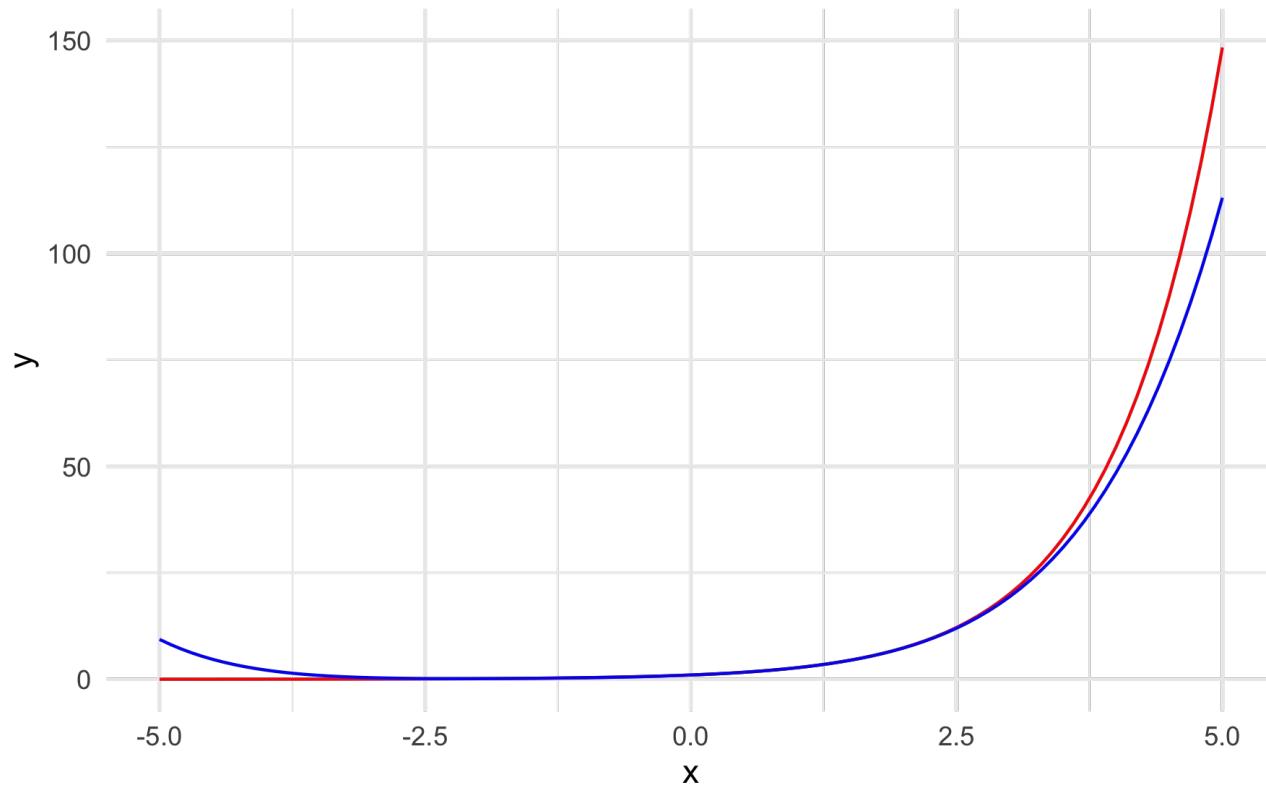
Taylor Series (visually)

e^x approximated by a 5-degree Taylor Polynomial



Taylor Series (visually)

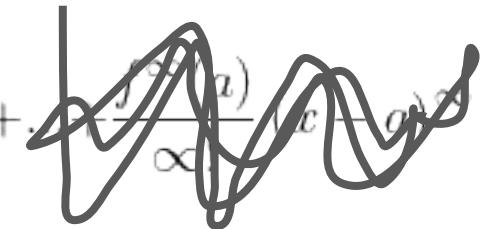
e^x approximated by a 6-degree Taylor Polynomial



Taylor Series

$$\sum_{n=0}^{\infty} \frac{f^n(a)}{n!} (x-a)^n = f(a) + \frac{f'(a)}{1} (x-a) + \frac{f''(a)}{2!} (x-a)^2 + \dots + \frac{f^\infty(a)}{\infty!} (x-a)^\infty$$

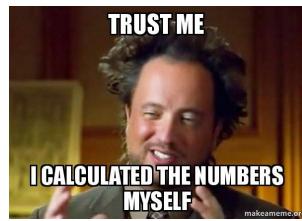
$$\sum_{n=0}^{\infty} \frac{f^n(a)}{n!} (x-a)^n = \underbrace{f(a) + \frac{f'(a)}{1} (x-a) + \frac{f''(a)}{2!} (x-a)^2}_{\text{use just this for approximation of } f(x)} + \dots$$



Radial Kernel (Taylor's Version)

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

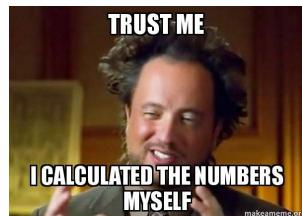
$$\underbrace{K(x, y)}_{\text{kernel}} = (x * y + r)^d$$



Radial Kernel (Taylor's Version)

$$K(x, y) = (x * y)^1 = x^1 y^1 = x^1 \cdot y^1$$

$$K(x, y) = (x * y)^2 = x^2 y^2 = x^2 \cdot y^2$$



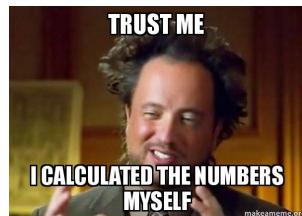
Radial Kernel (Taylor's Version)

$$K(x, y) = (x * y)^1 = x^1 y^1 = x^1 \cdot y^1$$

$$K(x, y) = (x * y)^2 = x^2 y^2 = x^2 \cdot y^2$$

$$x^1 y^1 + x^2 y^2 = (x^1, x^2) \cdot (y^1, y^2)$$

$$x^1 y^1 + x^2 y^2 + x^3 y^3 = (x^1, x^2, x^3) \cdot (y^1, y^2, y^3)$$



Radial Kernel (Taylor's Version)

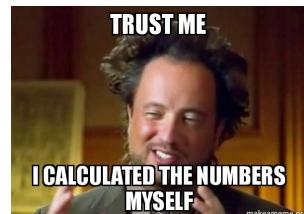
$$K(x, y) = (x * y)^1 = x^1 y^1 = x^1 \cdot y^1$$

$$K(x, y) = (x * y)^2 = x^2 y^2 = x^2 \cdot y^2$$

$$x^1 y^1 + x^2 y^2 = (x^1, x^2) \cdot (y^1, y^2)$$

$$x^1 y^1 + x^2 y^2 + x^3 y^3 = (x^1, x^2, x^3) \cdot (y^1, y^2, y^3)$$

$$x^1 y^1 + x^2 y^2 + x^3 y^3 + \dots + x^\infty y^\infty = (x^1, x^2, x^3, \dots, x^\infty) \cdot (y^1, y^2, y^3, \dots, y^\infty)$$



Radial Kernel (Taylor's Version)

$$K(x, y) = (x * y)^1 = x^1 y^1 = x^1 \cdot y^1$$

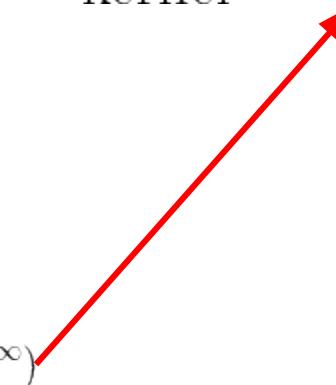
$$K(x, y) = (x * y)^2 = x^2 y^2 = x^2 \cdot y^2$$

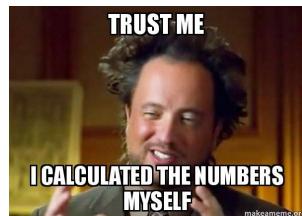
$$x^1 y^1 + x^2 y^2 = (x^1, x^2) \cdot (y^1, y^2)$$

$$x^1 y^1 + x^2 y^2 + x^3 y^3 = (x^1, x^2, x^3) \cdot (y^1, y^2, y^3)$$

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

$$x^1 y^1 + x^2 y^2 + x^3 y^3 + \dots + x^\infty y^\infty = (x^1, x^2, x^3, \dots, x^\infty) \cdot (y^1, y^2, y^3, \dots, y^\infty)$$





Radial Kernel (Taylor's Version)

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

$$e^{-\gamma(x-y)^2} = e^{-\gamma(x^2+y^2-2xy)}$$

$$e^{-\gamma(x^2+y^2)} e^{\gamma 2xy}$$

Radial Kernel (Taylor's Version)

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

$$e^{-\gamma(x-y)^2} = e^{-\gamma(x^2+y^2-2xy)}$$

$$e^{-\gamma(x^2+y^2)} e^{\gamma 2xy}$$

$$e^{-\frac{1}{2}(x^2+y^2)} e^{xy}$$

Radial Kernel (Taylor's Version)

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

$$e^{-\gamma(x-y)^2} = e^{-\gamma(x^2+y^2-2xy)}$$

$$e^{-\gamma(x^2+y^2)} e^{\gamma 2xy}$$

$$e^{-\frac{1}{2}(x^2+y^2)} \boxed{e^{xy}}$$

Radial Kernel (Taylor's Version)

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

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

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

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

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

$$x^0y^0 + x^1y^1 + x^2y^2 + \dots + x^\infty y^\infty = (1, x, x^2, \dots, x^\infty) \cdot (1, y, y^2, \dots, y^\infty)$$

Radial Kernel (Taylor's Version)

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

$$e^{xy} = \boxed{1} + \frac{1}{1!}(xy) + \frac{1}{2!}(xy)^2 + \frac{1}{3!}(xy)^3 + \dots + \frac{1}{\infty!}(xy)^\infty$$


$$\boxed{x^0 y^0} + x^1 y^1 + x^2 y^2 + \dots + x^\infty y^\infty = (1, x, x^2, \dots, x^\infty) \cdot (1, y, y^2, \dots, y^\infty)$$

Radial Kernel (Taylor's Version)

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

$$e^{xy} = \boxed{1} + \frac{1}{1!}\boxed{(xy)} + \frac{1}{2!}(xy)^2 + \frac{1}{3!}(xy)^3 + \dots + \frac{1}{\infty!}(xy)^\infty$$

$$\boxed{x^0y^0} + \boxed{x^1y^1} + x^2y^2 + \dots + x^\infty y^\infty = (1, x, x^2, \dots, x^\infty) \cdot (1, y, y^2, \dots, y^\infty)$$

Radial Kernel (Taylor's Version)

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

$$e^{xy} = \boxed{1} + \frac{1}{1!}\boxed{(xy)} + \frac{1}{2!}\boxed{(xy)^2} + \frac{1}{3!}(xy)^3 + \dots + \frac{1}{\infty!}(xy)^\infty$$

$$\boxed{x^0y^0} + \boxed{x^1y^1} + \boxed{x^2y^2} + \dots + x^\infty y^\infty = (1, x, x^2, \dots, x^\infty) \cdot (1, y, y^2, \dots, y^\infty)$$

$$\underbrace{K(x, y)}_{\text{kernel}} = e^{-\gamma(x-y)^2}$$

Radial Kernel (Taylor's Version)

$$e^{xy} = \boxed{1} + \frac{1}{1!}\boxed{(xy)} + \frac{1}{2!}\boxed{(xy)^2} + \frac{1}{3!}(xy)^3 + \dots + \frac{1}{\infty!}\boxed{(xy)^\infty}$$
$$\boxed{x^0y^0} + \boxed{x^1y^1} + \boxed{x^2y^2} + \dots + \boxed{x^\infty y^\infty} = (1, x, x^2, \dots, x^\infty) \cdot (1, y, y^2, \dots, y^\infty)$$

Radial Kernel (Taylor's Version)

$$K(x, y) = e^{-\gamma(x-y)^2}$$

kernel

$e^{xy} = 1 + \frac{1}{1!}(x^0y^0 + x^1y^1 + \dots)$

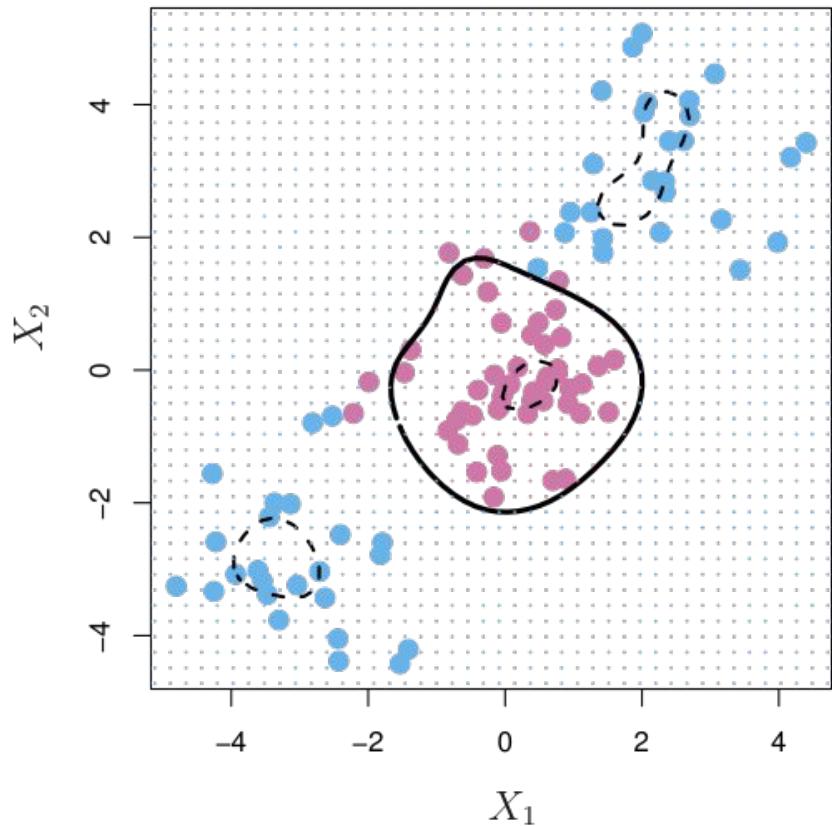
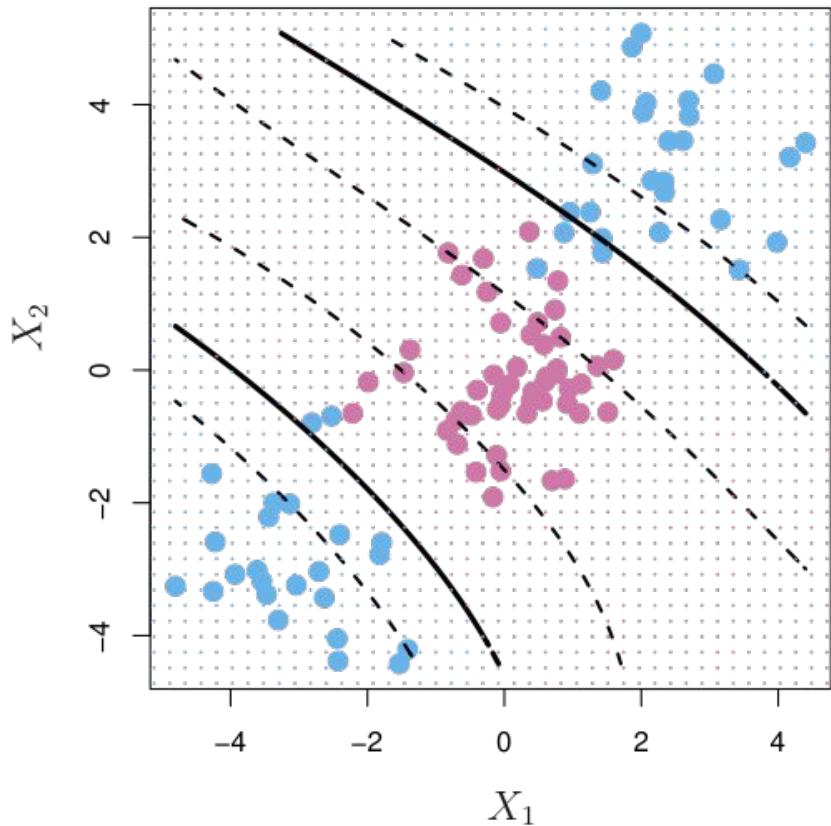
Radial Basis Kernels have coordinates for infinite dimensions!

Support Vector Machines Part 3:

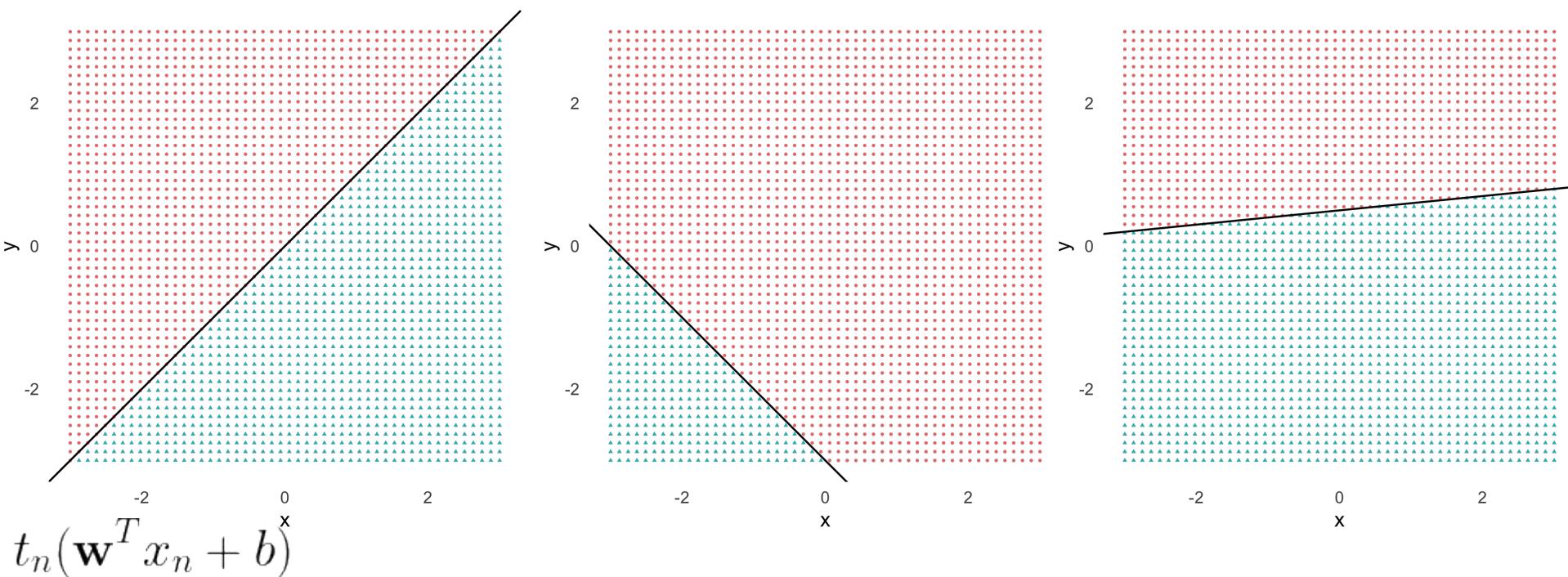
$$e^{-\gamma(a-b)^2}$$

...The Radial Kernel!!!

Radial Basis Kernel Example



Kernels



$$t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b)$$

Kernels

$$\underbrace{y(\mathbf{w}) = \sum_{n=1}^N a_n t_n \phi(x) \cdot \phi(x_n) + b}_{\text{formula to classify new data point } x}$$

$$\underbrace{y(\mathbf{w}) = \sum_{n=1}^N a_n t_n \underbrace{K(x, y)}_{\text{kernel}} + b}_{\text{formula to classify new data point } x}$$