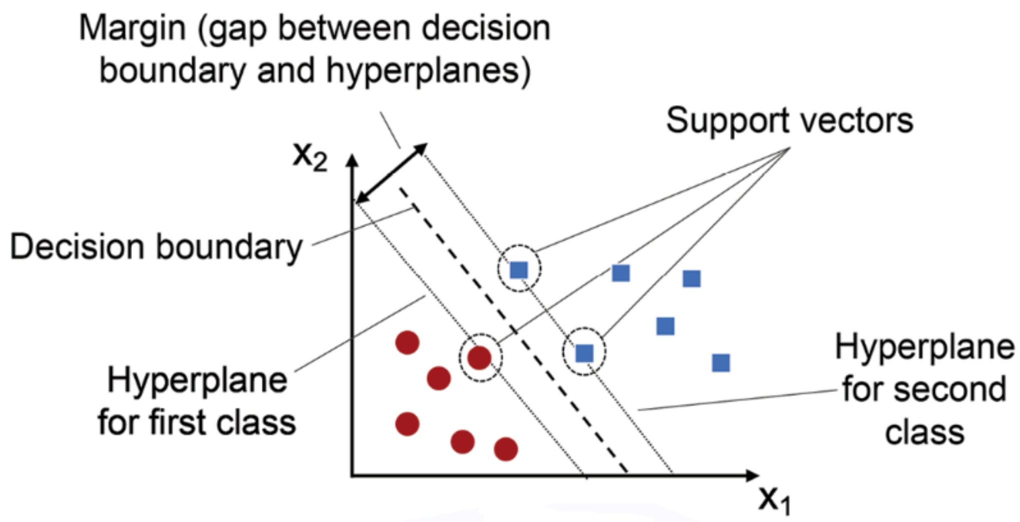
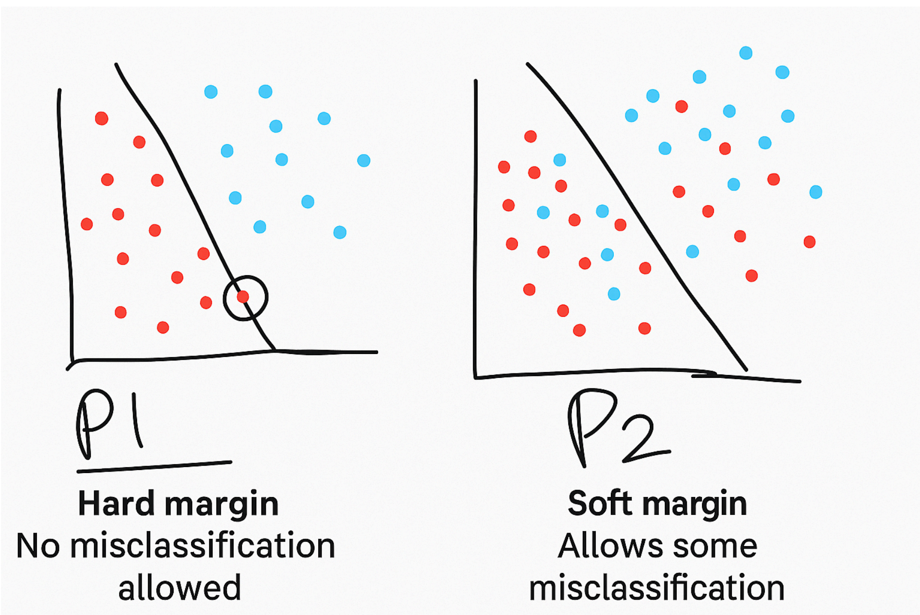
<u>**SVM**</u>

For linearly separable data

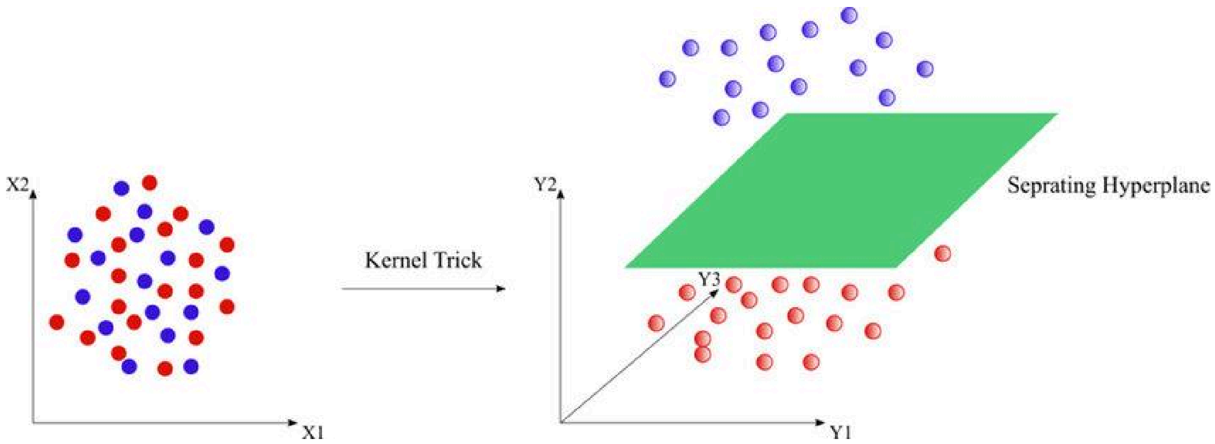




P1 is overfitting because the model learns the training data too well so it will perform good on training data but it will fail with test data.

P2 is underfitting and it will fail both in train and test data.

For non-linearly separable data



SVM – Support Vector Machine

- **Type: Supervised Machine Learning Algorithm**
- **Works on: Both Classification and Regression problems**

Types:

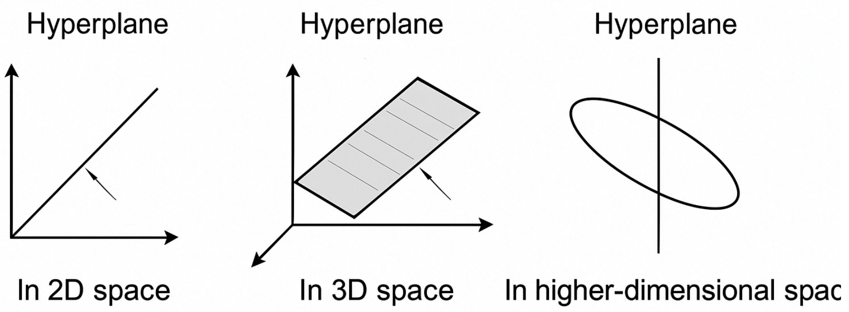
- **SVC (Support Vector Classifier):**
  Used when SVM deals with classification problems.
- **SVR (Support Vector Regressor):**
  Used when SVM deals with regression problems (from Python's point of view).

Key Terminologies:

- **Support Vector: Data points that are closest to the decision boundary (hyperplane).**

- **Margin: The distance between the hyperplane and the nearest data points (support vectors).**

- **Hyperplane: A hyperplane is a decision boundary that separates different classes in an SVM.**
  - **In a 2D space, it's a line; in 3D, it's a plane; and in higher dimensions, it's called a hyperplane.**
  - **The SVM actually finds the optimal hyperplane that maximizes the margin between the two classes.**



- **Hard Margin: Strict separation between classes without allowing misclassification (used when data is linearly separable).**

- **Soft Margin: Allows some misclassification for better generalization (used when data is not perfectly separable).**

- **Kernel: A function used to transform data into a higher-dimensional space to make it separable (e.g., linear,** polynomial, RBF).

- **C (Cost parameter): Controls the trade-off between maximizing the margin and minimizing classification errors.**

Interpretation

C Value	Behavior	Effect
Very High (e.g., 100)	Tight margin, strict classification	May overfit (too sensitive to noise/outliers)
Very Low (e.g., 0.01)	Wide margin, tolerant to errors	May underfit (too simple)
Moderate (1.0)	Balanced margin	Good trade-off between bias and variance

- **Gamma (Kernel Coefficient):**
  Defines how far the influence of a single training point reaches.

  High gamma → focus on nearby points (tight boundary, overfit risk)

  Low gamma → smoother boundary (underfitting risk)

Interpretation

Gamma Value	Behavior	Effect

Low Gamma (e.g., 0.001)	**Each point's influence spreads far**	**Smooth, simple boundary → can underfit**
High Gamma (e.g., 1 or 10)	**Each point affects only nearby points**	**Complex, tight boundary → can overfit**
'scale' (auto)	**Automatically balanced**	**Adapts to data → good starting point**

💡 **Gamma Intuition (Lamp Example)**

Each data point in SVM is like a lamp that spreads light around it:

- **Low Gamma: Each lamp lights up a large area, so many lamps overlap — the model creates a smooth, simple boundary that may miss small details (underfitting).**
- **High Gamma: Each lamp lights up only a tiny area, so the model focuses on every single point — creating a very tight, wiggly boundary (overfitting).**
- **Medium Gamma ('scale'): Each lamp lights up just enough area — making the boundary balanced and generalizing well.**

In short - Gamma controls how far the "influence" of each data point spreads when SVM tries to draw the boundary.

What is a Kernel in SVM?

A **kernel** is a **mathematical function** that transforms your original data into a **higher-dimensional space**, so that data that was **not linearly separable** before can now be separated by a hyperplane.

> Think of it like adding an extra "dimension" to make a curve look like a line.

Types of Kernel

Kernel Name	Equation / Idea	When to Use	Shape of Decision Boundary
1. Linear Kernel	$K(x, y) = x^T y$	When data is **linearly separable**	Straight line
2. Polynomial Kernel	$K(x, y) = (x^T y + c)^d$	When relationship between features is **polynomial** (curved but smooth)	Curved boundary (depends on degree)
3. **RBF (Radial Basis Function)** or **Gaussian Kernel**	(K(x, y) = \exp(-\gamma		x - y
4. Sigmoid Kernel	$K(x, y) = \tanh(\alpha x^T y + c)$	When data resembles **neural network-like patterns**	S-shaped boundary
5. Custom Kernel	User-defined (combination of above)	For **specific domains** like text or images	Depends on function

Intuitive Understanding

- **Linear Kernel:** Keeps the data as-is → best for simple, straight separations.
- **Polynomial Kernel:** Adds curvature → helps when data has curved but smooth boundaries.
- **RBF Kernel:** Adds infinite dimensions → great for complex, circular or irregular clusters.
- **Sigmoid Kernel:** Acts similar to activation in neural nets → rarely used now.

Equation in the Image

👉 $y = w^T x + b$

This is the **general equation of a hyperplane** used in **Support Vector Machine**.

👉 $y = mx + c$

This is the **equation of a straight line** in 2D (from basic coordinate geometry).

So actually,

> $y = w^T x + b$ is the **multidimensional version** of $y = mx + c$.

Meaning of Each Term

Symbol	Meaning	Analogy in simple 2D
w	Weight vector (defines the orientation of the hyperplane)	Like the slope (m)
x	Input feature vector	Like the x value
b	Bias term (decides the shift of hyperplane from origin)	Like the y-intercept (c)
y	Output or decision value	Whether point lies on which side of the boundary
C	The intercept- value of y when x is at 0.	Adjusts balance between smooth boundary and correct classification (tolerance for errors)

How it works in SVM

In SVM, this equation helps classify data:

Condition	Meaning	Region
$w^T x + b = 0$	Points **on the hyperplane**	Decision boundary

$w^T x + b \geq +1$	Points **on or above** upper boundary	Positive class
$w^T x + b < -1$	Points **on or below** lower boundary	Negative class

👉 So, the distance between these two boundaries (+1 and −1) is the **margin**, which SVM maximizes.

🧠 Equations of the two supporting hyperplanes

1️⃣ $w^T x_1 + b = +1$
2️⃣ $w^T x_2 + b = -1$

These represent the two **supporting hyperplanes** that define the **margin** on either side of the decision boundary.

✏️ Subtracting Equation (2) from Equation (1):

$$(w^T x_1 + b) - (w^T x_2 + b) = (+1) - (-1)$$

Simplify it:

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

📏 Distance between the two hyperplanes (Margin)

The distance (margin) between these two parallel hyperplanes is given by:

$$\text{Margin} = \frac{2}{||w||}$$

Meaning:

- w → Weight vector (direction/slope of the hyperplane)
- ||w|| → Magnitude (length) of the weight vector
- SVM tries to **maximize this margin** to achieve the best separation between classes.