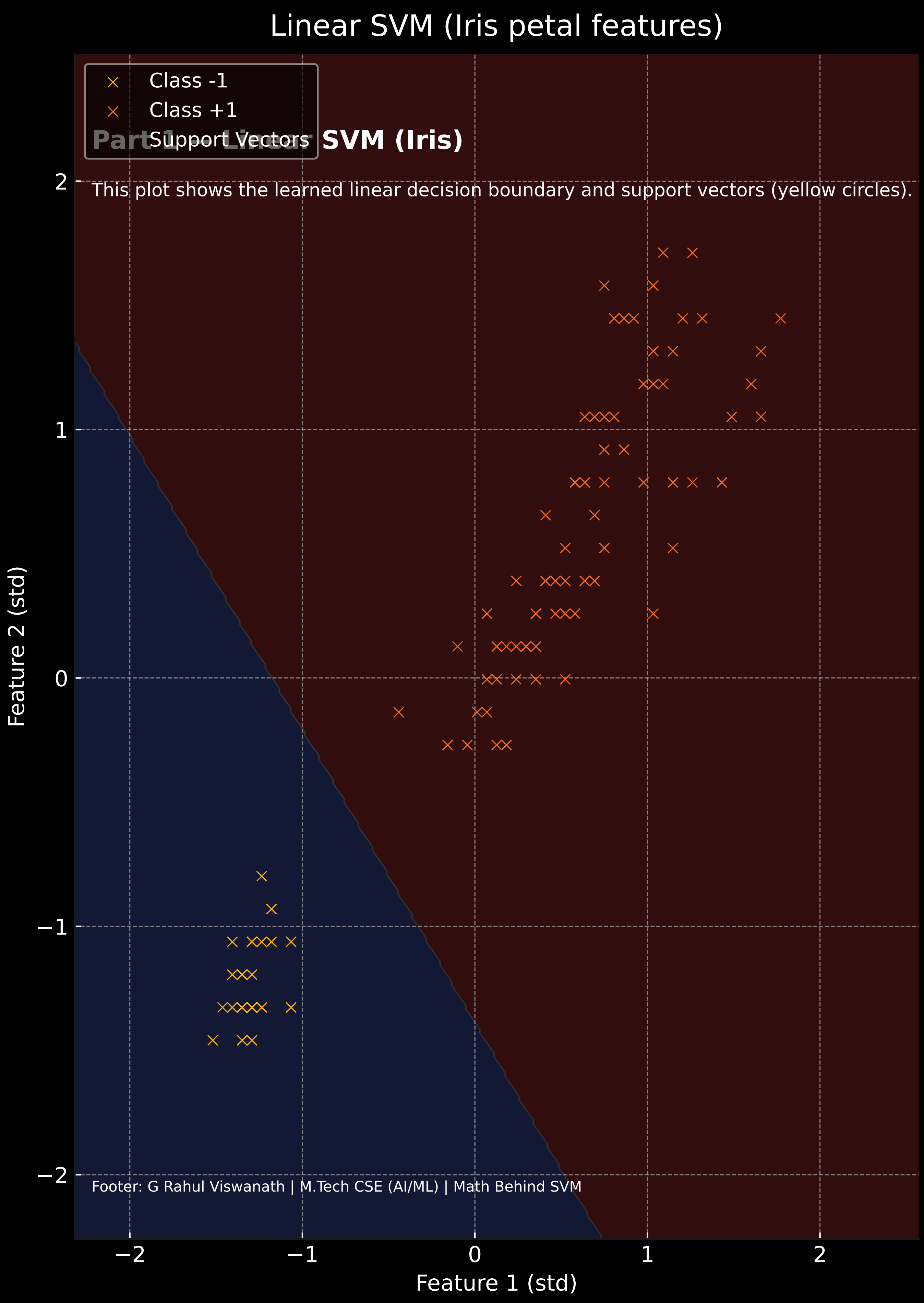
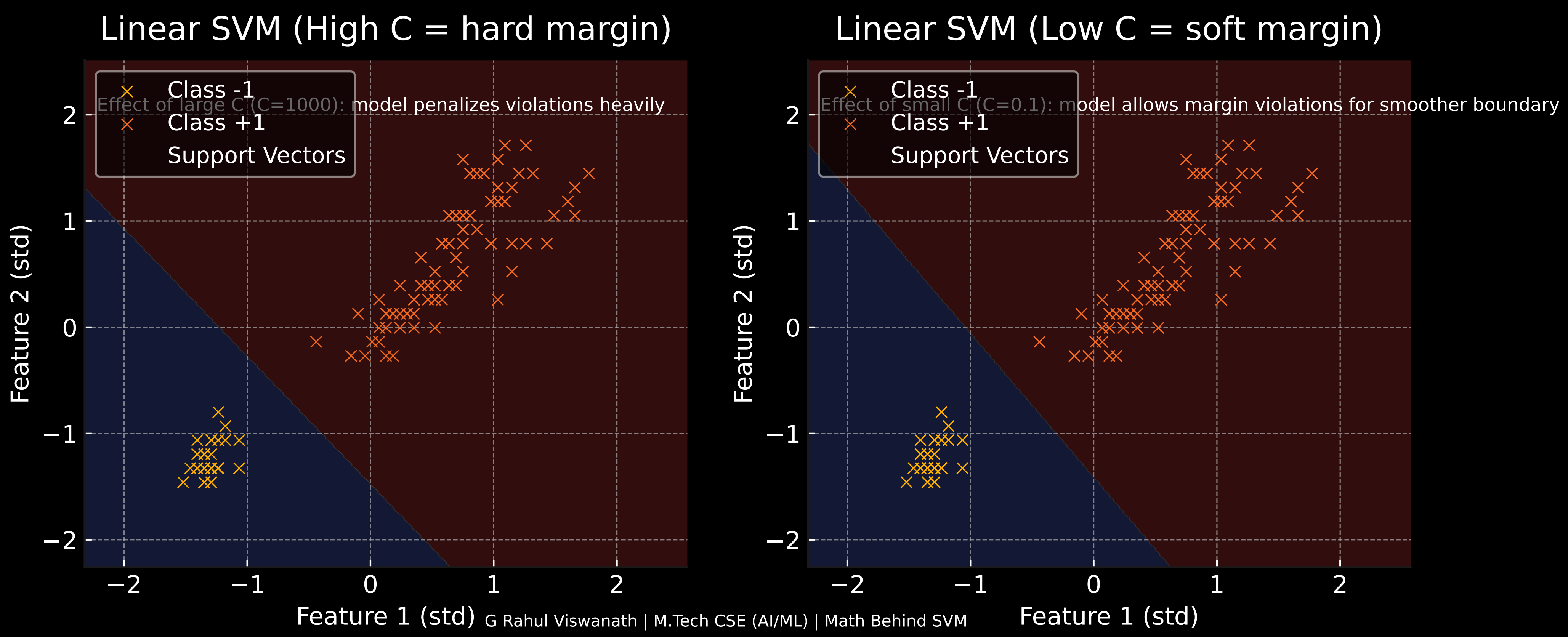
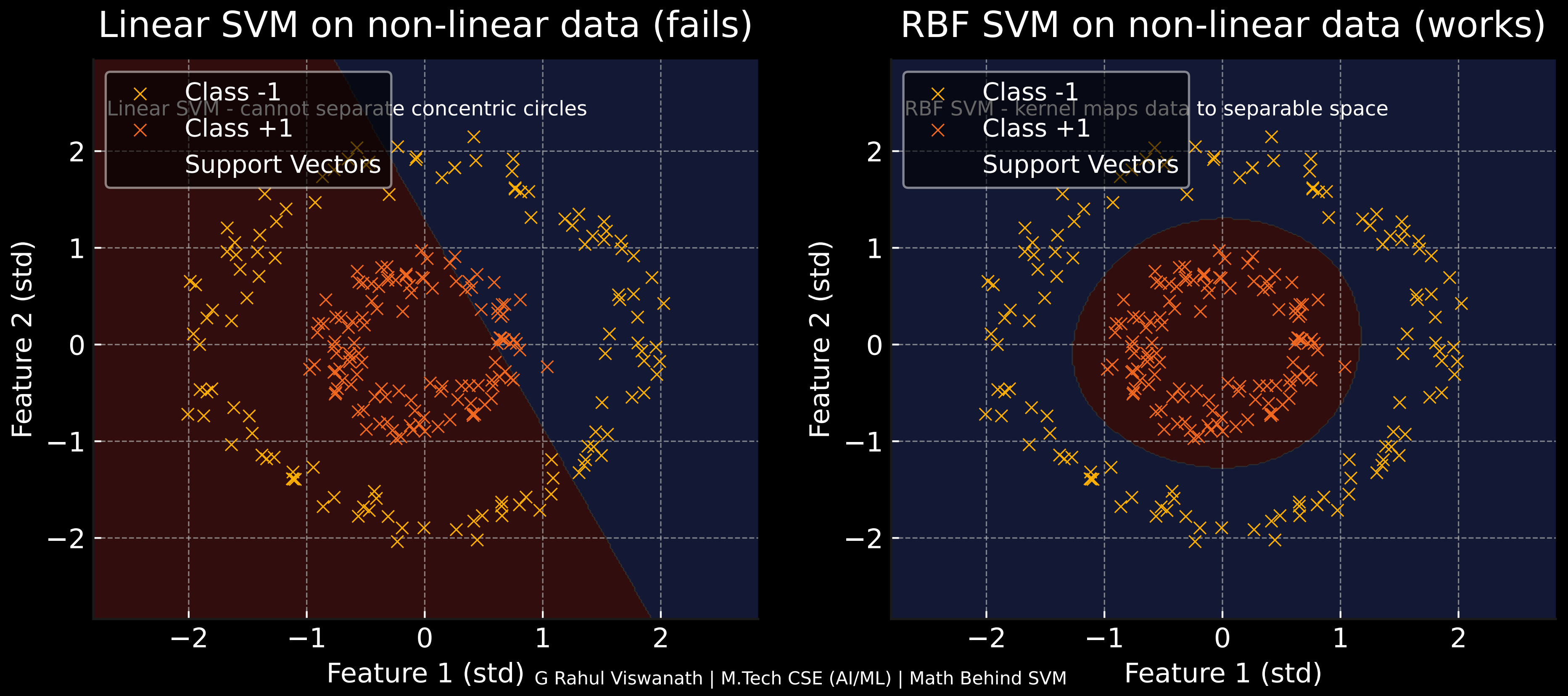

Linear SVM (Iris petal features)
Class -1
Class +1
Support Vectors
Part 1 — Linear SVM (Iris)
This plot shows the learned linear decision boundary and support vectors (yellow circles).
Footer: G Rahul Viswanath | M.Tech CSE (AI/ML) | Math Behind SVM
Feature 1 (std)
Feature 2 (std)
−2
−1
0
1
2


Linear SVM (High C = hard margin)
Class -1
Class +1
Support Vectors
Effect of large C (C=1000): model penalizes violations heavily
Feature 2 (std)
Feature 1 (std)
Linear SVM (Low C = soft margin)
Class -1
Class +1
Support Vectors
Effect of small C (C=0.1): model allows margin violations for smoother boundary
Feature 2 (std)
Feature 1 (std)



G Rahul Viswanath | M.Tech CSE (AI/ML) | Math Behind SVM

Math Behind SVM (Support Vector Machine) - Part 1

By G Rahul Viswanath | M.Tech CSE (AI/ML)

The main objective of SVM is to find the optimal hyperplane that linearly separates data points of different classes by maximizing the margin. This blog is the first in a three-part series covering the mathematical foundation behind SVMs.

1. Linearly separable data
2. Non-separable data with slack variables
3. Non-linear separable data using kernel trick

In this first part, we assume data is perfectly linearly separable and go through the mathematical derivation step-by-step, from the primal to the dual form.

1. Basic Linear Algebra Recap

A vector in R^d has both magnitude and direction. The dot product between two vectors u and v is u·v = ||u|| ||v|| cos?. The hyperplane in SVM is defined using these concepts.

A hyperplane in d-dimensional space is represented as:

w^T x + b = 0

where w is the normal vector to the hyperplane and b is the bias (offset). The sign of (w^T x + b) determines which side of the hyperplane a point lies on.

2. Hyperplane and Margin

For linearly separable data, there are infinite hyperplanes that can separate two classes. The goal of SVM is to find the one with the maximum margin, i.e., the farthest distance from the nearest data points of both classes.

The margin is defined as the perpendicular distance between the two support hyperplanes:

w^T x + b = +1

w^T x + b = -1

The distance between these planes is 2 / ||w||, so maximizing the margin is equivalent to minimizing ||w||.

3. Primal Formulation

Given training data (x_i, y_i) where y_i ? {+1, -1}, we want to find w and b that minimize:

minimize (1/2) ||w||^2

subject to y_i (w^T x_i + b) ? 1 for all i.

This ensures that all points are correctly classified with a margin of at least 1. The optimization problem is convex, which guarantees a unique global minimum.

4. Dual Formulation Using Lagrange Multipliers

To solve the constrained problem, we introduce Lagrange multipliers ?_i ? 0 for each constraint and form the Lagrangian:

L(w, b, ?) = (1/2)||w||^2 - ? ?_i [y_i(w^T x_i + b) - 1]

Setting partial derivatives ?L/?w = 0 and ?L/?b = 0, we obtain:

w = ? ?_i y_i x_i

? ?_i y_i = 0

Substituting these back into the Lagrangian leads to the dual problem:

maximize ? ?_i - 1/2 ?? ?_i ?_j y_i y_j (x_i^T x_j)

subject to ? ?_i y_i = 0 and ?_i ? 0

Once ?* is obtained, the optimal weight vector is:

w* = ? ?*_i y_i x_i

The data points with ?_i > 0 are the support vectors that define the margin.

5. Why Dual Form? (and Kernel Trick Preview)

The dual form is preferred because it depends only on dot products (x_i^T x_j), which allows us to use kernel functions to handle non-linear decision boundaries.

In kernelized SVMs, we replace x_i^T x_j with K(x_i, x_j), where K is a kernel function such as:

- Linear Kernel: K(x, x') = x^T x'
- Polynomial Kernel: K(x, x') = (x^T x' + 1)^d
- RBF Kernel: K(x, x') = exp(-?||x - x'||^2)

6. Summary of Part 1

In this post, we derived the SVM formulation for linearly separable data starting from the concept of hyperplanes to the dual optimization problem.
We learned how support vectors are identified by non-zero ? values and how the margin maximization leads to better generalization.

In the next part (Part 2), we will introduce slack variables to handle non-separable datasets and derive the soft-margin SVM.

References

1. Cortes, C., Vapnik, V. (1995). "Support Vector Networks", Machine Learning Journal.
2. Ankit Kumar, "Math Behind Support Vector Machine (SVM)", Medium, 2021.
3. Scikit-learn Documentation: https://scikit-learn.org/
4. Wikipedia: Sequential Minimal Optimization (SMO) Algorithm.