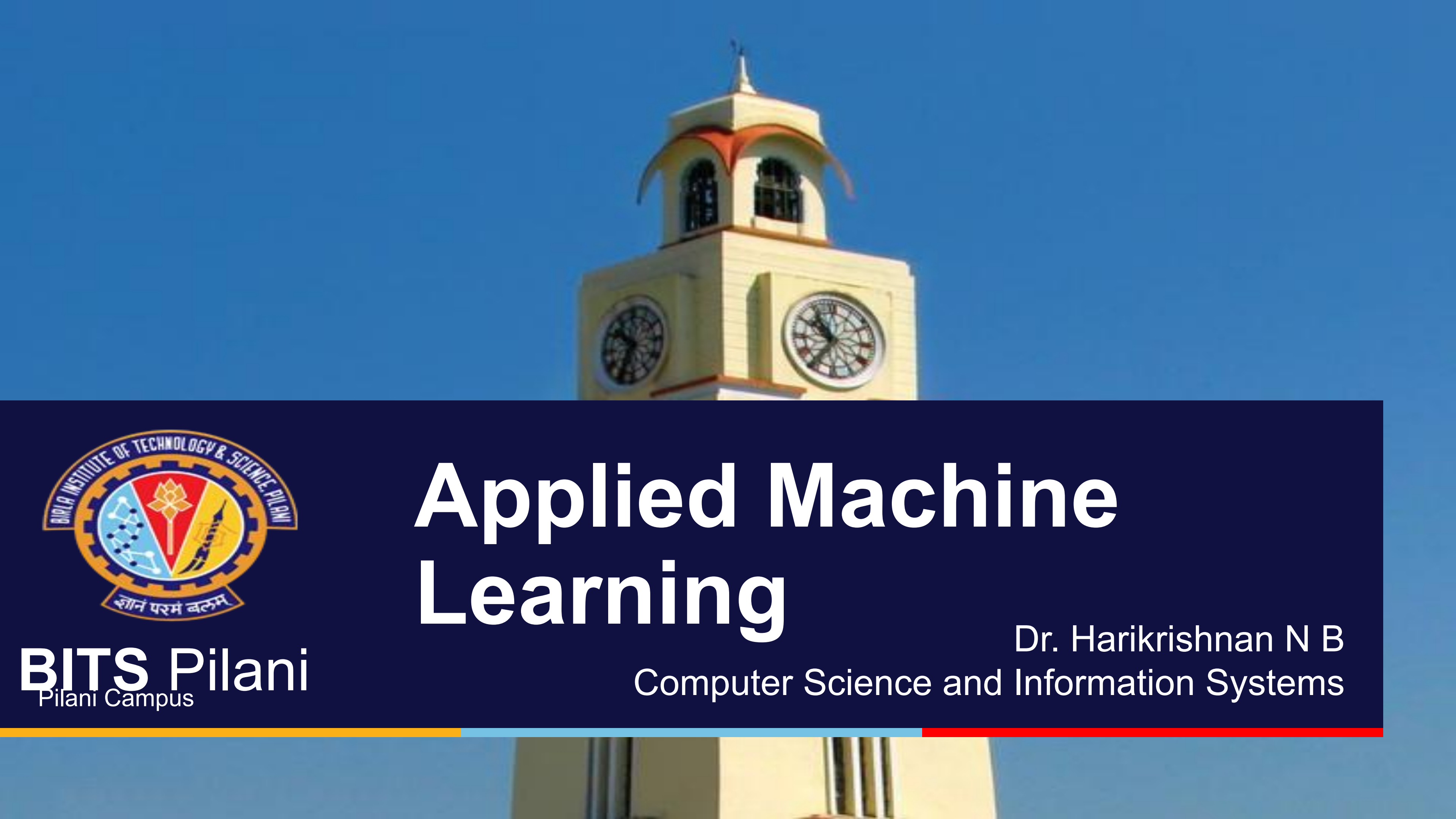


Applied Machine Learning

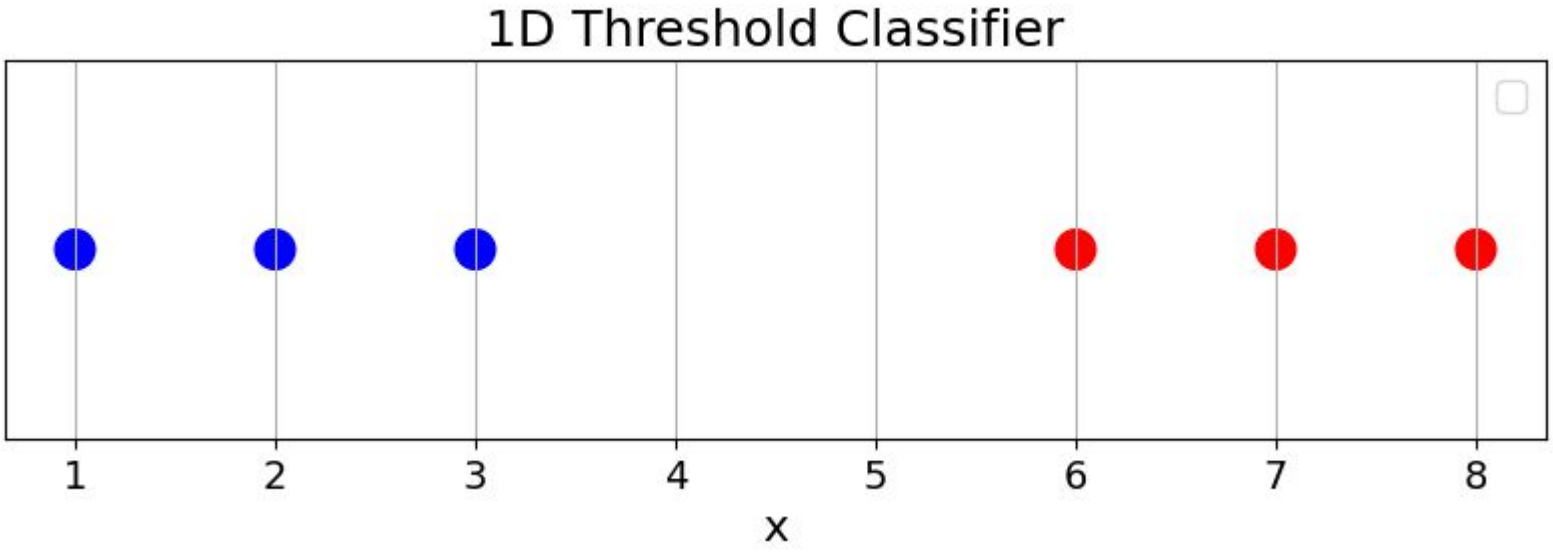
BITS Pilani
Pilani Campus

Dr. Harikrishnan N B
Computer Science and Information Systems

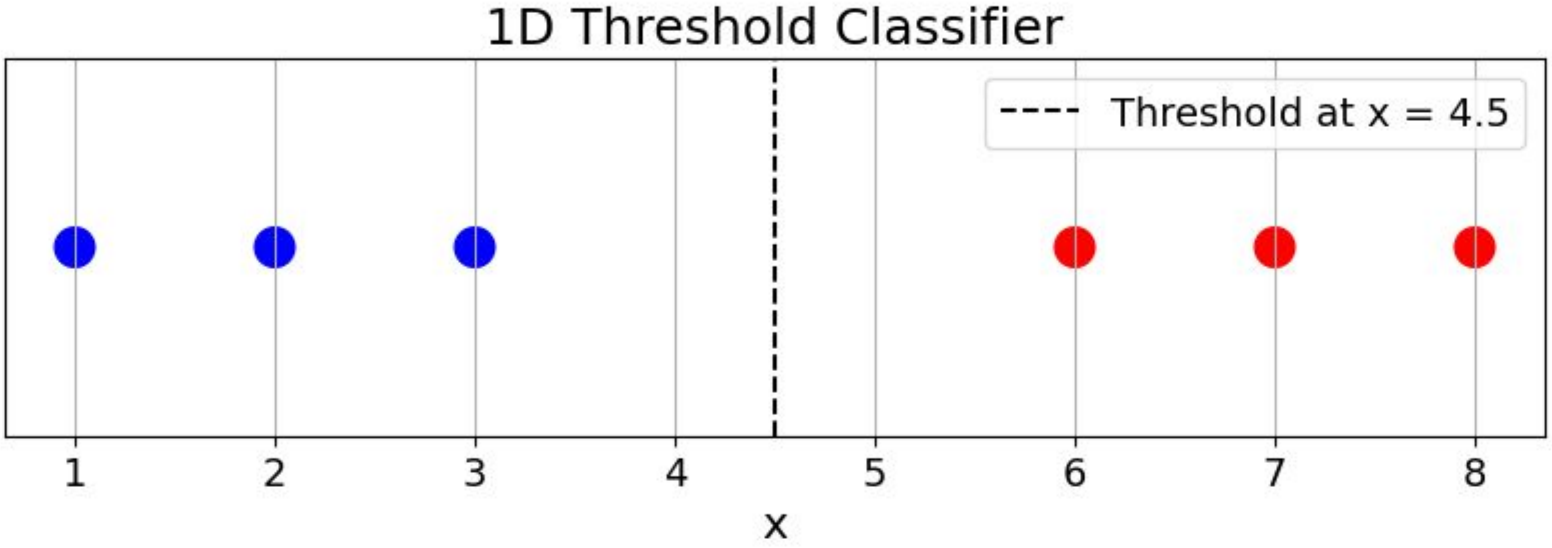
SE ZG568 / SS ZG568, Applied Machine Learning Lecture No. 12 [20 April 2025]

Perceptron and
Support Vector Machine

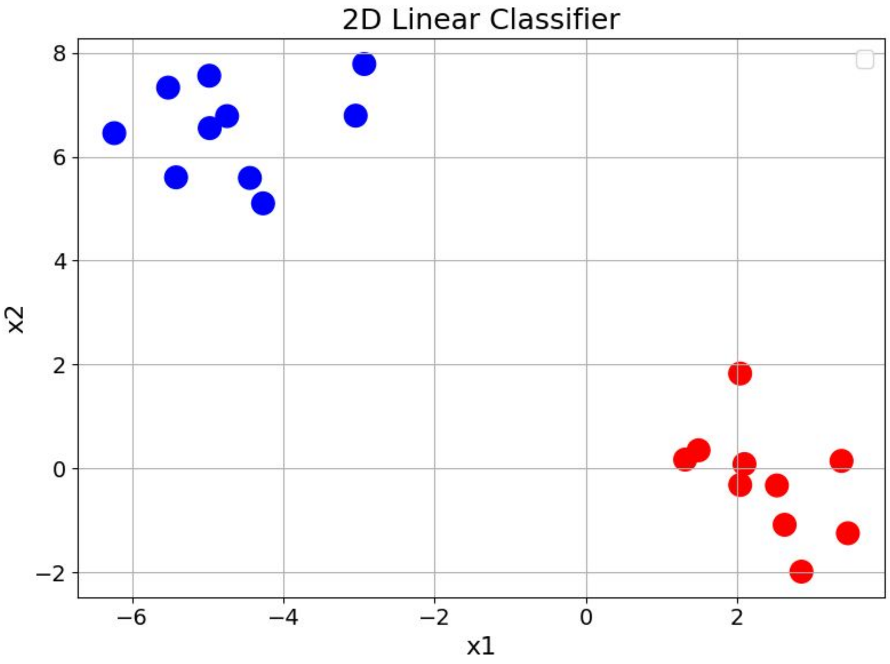
Points lying on x-axis: Come up with a classification rule!!!



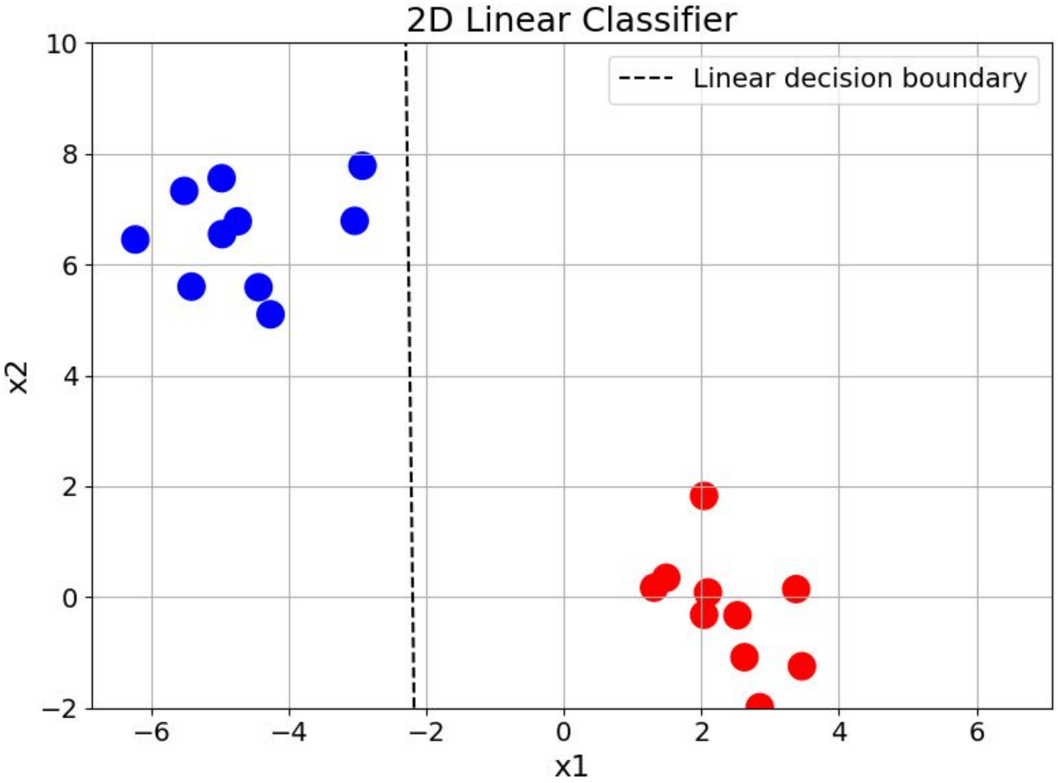
Points lying on x-axis: Come up with a classification rule!!!



Let’s move to 2D: Come up with a classification rule using a line!!!

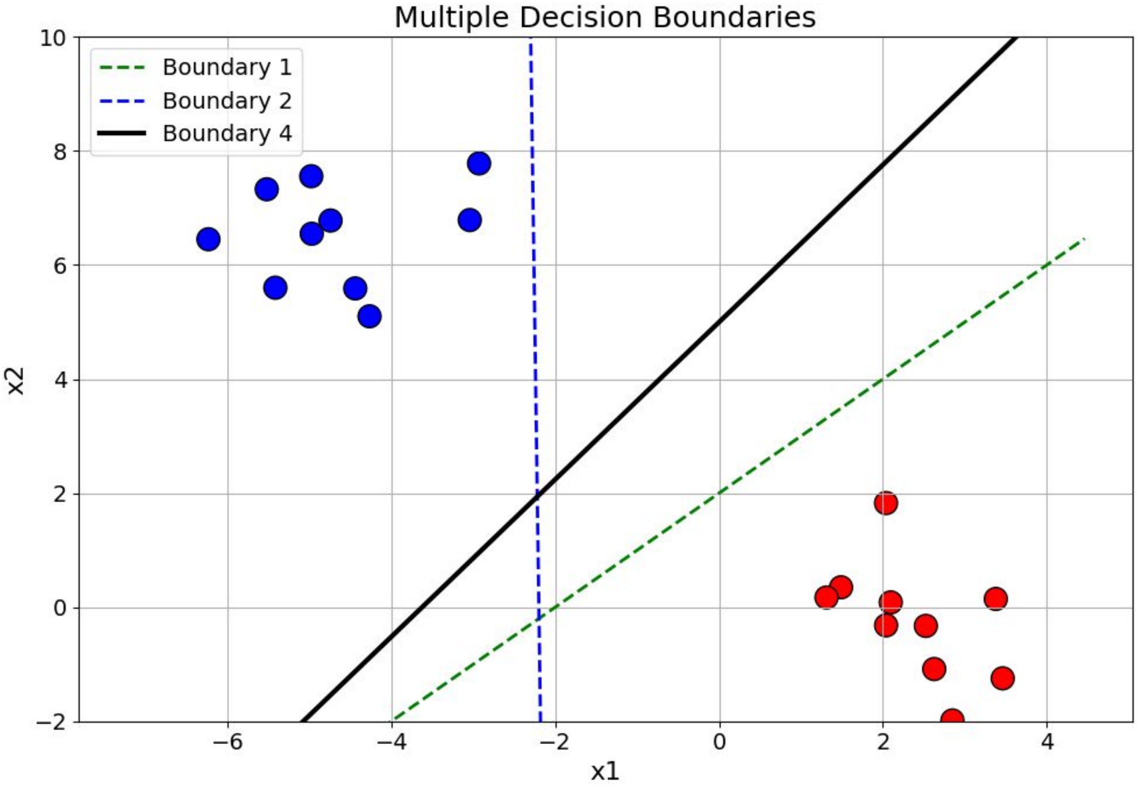


Let's move to 2D: Come up with a classification rule using a line!!!



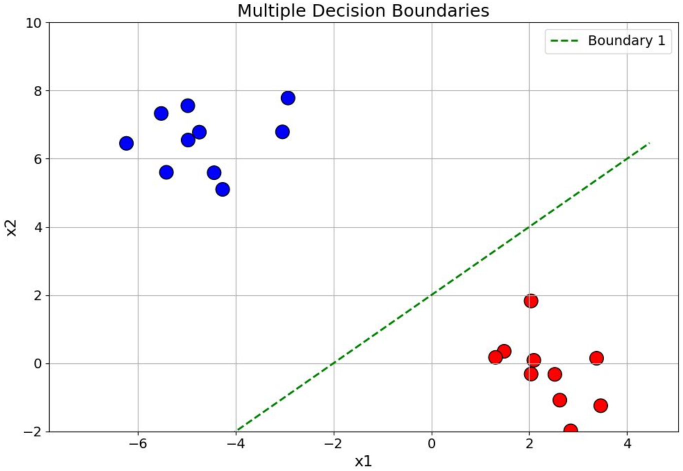
$x_2 = -100x_1 -220$

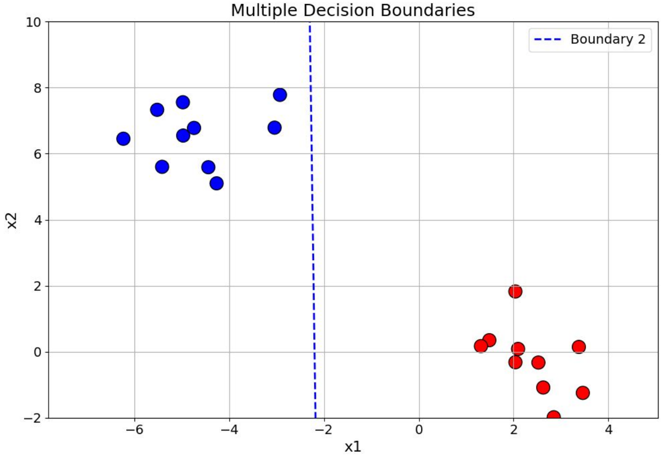
This is not the only classification rule!!!

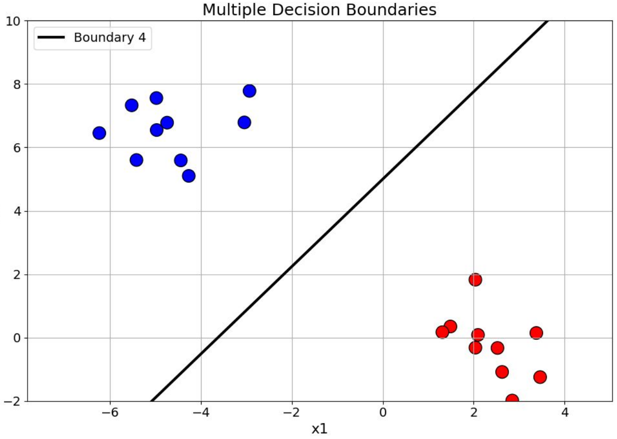


Which Decision Boundary will you choose?

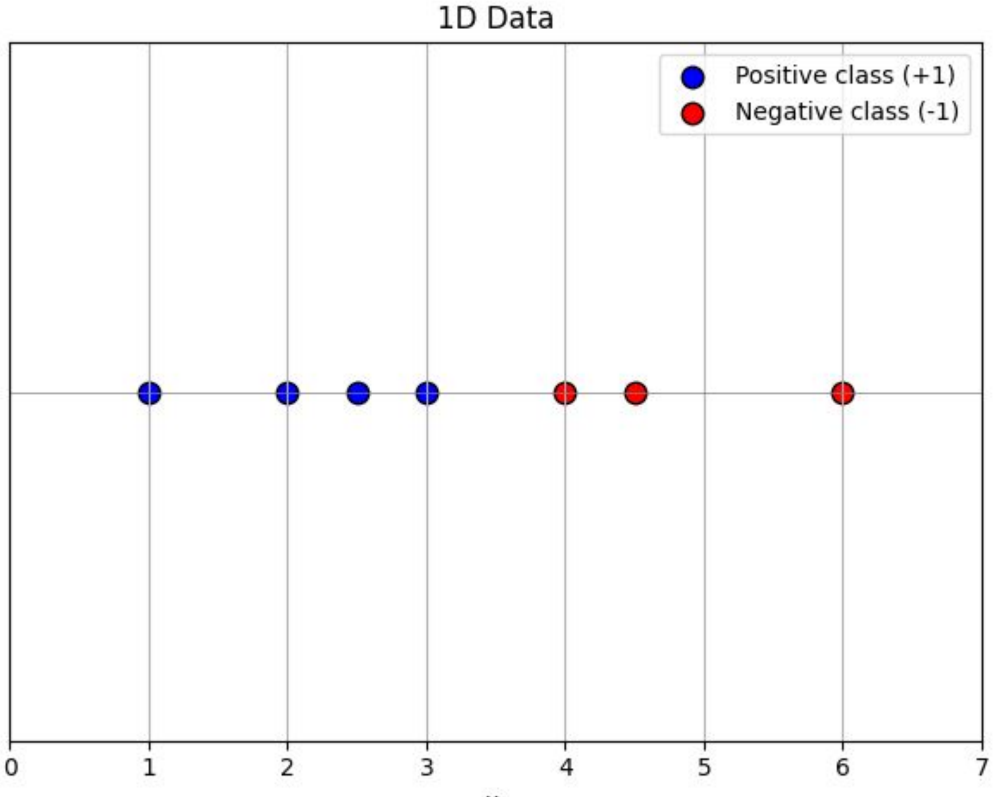
Which decision boundary would you choose?

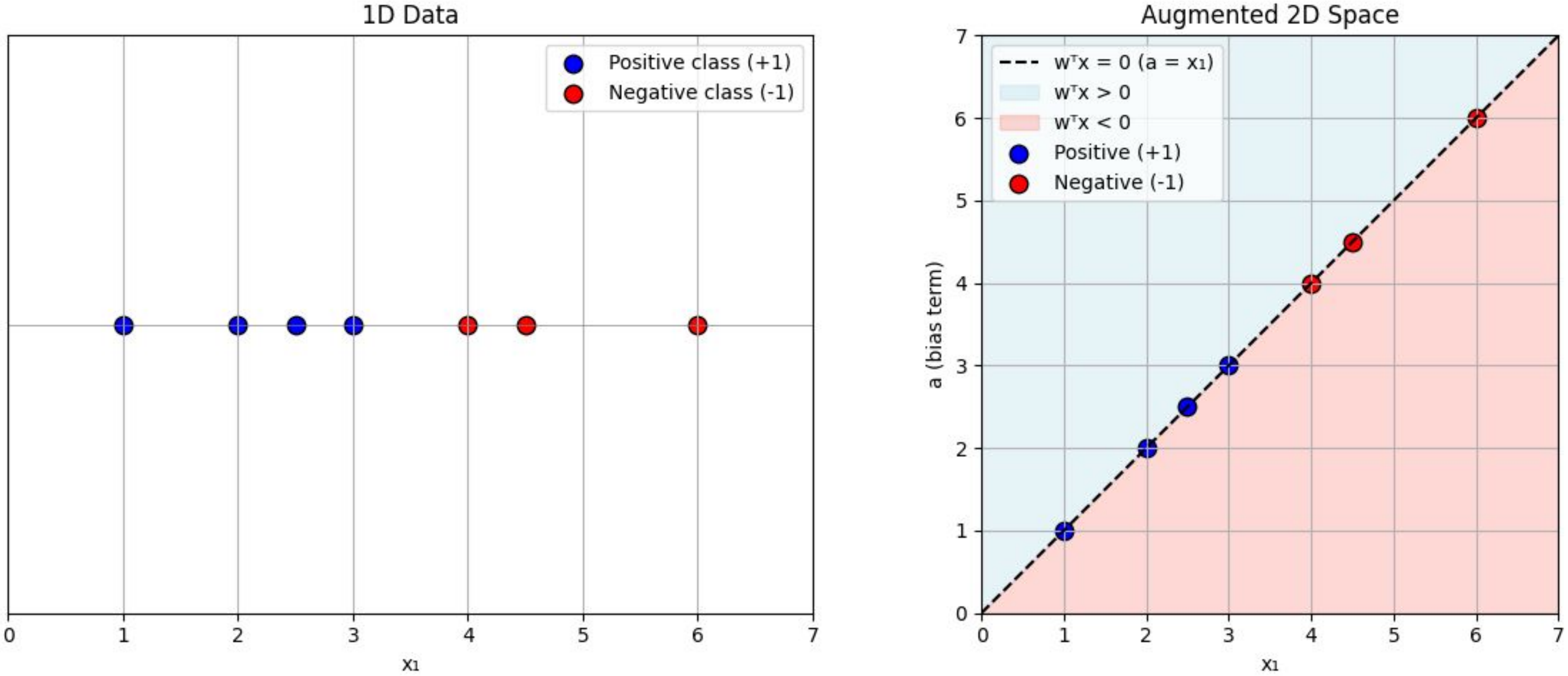




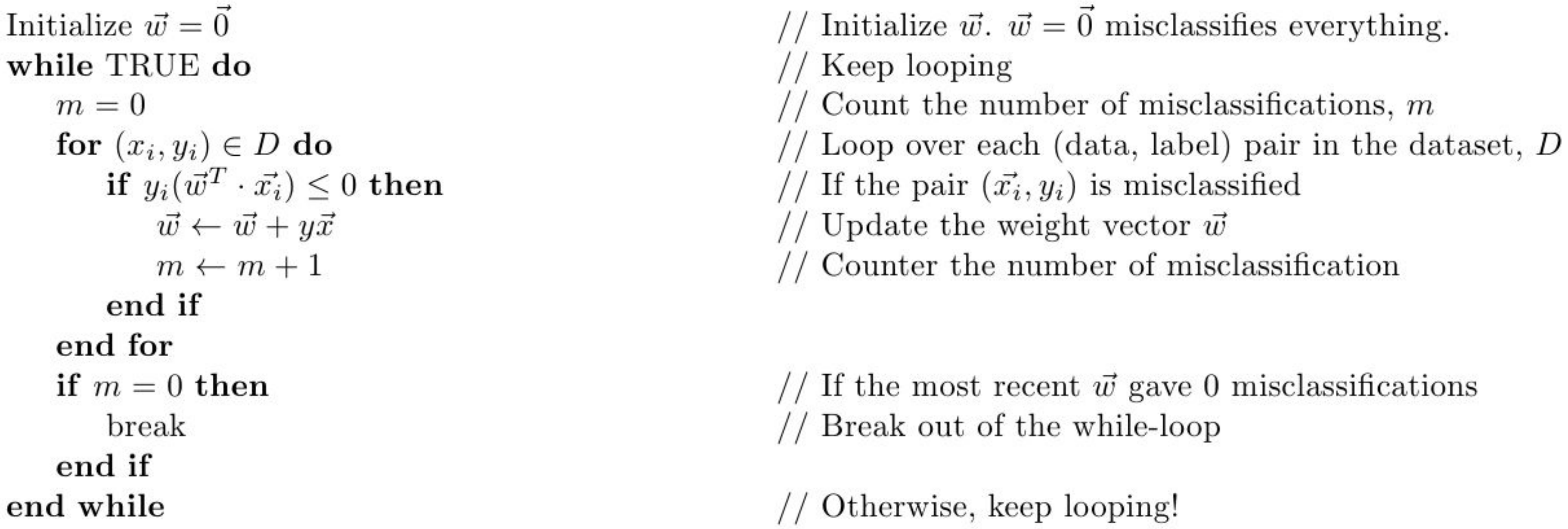


Basics of Equation of Line


1D Data
Positive class (+1)
Negative class (-1)
0
1
2
3
4
5
6
7


1D Data
Positive class (+1)
Negative class (-1)
0
1
2
3
4
5
6
7
x_1
Augmented 2D Space
$w^T x = 0$ ($a = x_1$)
$w^T x > 0$
$w^T x < 0$
Positive (+1)
Negative (-1)
a (bias term)
0
1
2
3
4
5
6
7
x_1

Perceptron



```
Initialize $\vec{w} = \vec{0}$                          // Initialize $\vec{w}$. $\vec{w} = \vec{0}$ misclassifies everything.
while TRUE do                                           // Keep looping
    $m = 0$                                             // Count the number of misclassifications, $m$
    for $(x_i, y_i) \in D$ do                           // Loop over each (data, label) pair in the dataset, $D$
        if $y_i(\vec{w}^T \cdot \vec{x_i}) \leq 0$ then // If the pair $(\vec{x_i}, y_i)$ is misclassified
            $\vec{w} \leftarrow \vec{w} + y\vec{x}$     // Update the weight vector $\vec{w}$
            $m \leftarrow m + 1$                        // Counter the number of misclassification
        end if
    end for
    if $m = 0$ then                                     // If the most recent $\vec{w}$ gave 0 misclassifications
        break                                           // Break out of the while-loop
    end if
end while                                               // Otherwise, keep looping!
```

```
Initialize $\vec{w} = \vec{0}$
while TRUE do
    $m = 0$
    for $(x_i, y_i) \in D$ do
        if $y_i(\vec{w}^T \cdot \vec{x_i}) \leq 0$ then
            $\vec{w} \leftarrow \vec{w} + y\vec{x}$
            $m \leftarrow m + 1$
        end if
    end for
    if $m = 0$ then
        break
    end if
end while
```

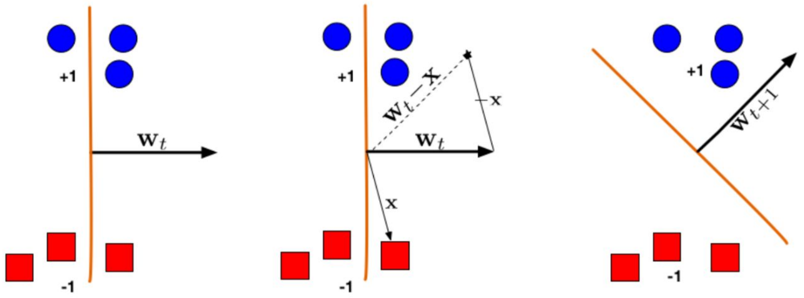
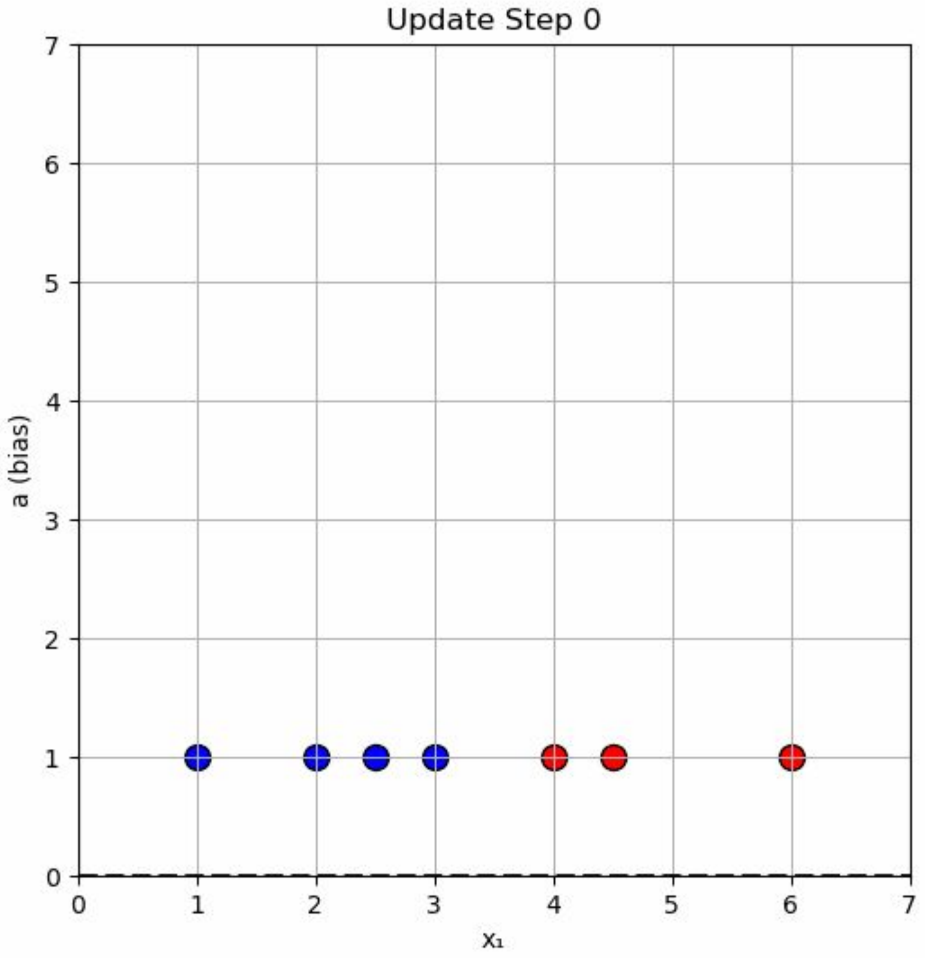


*Illustration of a Perceptron update. (Left:) The hyperplane defined by $\mathbf{w}_t$ misclassifies one red (-1) and one blue (+1) point. (Middle:) The red point $\mathbf{x}$ is chosen and used for an update. Because its label is -1 we need to **subtract** $\mathbf{x}$ from $\mathbf{w}_t$. (Right:) The udpated hyperplane $\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{x}$ separates the two classes and the Perceptron algorithm has converged.*


Update Step 0
a (bias)
0
1
2
3
4
5
6
7
x₁
0
1
2
3
4
5
6
7