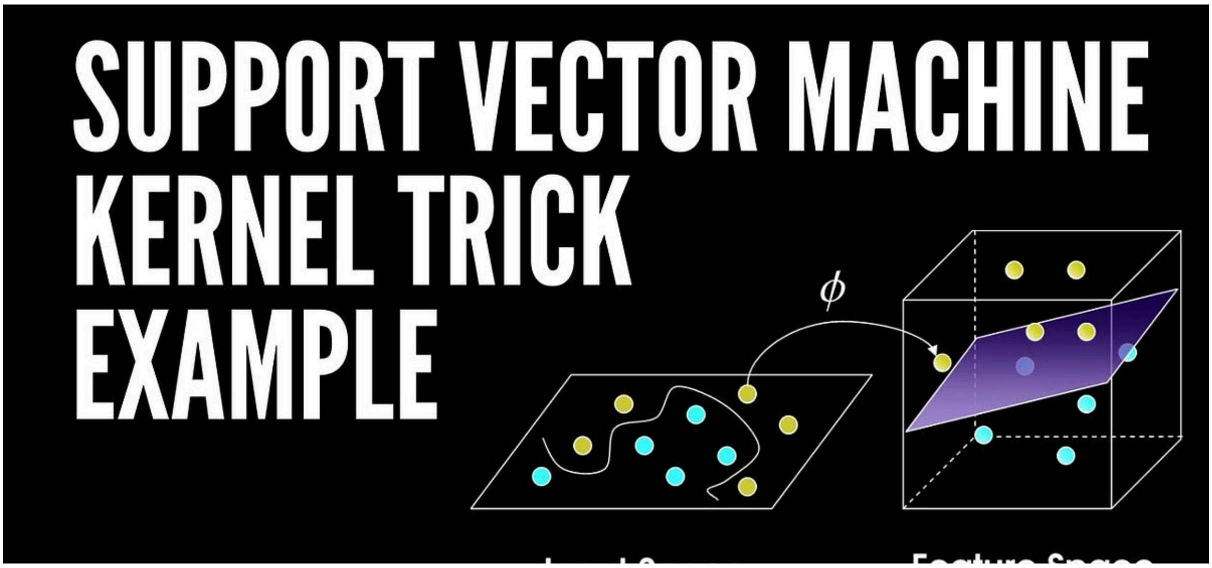
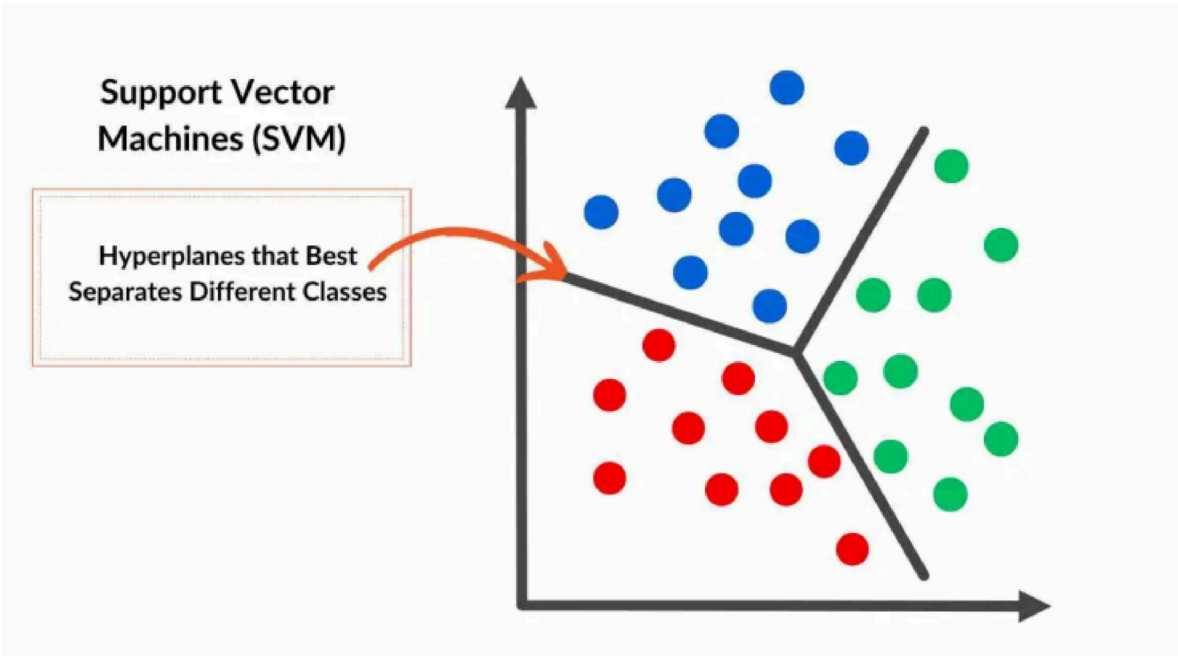


Modules 3: Support Vector Machine (SVM) & Kernel Trick

1. SVM Theory (Support Vector Machine)



🔷 Concept

SVM is a supervised learning algorithm for classification and regression.

Main goal: find a hyperplane that separates classes with the largest possible margin.

🔷 Key concepts

Term	<Explanation
Support Vectors	Points closest to the hyperplane that determine its position.
Hyperplane	A separating plane. In 2D it's a line, in 3D a plane, in nD a hyperplane.
Margin	Distance from the hyperplane to the nearest points of each class. SVM maximizes this.
Linear / Non-linear	Linear data can be split by a line. Non-linear data needs a kernel mapping.

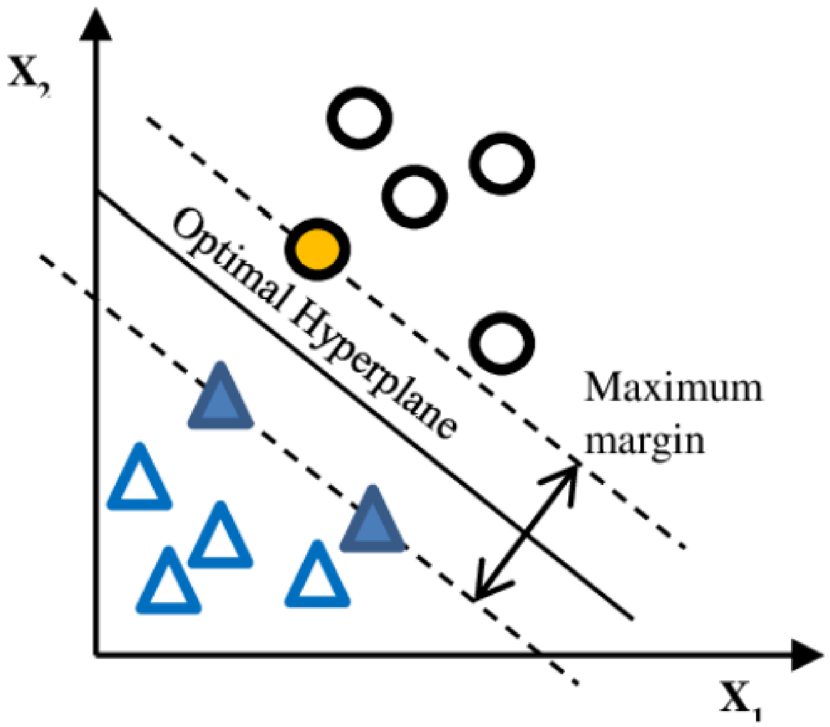
🔷 Basic idea

Two classes: +1 and −1.

Find a line (2D) or hyperplane (nD) such that:

- All +1 points are on one side
- All −1 points are on the other side
- The distance to the nearest points of both classes is as large as possible

This is the Maximum Margin Hyperplane (MMH).



🔷 Math formulation

Linear SVM model:

$$f(x) = w^\top x + b$$

with w the weight vector and b the bias.

Correct classification condition:

$$y_i\left(w^\top x_i + b\right) \geq 1 \quad \forall i$$

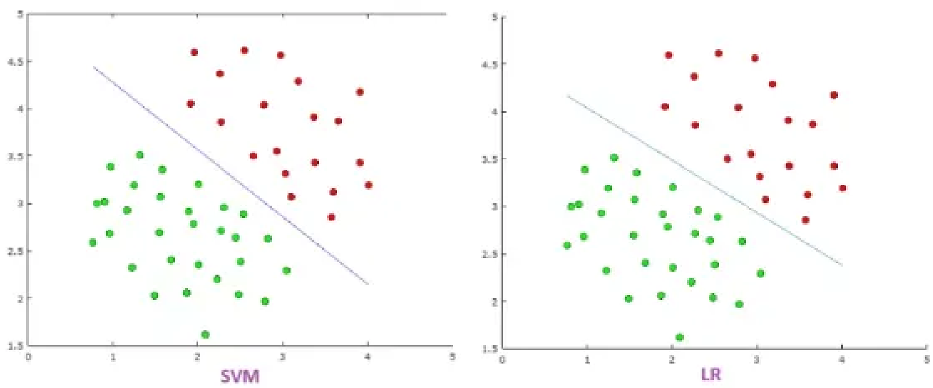
Optimization (hard margin):

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

which maximizes the margin by minimizing $\|w\|$.

2. SVM vs Logistic Regression

Criterion	Logistic Regression	SVM
Nature	Probabilistic	Geometric
Loss	Log loss	Hinge loss
Output	Probability (0–1)	Class label (+1, −1)
Outlier sensitivity	High	Lower
Pros	Interpretability	High performance on complex data



3. SVM cost function (soft margin)

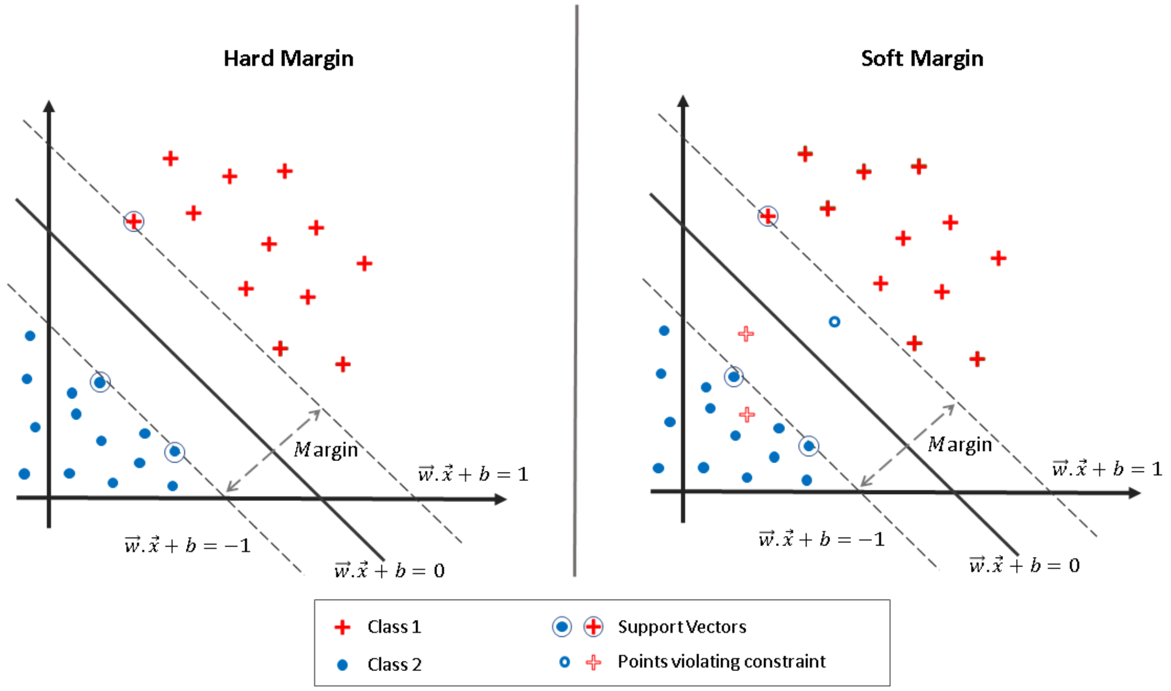
$$\text{Cost} = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \max\left(0,\, 1 - y_i\left(w^\top x_i + b\right)\right)$$

Where:

- w: weight vector
- b: bias
- C: regularization parameter
- $y_i \in \{-1, +1\}$: label of sample i

- x_i: feature vector of sample if sample i

Goal: minimize the cost to find the optimal separating hyperplane.



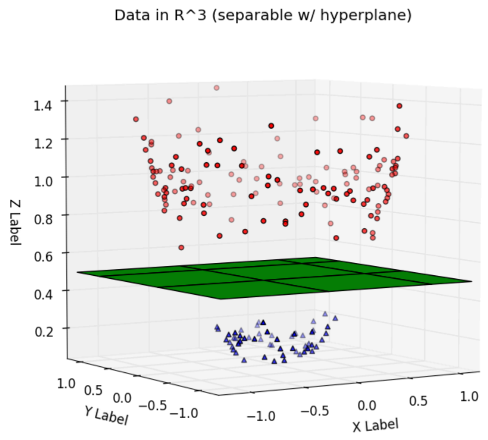
4. Regularization in SVM

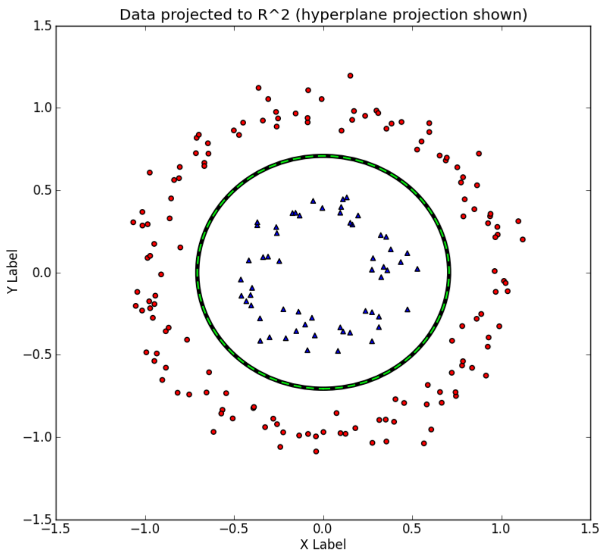
- Large C → weak regularization → tight fit on training data → risk of overfitting
- Small C → strong regularization → allow some errors → better generalization

5. Non-linear data

Most real datasets are not linearly separable. A linear SVM will fail in the original space.

6. Kernel Trick





⭐ Tóm tắt nhanh: Kernel trick ánh xạ dữ liệu sang không gian đặc trưng cao hơn để tách tuyến tính, làm việc trực tiếp qua hàm kernel $K(x_i, x)$ mà không cần tính $ `\phi(\cdot)` $.

🔷 Idea

Map data to a higher-dimensional feature space where it becomes linearly separable.

🔷 What is a kernel?

A kernel measures similarity:

$$K(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$$

where $\phi(\cdot)$ maps inputs to a feature space and $K(\cdot, \cdot)$ is the inner product there.

🔷 Kernel trick

We do not need explicit $phi(cdot)$. It suffices to define K so that

$$K(x_i, x) = \phi(x_i)^\top \phi(x)$$

and compute directly in input space.

🔷 Decision function

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i \, y_i \, K(x_i, x) + b\right)$$

Trong đó:

- x_i: vector huấn luyện
- $y_i \in \{-1, +1\}$: nhãn lớp
- α_i: hệ số Lagrange (trọng số)
- $K(x_i, x)$: giá trị kernel giữa hai điểm
- b: hệ số điều chỉnh

---

7. Common kernels

Linear

$$K(x, x') = x^\top x' + c$$

Use when the boundary is close to linear.

Polynomial

$$K(x, x') = (\gamma\, x^\top x' + r)^d$$

Parameters:

- d: polynomial degree
- γ: scale factor
- r: bias term

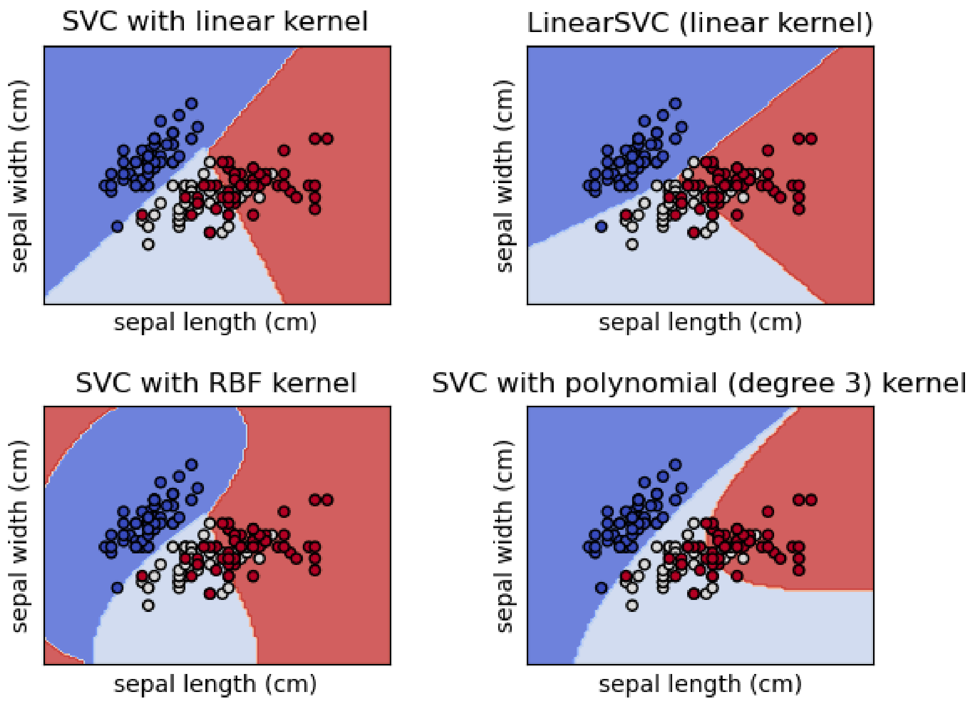
RBF (Gaussian)

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

Gần nhau → $K(x, x') \approx 1$, xa nhau → $K(x, x') \approx 0$.

Sigmoid

$$K(x, x') = \tanh(\gamma\, x^\top x' + r)$$



8. Kernel comparison

Kernel	Formula	Non-linearity	Pros	Cons	When to use
Linear	$x^\top x'$	Low	Fast, simple	Cannot model non-linear	Clearly linear data
Polynomial	$(\gamma x^\top x' + r)^d$	Medium	Captures higher-order interactions	Can overfit at high d	Polynomial relationships
RBF	$e^{-\gamma \lVert x - x' \rVert^2}$	High	Flexible, strong	Slow on very large data; needs scaling	Complex curved boundaries
Sigmoid	$\tanh(\gamma x^\top x' + r)$	Medium	NN-like	Less stable	Small datasets

9. Important hyperparameters

🎯 Quan trọng: Điều chỉnh C và gamma ảnh hưởng trực tiếp biên và độ cong ranh giới. Bắt đầu với C=1, gamma = 1/num_features cho RBF, rồi tinh chỉnh qua GridSearch.

Parameter	Meaning	Effect
C	Misclassification penalty	Large C → small margin, tight fit. Small C → stronger regularization.

Parameter	Meaning	Effect
γ	Locality of influence (RBF, Poly, Sigmoid)	Large γ → curvy boundary. Small γ → flatter boundary.
d	Polynomial degree	Higher d → more complex model
$\mathrm{coef0}$	Bias term r in Poly/Sigmoid	Affects boundary curvature

10. Code examples

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

# Linear toy data
X, y = datasets.make_classification(
    n_samples=200, n_features=2,
    n_redundant=0, n_informative=2,
    random_state=42
)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)

clf = SVC(kernel='linear', C=1.0)
```

```
# Non-linear moons
from sklearn.datasets import make_moons
X, y = make_moons(n_samples=300, noise=0.2, random_state=42)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)
svm_rbf = SVC(kernel='rbf', gamma=0.5, C=1.0)
svm_
```

11. SVM ML workflow

1. Data prep → collect
2. Preprocess → handle missing, scale
3. Choose model → Linear or Kernel SVM
4. Train → .fit()

5. Evaluate → accuracy, confusion matrix
6. Tune → GridSearchCV for C and γ
7. Deploy → save model, serve predictions

---

12. Pros and cons

Pros

- Works well for non-linear data
- Flexible via kernels
- Effective in high dimensions

Cons

- Slow on very large datasets
- Requires scaling
- Hyperparameter tuning can be tricky
- No native probabilities

---

13. Applications

- Text classification
- Handwriting recognition
- Face recognition
- Fraud detection
- Anomaly detection
- Bioinformatics

---

14. Final summary

✅ Tóm tắt then chốt: SVM tìm siêu phẳng tối ưu. Kernel trick giúp xử lý dữ liệu phi tuyến. RBF là lựa chọn mặc định an toàn, tinh chỉnh C và gamma.

Item	Content
Goal	Find an optimal separating hyperplane
Kernel trick	Map to a higher-dimensional feature space
Decision	$f(x) = \text{sign}\left(\sum_i \alpha_i y_i K(x_i, x) + b\right)$

Item	Content
Main kernels	Linear, Polynomial, RBF, Sigmoid
Key params	C, $gamma$, d, $\mathrm{coef0}$
Most common	RBF (Gaussian)