LAB 2

Support Vector Machines (SVM)

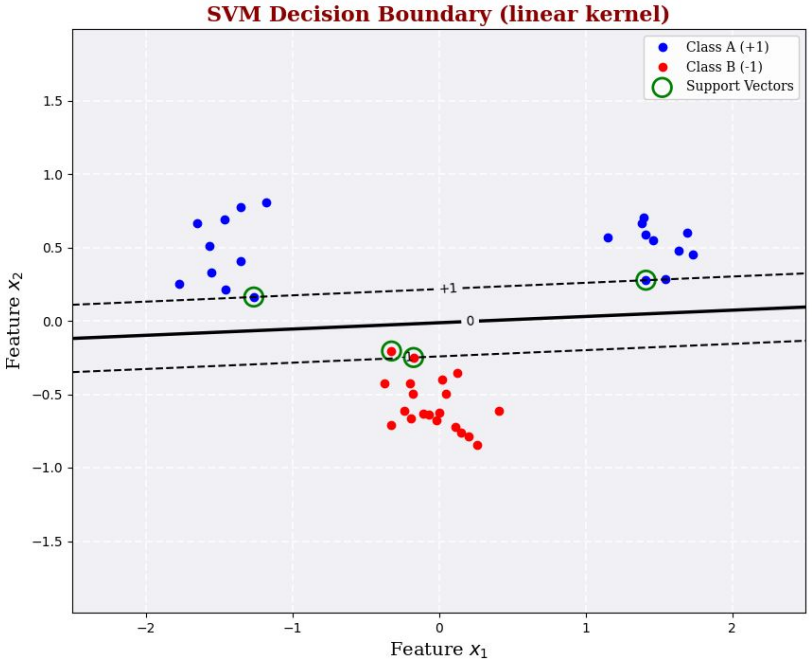
Riccardo Fragale
Matteo Camellini

Linear kernel

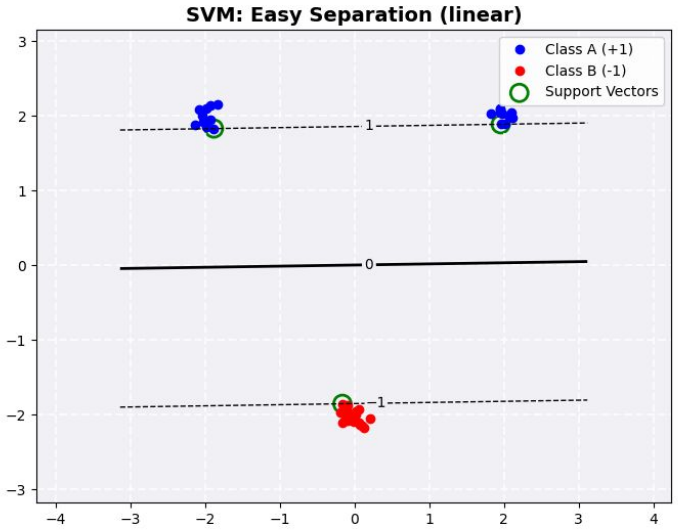
$$\mathcal{K}(\vec{x}, \vec{y}) = \vec{x}^T \cdot \vec{y}$$

This kernel simply returns the scalar product between the two points.
This results in a linear separation.

Linear separation base case



Question 1: Data Geometry and Optimizer Stability

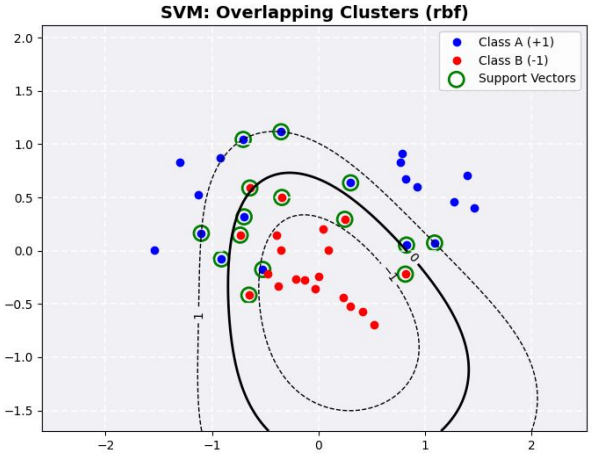


We examined 3 distinct scenarios to evaluate how the spatial distribution, overlap, and noise levels of data clusters affect the effectiveness of the Support Vector Machine (SVM).

Scenario 1: Easy Linear Separation

In this configuration, the clusters are small, tightly packed, and widely separated along the feature space. Because there is a clear, unobstructed path for a straight decision boundary, the optimizer easily converges on a linear kernel solution with a high C value. The resulting margin is clean, and only a minimum number of data points—those on the very edge of the clusters—serve as support vectors to define the boundary. In this "ideal" case, the model has very low bias because the simple linear assumption perfectly matches the underlying data distribution.

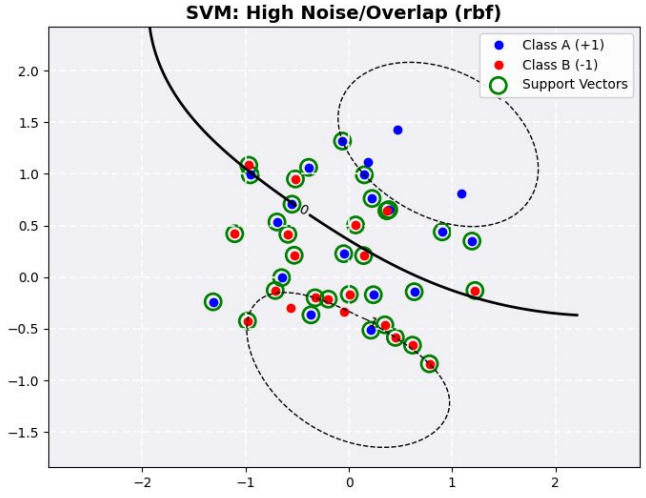
Question 1: Data Geometry and Optimizer Stability



Scenario 2: Non-Linearity and Managed Overlap

As clusters are moved closer or resized to overlap, a linear boundary becomes insufficient. In this middle scenario, an RBF kernel is employed to create a flexible, curved boundary that "scoops" around the encroaching points of Class B. By introducing a moderate amount of slack (C), the SVM handles points that fall within the margin or slightly on the wrong side without forcing the boundary into extreme, jagged shapes. This balance prevents the optimizer from failing, allowing it to find a solution that captures the non-linear trend while maintaining a relatively smooth boundary that generalizes well to the overlapping regions.

Question 1: Data Geometry and Optimizer Stability



Scenario 3: High Noise and Extreme Overlap

When clusters are positioned to be almost entirely on top of each other with high internal variance (noise), the optimizer faces significant difficulty. As seen in the final image, the boundary must become extremely complex to attempt separation, leading to a high density of support vectors—nearly every point in the overlapping region is used to define the boundary. If the slack parameter C is set too high in this noisy environment, the minimize function may fail to converge entirely because it cannot satisfy the strict requirement of separating such chaotic data. In this case, the only way to achieve a "decent" boundary is to opt for a very low C, effectively telling the model to ignore the noise and prioritize a simpler, more biased trend over individual point accuracy.

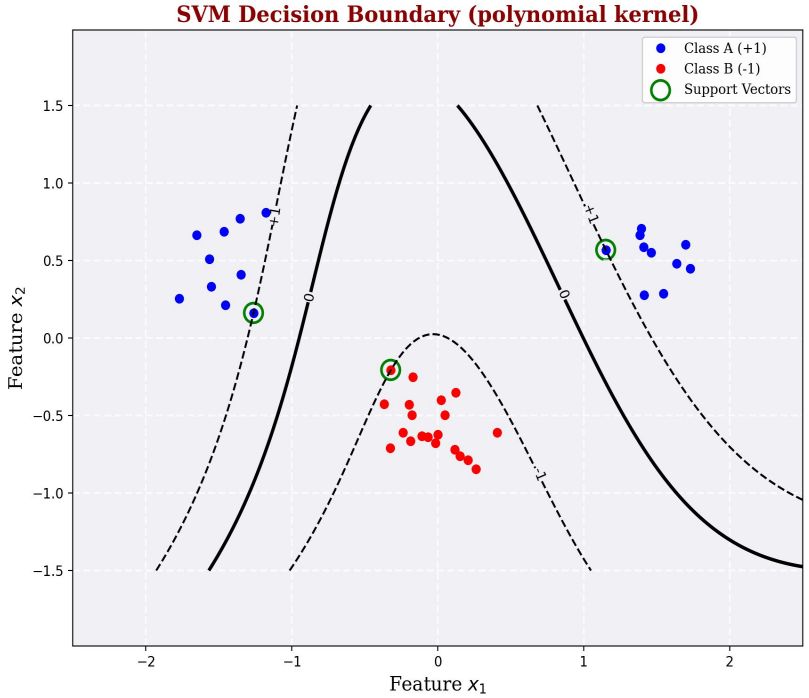
Polynomial kernel

$$K(\vec{x}, \vec{y}) = (\vec{x}^T \cdot \vec{y} + 1)^p$$

This kernel allows for curved decision boundaries. The exponent p (a positive integer) controls the degree of the polynomials.

- p = 2 will make quadratic shapes (ellipses, parabolas, hyperbolas).
- p = 3 or higher will result in more complex shapes.

Question 2: Beyond Linearity: Tackling Non-Separable Data with Poly and RBF Kernels

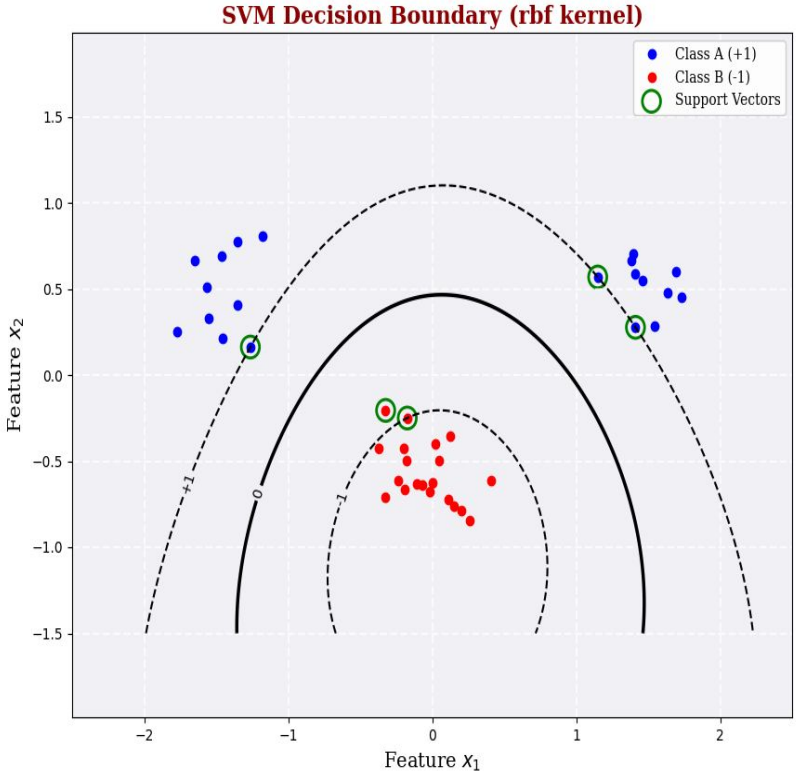


Radial Basis Function Kerr

$$\mathcal{K}(\vec{x}, \vec{y}) = e^{-\frac{||\vec{x}-\vec{y}||^2}{2\sigma^2}}$$

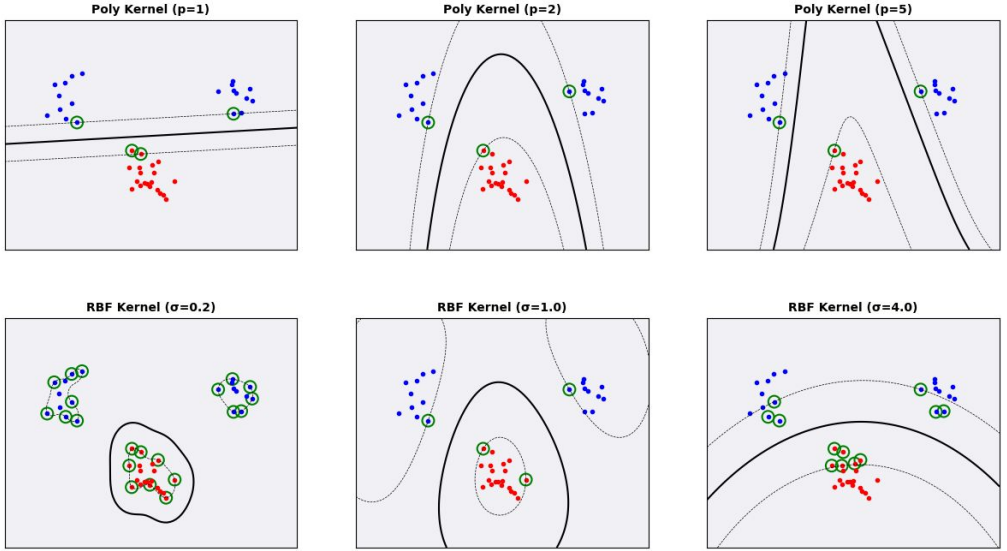
This kernel uses the explicit euclidian distance between the two datapoints and often results in very good boundaries. The parameter σ is used to
control the smoothness of the boundary.

Question 2: Beyond Linearity: Tackling Non-Separable Data with Poly and RBF Kernels



Question 3: Parameters p and σ

The parameter p controls the degree of the decision boundary. Increasing p allows for more complex, non-linear separations but increases the risk of overfitting and numerical instability. The parameter σ controls the “smoothness” of the boundary. A small σ implies high complexity; the model focuses on individual points, leading to “islands” (overfitting). Instead, a large σ involves low complexity; the model focuses on the global shape, leading to a stiffer, flatter boundary (underfitting).



Slack parameter C

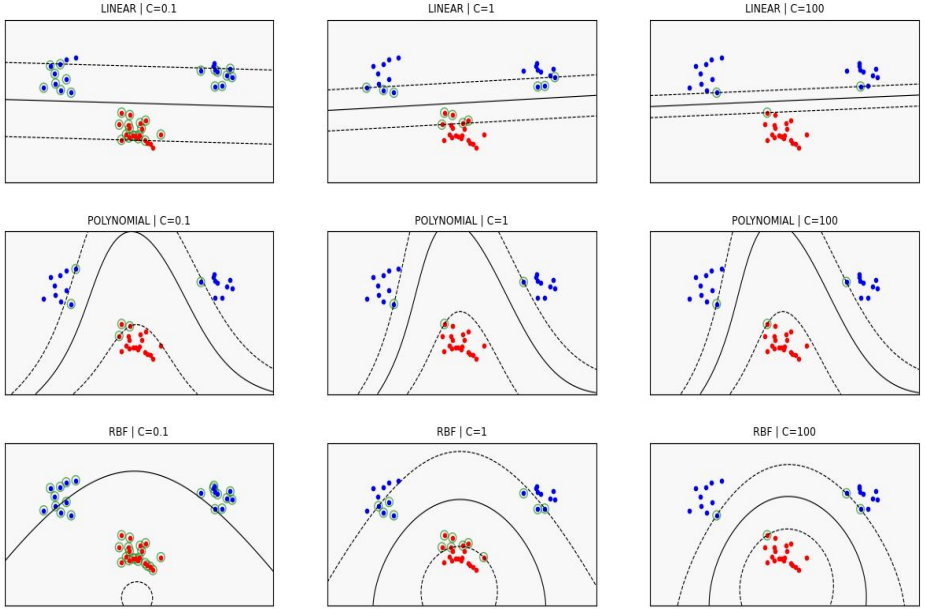
Instead of requiring that every datapoint is outside the margin we will now allow for mistakes, quantified by variables ξi (positive values; one for each datapoint). These are called slack variables.
To make sense, we must ensure that the slack variables do not become unnecessarily large. This is easily achieved by adding a penalty term to the cost function, such that large ξ values will be penalized.

$$\min_{\vec{w},b,\vec{\xi}} ||\vec{w}|| + C \sum_i \xi_i$$

The new parameter C sets the relative importance of avoiding slack versus getting a wider margin.

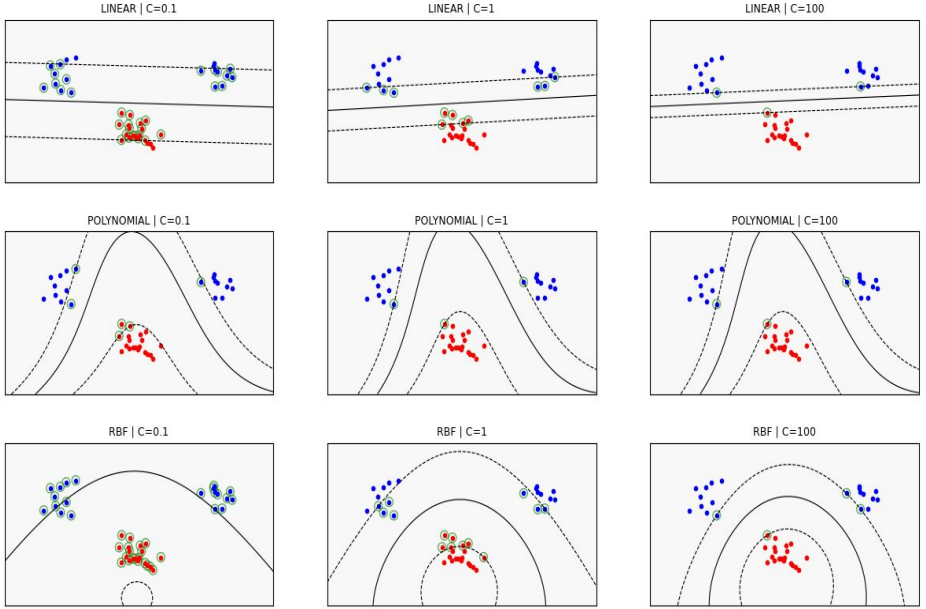
Question 4: the role of the slack parameter C

For small C values, such as C=0.1, the SVM adopts a "soft margin" approach that prioritizes model simplicity over perfect classification. The optimizer focuses on finding the widest possible margin, even if it means misclassifying some training points or allowing them to fall within the margin boundary. This results in a high bias, low variance model with a smoother, more generalized decision boundary that remains less sensitive to outliers or individual data fluctuations. Consequently, a larger number of data points become support vectors as the wide margin encompasses a greater portion of the dataset.



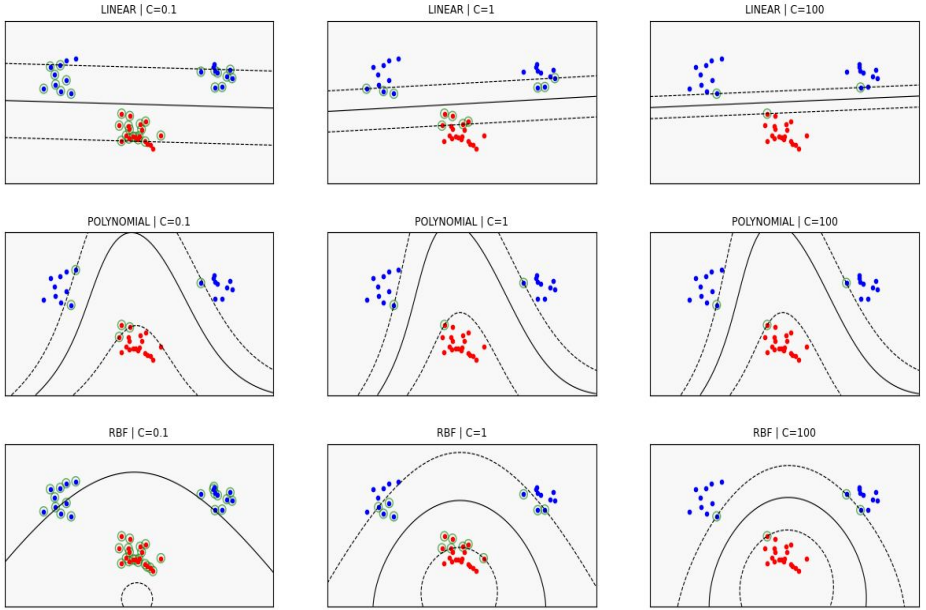
Question 4: the role of the slack parameter C

When C values are large, such as C=100, the model shifts to a "hard margin" approach that prioritizes fitting accuracy. In this state, the optimizer imposes a heavy penalty for every misclassified point, often "bending" the boundary significantly to ensure nearly every training point is correctly categorized. While this achieves low bias, it creates high variance and makes the model prone to overfitting by capturing noise rather than the underlying pattern. In this scenario, the number of support vectors typically decreases as the model focuses only on the most critical points defining a much narrower, strict margin.



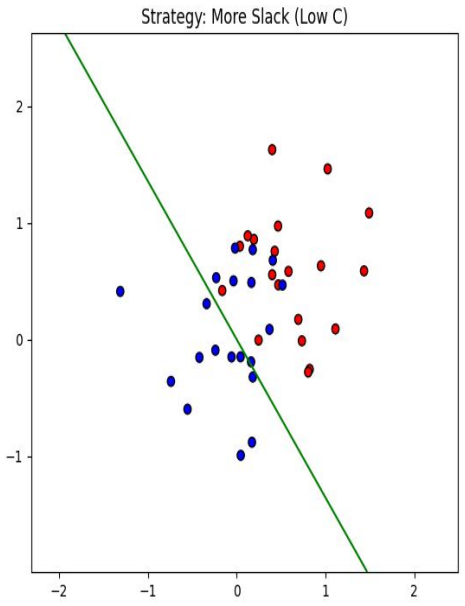
Question 4: the role of the slack parameter C

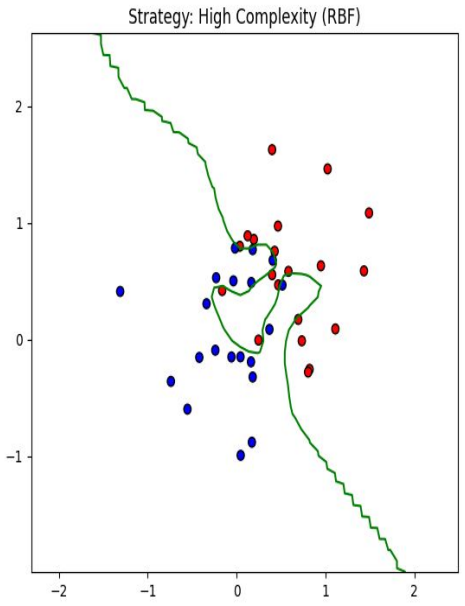
The interaction between the C parameter and the chosen kernel is critical in determining the final shape of the decision boundary. In linear kernels, changes in C primarily shift the slope or position of the straight line to accommodate more points within the margin. However, in non-linear kernels like Polynomial or RBF, C interacts directly with the kernel's inherent flexibility. High C values coupled with complex kernels produce highly intricate boundaries, such as the tight "bubbles" seen in RBF models, which are extremely tailored to the specific training set.



Question 5: more slack vs kernel

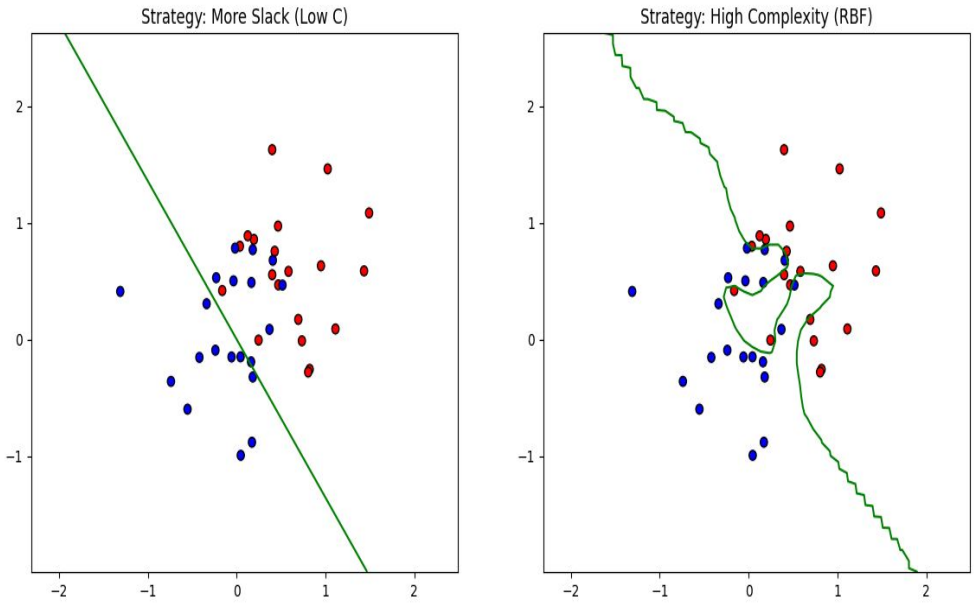
Choosing between increasing the slack parameter C or moving to a more complex kernel depends primarily on the underlying nature of the data's non-separability. When a dataset is not easily separable due to high noise, measurement errors, or naturally overlapping classes, opting for more slack, a lower C value, is often the superior strategy. This approach prioritizes model simplicity and a wider decision margin, allowing the Support Vector Machine to ignore individual outliers in favor of capturing a robust, generalized trend. As seen in comparisons of simple models with high slack, this strategy effectively prevents the decision boundary from "chasing" noise, which helps maintain high generalization performance on unseen data.





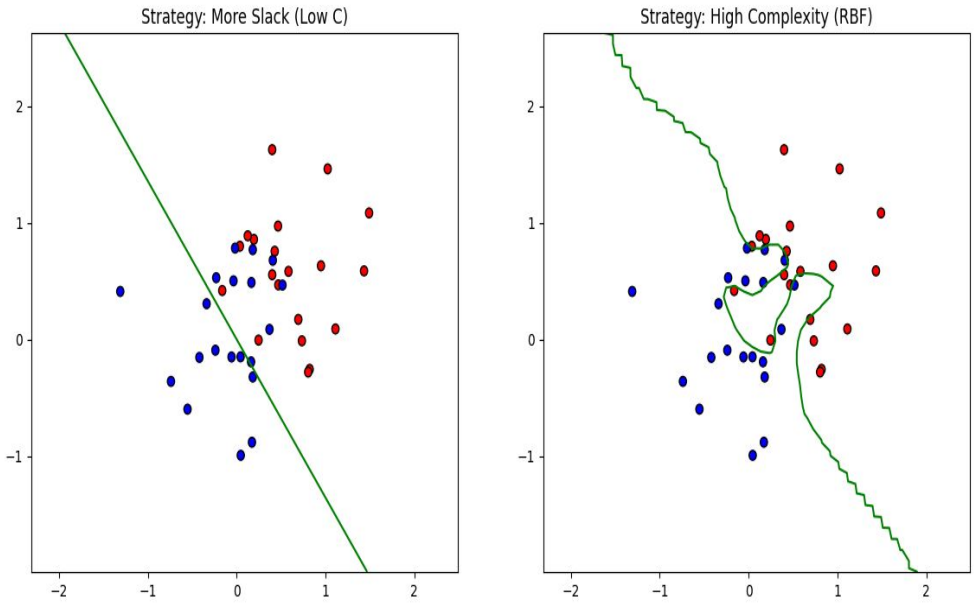
Question 5: more slack vs kernel

Conversely, a move toward a more complex model, such as transitioning from a linear to an RBF or high-degree polynomial kernel, is appropriate when the data is high-quality but follows a clear non-linear geometric structure. If the classes are distinct yet separated by intricate or curved boundaries, a complex kernel provides the necessary flexibility to model these shapes. However, this increased complexity must be managed carefully; using a highly flexible model with low slack (high C) on noisy data can lead to extreme overfitting. In such cases, the model may create tight, isolated "bubbles" or jagged boundaries around individual training points, capturing the noise of that specific dataset rather than the true underlying pattern.



Question 5: more slack vs kernel

Ultimately, the decision should be guided by the balance between bias and variance that best suits the specific application. More slack results in a model with higher bias but lower variance, which is ideal for noisy, small, or heavily overlapping datasets where a smooth transition is more sensible than a perfect fit. On the other hand, increasing model complexity lowers bias but raises variance, making it suitable for large, clean datasets where the goal is to achieve high precision across intricate boundaries. A balanced approach often involves using a non-linear kernel to capture the general shape of the data while maintaining a moderate level of slack to ensure the resulting boundary remains smooth enough to generalize well.



Thank you for listening