Applied AI & Machine Learning
CS-333

Dr. Abbas Hussain

PNEC, NUST

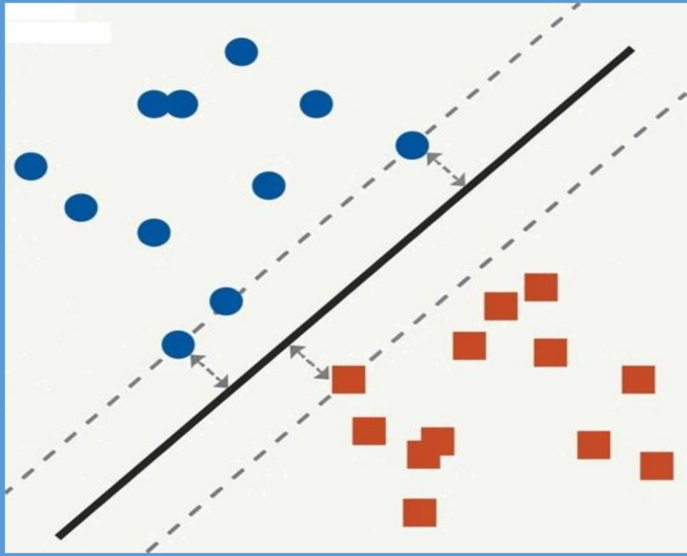
Lecture 4

Spring 2026

Machine Learning (ML) Models

From Data to Prediction

Data → Model → Loss → Optimization → Prediction



Beyond the Line

Support Vector Machines (SVMs)

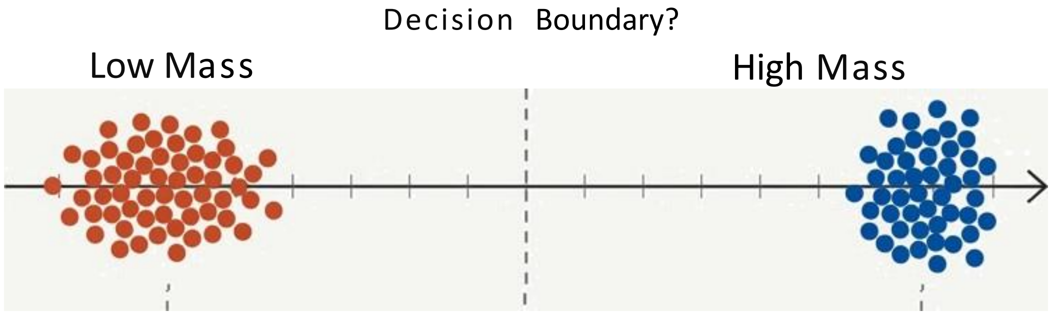
Learning Outcomes

By the end of this lecture, students will be able to:

- Define Support Vector Machine (SVM).
- Explain hyperplane, margin, and support vectors.
- Interpret the decision boundary equation $w^T x + b = 0$.
- Explain why SVM maximizes the margin.
- Compute the margin using $2/\| w \|$.
- Describe the role of w and b.
- Compare One-vs-Rest and One-vs-One strategies.
- Determine the number of hyperplanes in multi-class SVM.
- Justify why SVM provides better generalization.

The Goal is Separation

We start with labeled data (Supervised Learning) and a simple objective: Classification.



The Analogy: Imagine a 1D scale representing the mass of mice,

- Low Mass: Not Obese (Red)
- High Mass: Obese (Blue)

The Task: Where do we draw the Threshold? Any new observation to the left is Class A; to the right is Class B.

Support Vector Machines (SVMs)

- Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It tries to find the best boundary known as **hyperplane** that separates different classes in the data.
- The main goal of SVM is to maximize the **margin** between the two classes. The larger the margin the better the model performs on new and unseen data.

Hyperplane

- A hyperplane is a decision boundary that separates data points into different classes in a high-dimensional space.
- **N-dimensional space**, a hyperplane has **(N-1)-dimensions**.

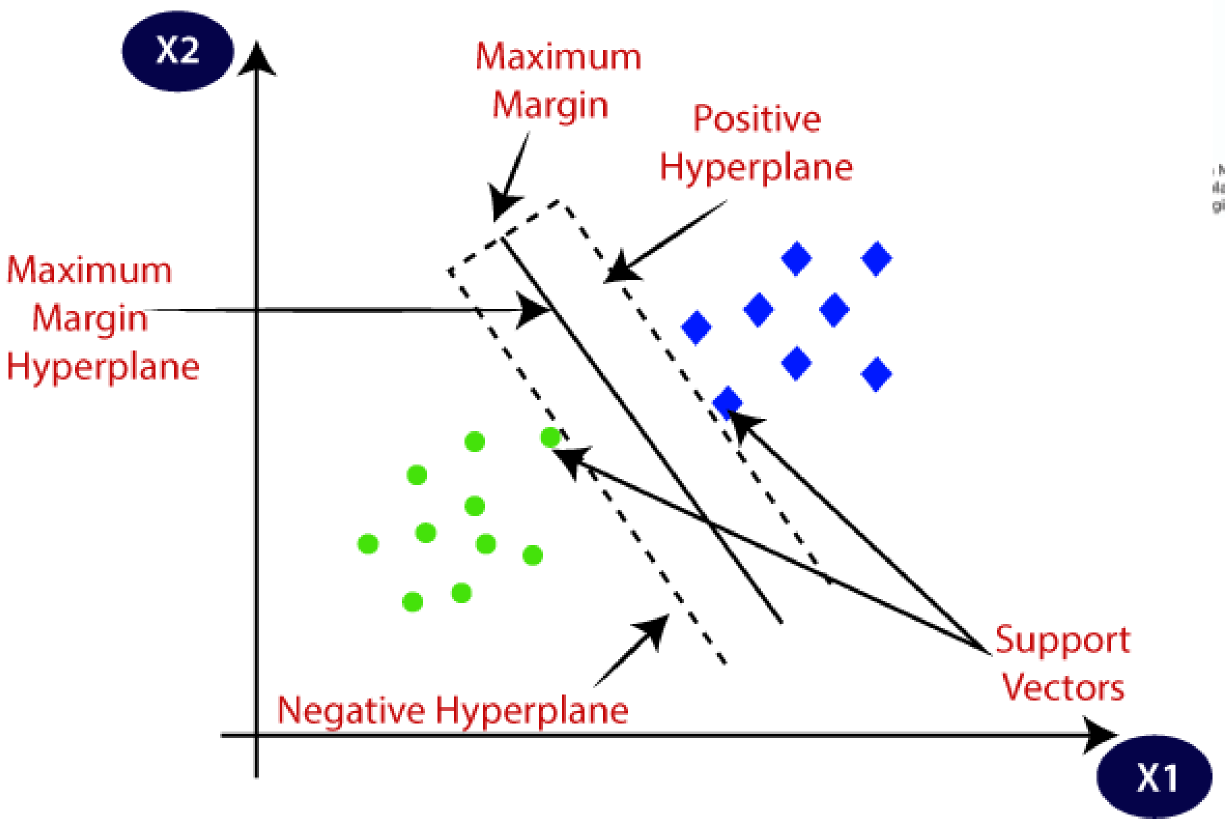
Margin

- A **margin** is the distance between the decision boundary (hyperplane) and the closest data points from each class.
- The goal of SVMs is to maximize this margin while minimizing classification errors.

Support Vector Machines (SVMs)

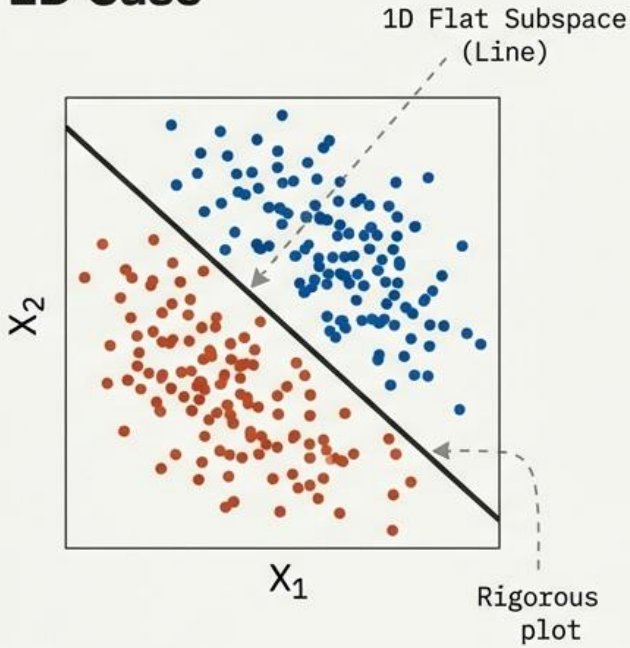
Support Vectors

- They are the data points that lie closest to the decision boundary (hyperplane) in a Support Vector Machine (SVM).



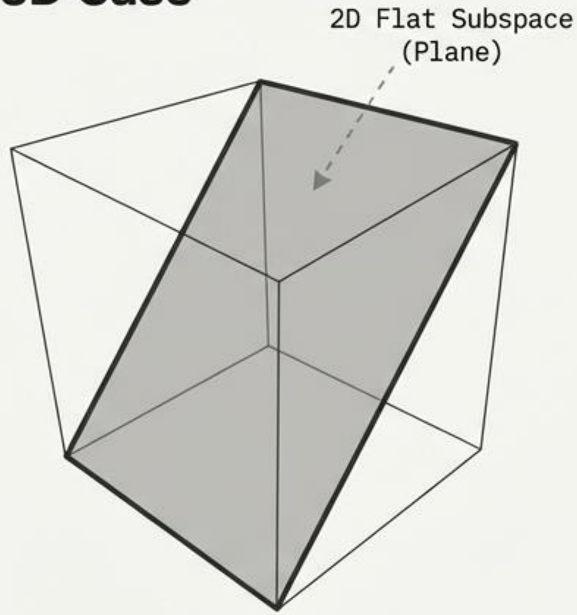
Formalizing the Boundary: The Hyperplane

2D Case



B0 + B1X1 + B2X2 = 0

3D Case



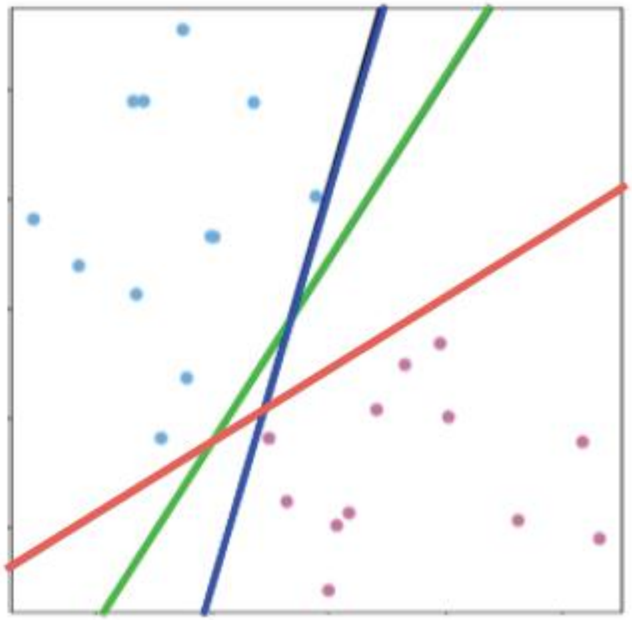
p-Dimensions

In p-dimensions, a hyperplane is a flat affine subspace of dimension p-1.

B0 + B1X1 + … + BpXp = 0

The Paradox of Choice

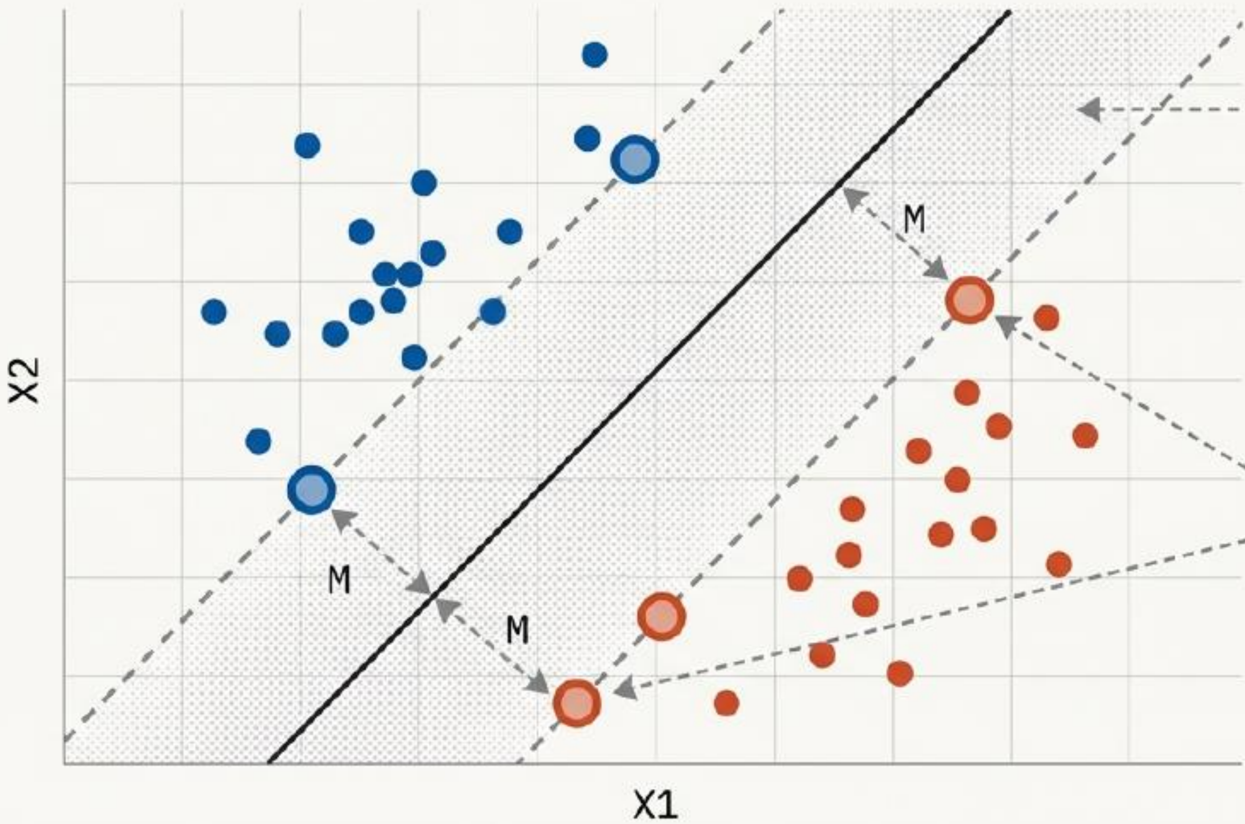
Infinite solutions exist. Which one is safe?



Which decision boundary?

High Risk: A boundary too close to data is sensitive to noise. New data might cross the line.

The Maximal Margin Solution



The Margin

These specific points 'support' the boundary. Moving other points doesn't change the line.

Support Vectors

Geometry of SVM

Decision Boundary Equation:

$w^T x + b = 0$

- w is a vector perpendicular (normal) to the boundary.
- b shifts the boundary.

SVM defines two parallel hyperplanes:

$w^T x + b = +1$

$w^T x + b = -1$

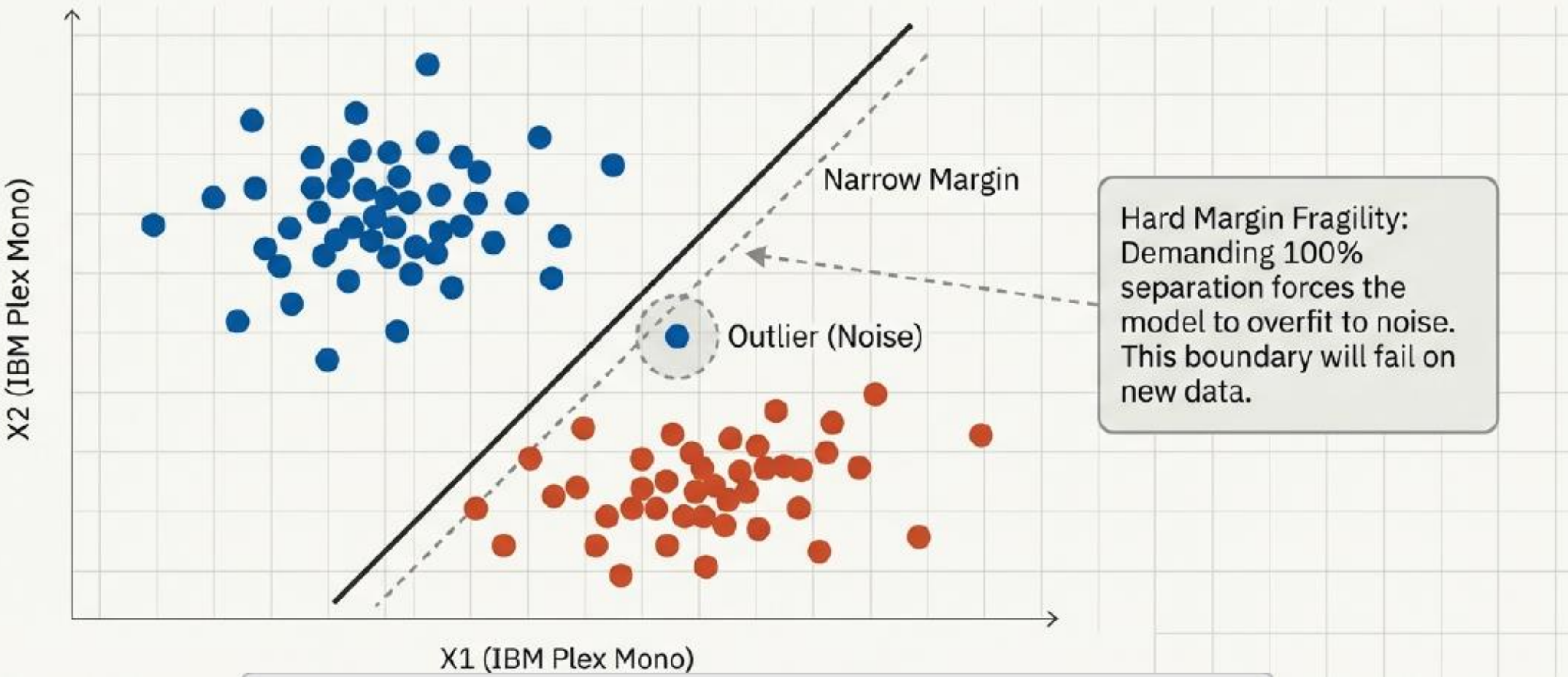
Distance between them (margin):

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

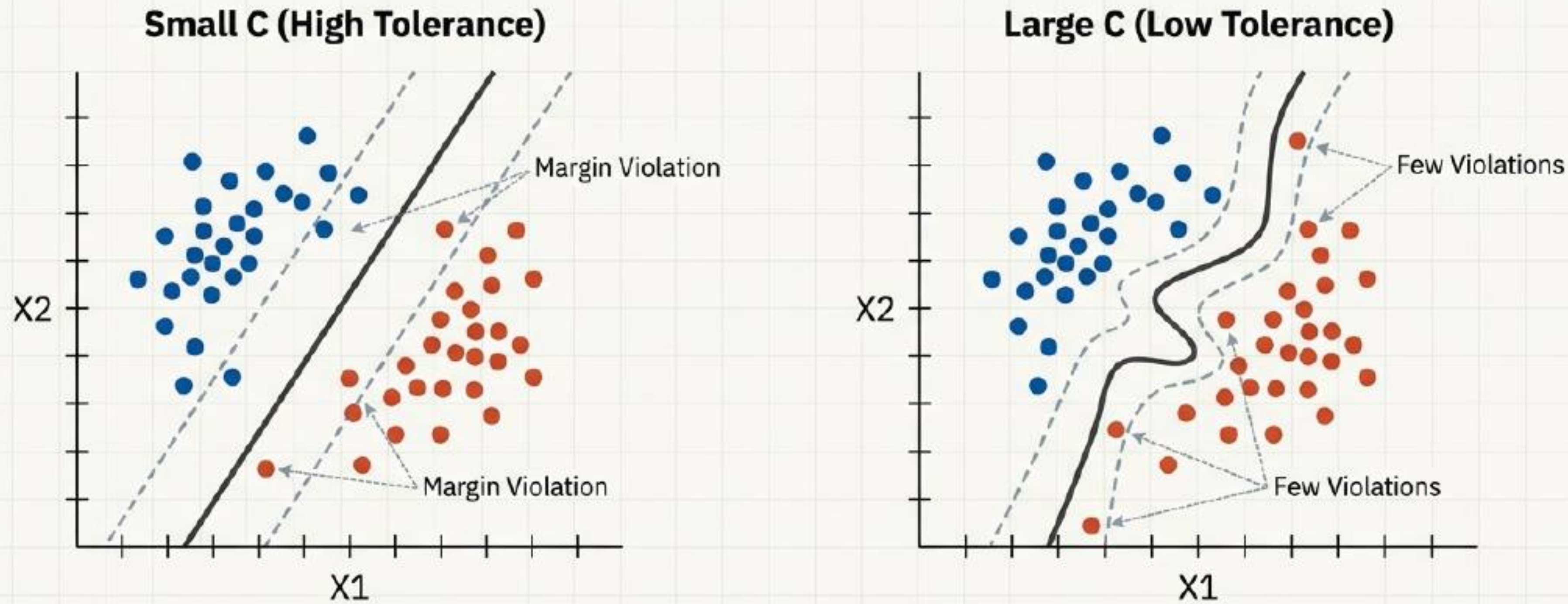
Smaller ||w|| → Larger margin.

That is why SVM minimizes ||w||.

When Perfection Fails: The Outlier Problem



Tuning the 'Budget' (The C Parameter)



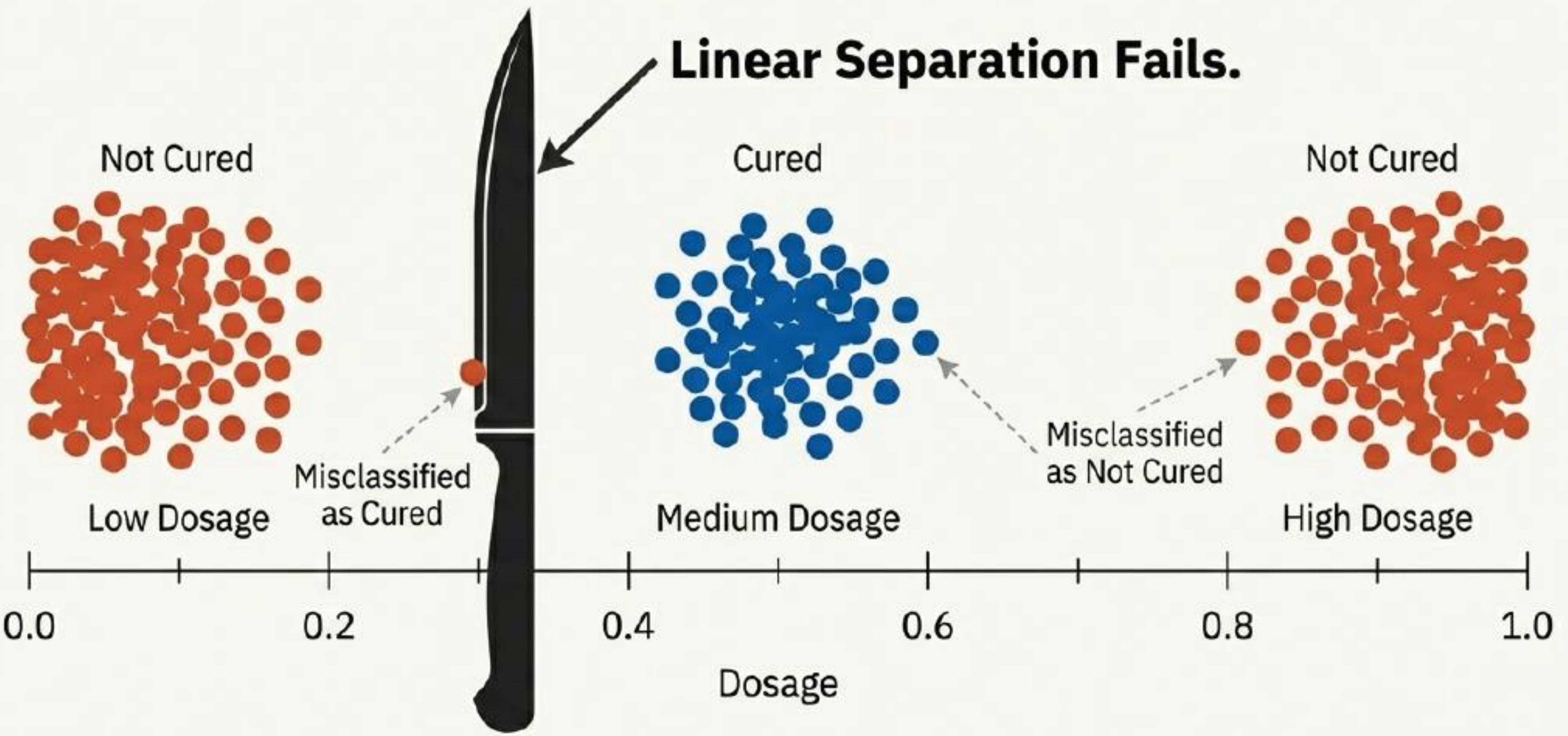
High Bias, Low Variance (More Robust)

Low Bias, High Variance (Risk of Overfitting)

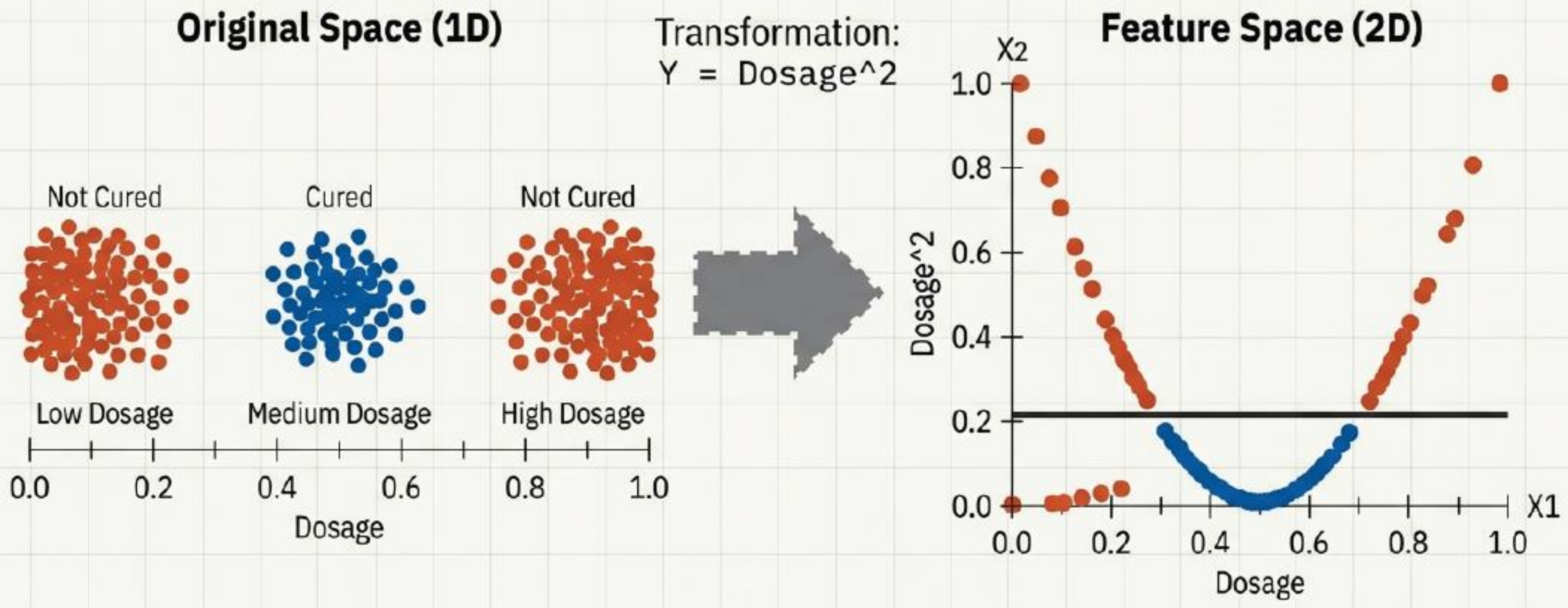
C acts as the penalty for margin violations.
It is a 'Budget for Misconduct'.

The Non-Linear Reality

When a straight line cannot solve the problem.



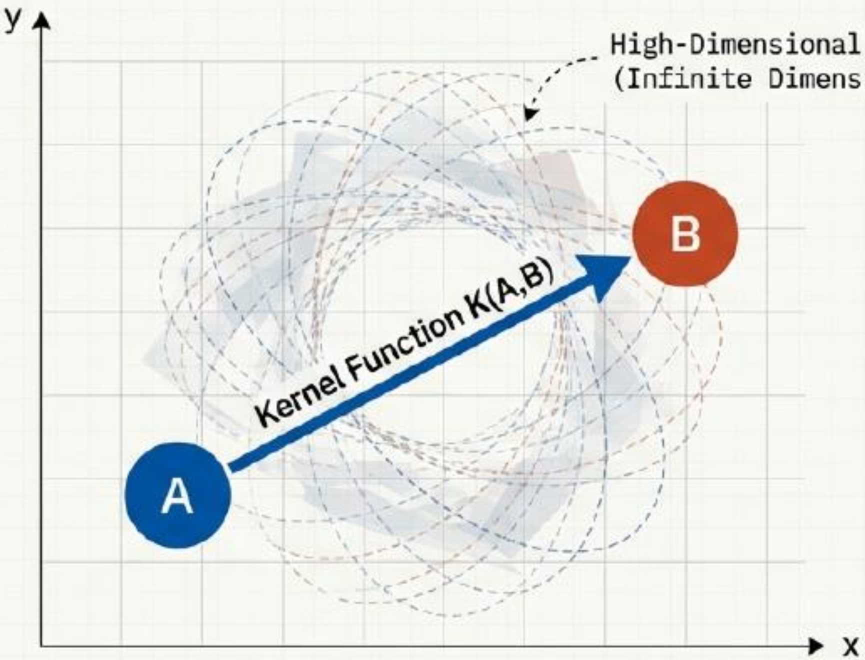
Expanding the Feature Space



Linear in Feature Space = Non-Linear in Original Space.

The Kernel Trick

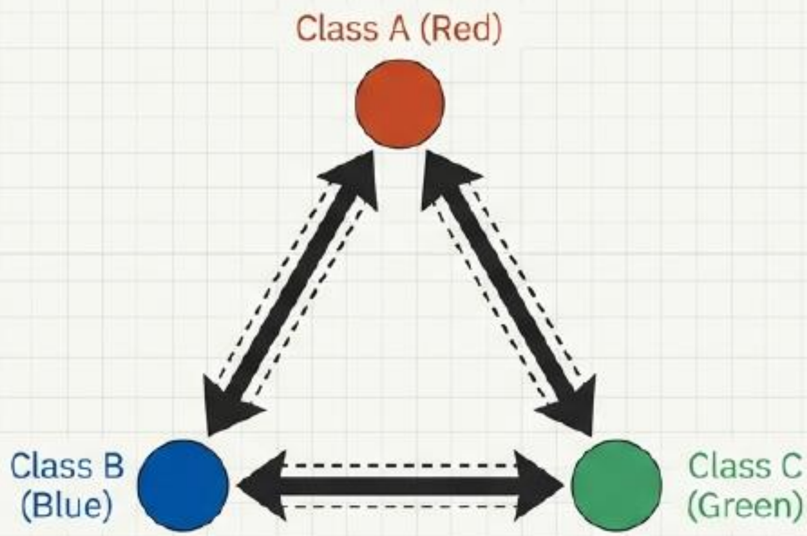
We don't need to actually calculate the coordinates in infinite dimensions. We just need the dot product.



Polynomial Kernel:	Radial Basis Function (RBF):
- $(x * y + c)^d$	- exp(-gamma * |x - y|^2)
- Creates curved boundaries.	- Infinite dimensions. Acts like a weighted nearest neighbor.

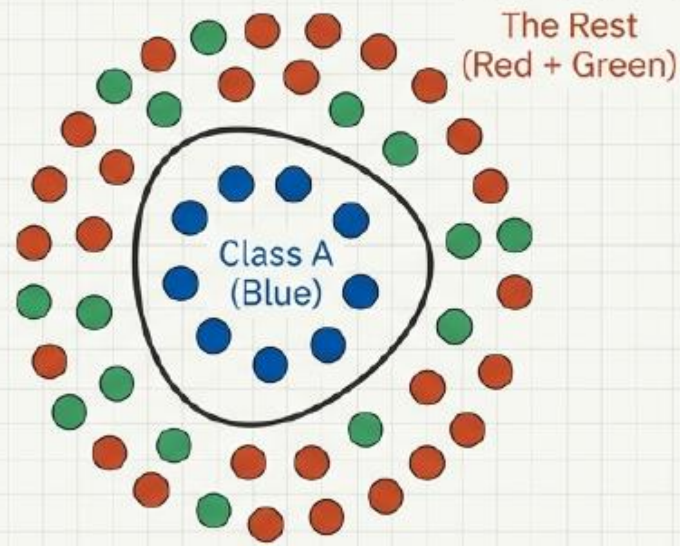
Beyond Binary: Multiclass Classification

One – versus- One (OVO)



Train k(k-1)/2 classifiers.
Every class fights every other class.
Majority vote wins.

One – versus- rest (OVR)



Train k classifiers.
Class A vs. The World.
Highest confidence wins.

Pros & Cons Evaluation

Advantages (+)

- **High Dimensionality:** Effective even when dimensions > samples.
    - Ideal for genomics and text analysis.
- **Memory Efficient:** Uses only support vectors (subset of training points).
    - Reduces storage requirements.
- **Versatile:** Kernels allow adaptation to complex data shapes.
    - Non-linear separation capability.

Disadvantages (-)

- **Scale:** Computationally expensive for large datasets (O(n^2)).
    - Training time grows quadratically.
- **Noise Sensitivity:** Performance drops with overlapping classes (if C is not tuned).
    - Requires careful hyperparameter tuning.
- **No Probability:** Doesn't provide direct probability estimates (unlike Logistic Regression).
    - Output is a decision boundary distance.

Implementation Cheat Sheet

```python
from sklearn.svm import SVC

# 1. Linear Support Vector Classifier
model_linear = SVC(kernel='linear', C=1.0)

# 2. Non-Linear (RBF) with Tuning
model_rbf = SVC(kernel='rbf', C=10.0, gamma='scale')

model_rbf.fit(X_train, y_train)
```

- ☑ **[Scale Data]** (Crucial for SVM distance calculations)
- ☑ **[Select Kernel]** (Linear vs. RBF)
- ☑ **[Tune Hyperparameters]** (GridSearch for C and Gamma)