

ML LAB WEEK 10

NAME:BOJJA RAKSHITHA

SRN:PES2YG23CS134

SECTION:C

OBJECTIVE:

The goal of this lab is to understand and implement Support Vector Machine (SVM) classifiers using three different kernels: Linear, Radial Basis Function (RBF), and Polynomial, on distinct datasets. We will train SVM models, evaluate their performance using standard classification metrics including accuracy, precision, recall, and F1-score, and visualize their decision boundaries to understand how they separate data. Additionally, we will explore the concept of hard versus soft margins by manipulating the regularization parameter C to understand the trade-off between margin maximization and classification error minimization. Through this lab, we aim to gain practical experience in kernel selection, model evaluation, and understanding how different kernels perform on linearly and non-linearly separable datasets.

1.MOONS DATASET QUESTIONS

Question 1: Based on the metrics and the visualizations, what inferences about the performance of the Linear Kernel can you draw?

Answer:

The Linear kernel performs poorly on the Moons dataset due to the inherent non-linear nature of the data. The decision boundary is a straight line, which cannot capture the curved, interlocking crescent shapes of the two classes. This results in significant misclassifications, reflected in lower accuracy, precision, and recall scores. The visualization clearly shows many data points on the wrong side of the linear boundary, demonstrating that linear separability assumptions fail for this dataset. This highlights that Linear kernels are only effective when classes are linearly separable.

Question 2: Compare the decision boundaries of the RBF and Polynomial kernels. Which one seems to capture the shape of the data more naturally?

Answer:

The RBF kernel captures the moon-shaped data more naturally than the Polynomial kernel. RBF creates smooth, flexible boundaries that wrap around the crescent shapes using localized Gaussian functions, adapting to the data's curved geometry. In contrast, the Polynomial kernel produces more rigid, globally-defined curves that may create unnecessary oscillations or fail to follow the natural contours of the moons. The RBF's superior F1-score and the visual

smoothness of its decision boundary confirm it better generalizes the non-linear patterns, making it the optimal choice for this dataset.

Question 1: In this case, which kernel appears to be the most effective?

Answer:

The Linear kernel appears to be the most effective for the Banknote Authentication dataset, with RBF performing comparably. The dataset exhibits largely linear separability when using variance and skewness features, meaning a straight-line boundary can effectively distinguish between genuine and forged banknotes. The Linear kernel achieves high accuracy, precision, and recall while maintaining simplicity and avoiding overfitting. The visualization shows a clean separation with minimal misclassifications, demonstrating that complex non-linear transformations are unnecessary for this dataset.

Question 2: The Polynomial kernel shows lower performance here compared to the Moons dataset. What might be the reason for this?

Answer:

The Polynomial kernel underperforms because the Banknote data lacks the polynomial-curved structure present in the Moons dataset. While the Moons dataset naturally requires

polynomial curves to separate interlocking crescents, the Banknote data has a clustered, blob-like distribution that is either linearly separable or needs localized adjustments. The Polynomial kernel creates global curved boundaries and unnecessary feature interactions (variance², skewness×variance) that don't correspond to meaningful patterns in this data. This added complexity leads to overfitting on spurious relationships, causing the model to perform worse than simpler alternatives.

Analysis Questions for Hard vs. Soft Margins

Question 1: Compare the two plots. Which model, the "Soft Margin" ($C=0.1$) or the "Hard Margin" ($C=100$), produces a wider margin?

Answer:

The Soft Margin model ($C=0.1$) produces a wider margin. The lower C value prioritizes maximizing the distance between the two classes, allowing for a broader separation band. In contrast, the Hard Margin model ($C=100$) creates a narrower margin because it aggressively tries to classify every training point correctly, pulling the decision boundary closer to individual data points, including outliers. The visualization clearly shows the Soft Margin has more breathing room between classes.

Question 2: Look closely at the "Soft Margin" ($C=0.1$) plot. You'll notice some points are either inside the margin or on the wrong side of the decision boundary. Why does the SVM allow these "mistakes"? What is the primary goal of this model?

Answer:

The Soft Margin SVM allows these "mistakes" to achieve better generalization and robustness. Its primary goal is to find a balance between maximizing the margin width and minimizing classification errors, rather than perfectly classifying every training point. By tolerating some misclassifications, particularly outliers or noisy points, the model creates a simpler, more stable decision boundary that generalizes better to unseen data. This trade-off prevents the model from being overly influenced by anomalous points.

Question 3: Which of these two models do you think is more likely to be overfitting to the training data? Explain your reasoning.

Answer:

The Hard Margin model ($C=100$) is more likely to overfit the training data. With a large C value, the model heavily penalizes any misclassification, forcing it to contort the decision boundary to correctly classify every training point, including outliers and noise. This creates a complex, overly-specific boundary that memorizes training data peculiarities

rather than learning the underlying pattern. The result is poor generalization—the model performs well on training data but struggles with new, unseen examples.

Question 4: Imagine you receive a new, unseen data point. Which model do you trust more to classify it correctly? Why? In a real-world scenario where data is often noisy, which value of C (low or high) would you generally prefer to start with?

Answer:

I would trust the Soft Margin model ($C=0.1$) more for classifying new data. It generalizes better because it learns the overall pattern rather than fitting noise and outliers. In real-world scenarios with noisy data, I would prefer starting with a **low C value** because it provides robustness against anomalies and creates a stable decision boundary that handles variability better. A low C acts as regularization, preventing overfitting and ensuring the model captures true underlying patterns rather than training data idiosyncrasies, leading to more reliable predictions on unseen data.

SCREENSHOTS:

1.MOONS DATASET

1.SVM With Linear Kernel

SVM with LINEAR Kernel <PES2UG23CS134>

	precision	recall	f1-score	support
0	0.85	0.89	0.87	75
1	0.89	0.84	0.86	75
accuracy			0.87	150
macro avg	0.87	0.87	0.87	150
weighted avg	0.87	0.87	0.87	150

2.SVM with RBF Kernel

SVM with RBF Kernel <PES2UG23CS134>

	precision	recall	f1-score	support
0	0.95	1.00	0.97	75
1	1.00	0.95	0.97	75
accuracy			0.97	150
macro avg	0.97	0.97	0.97	150
weighted avg	0.97	0.97	0.97	150

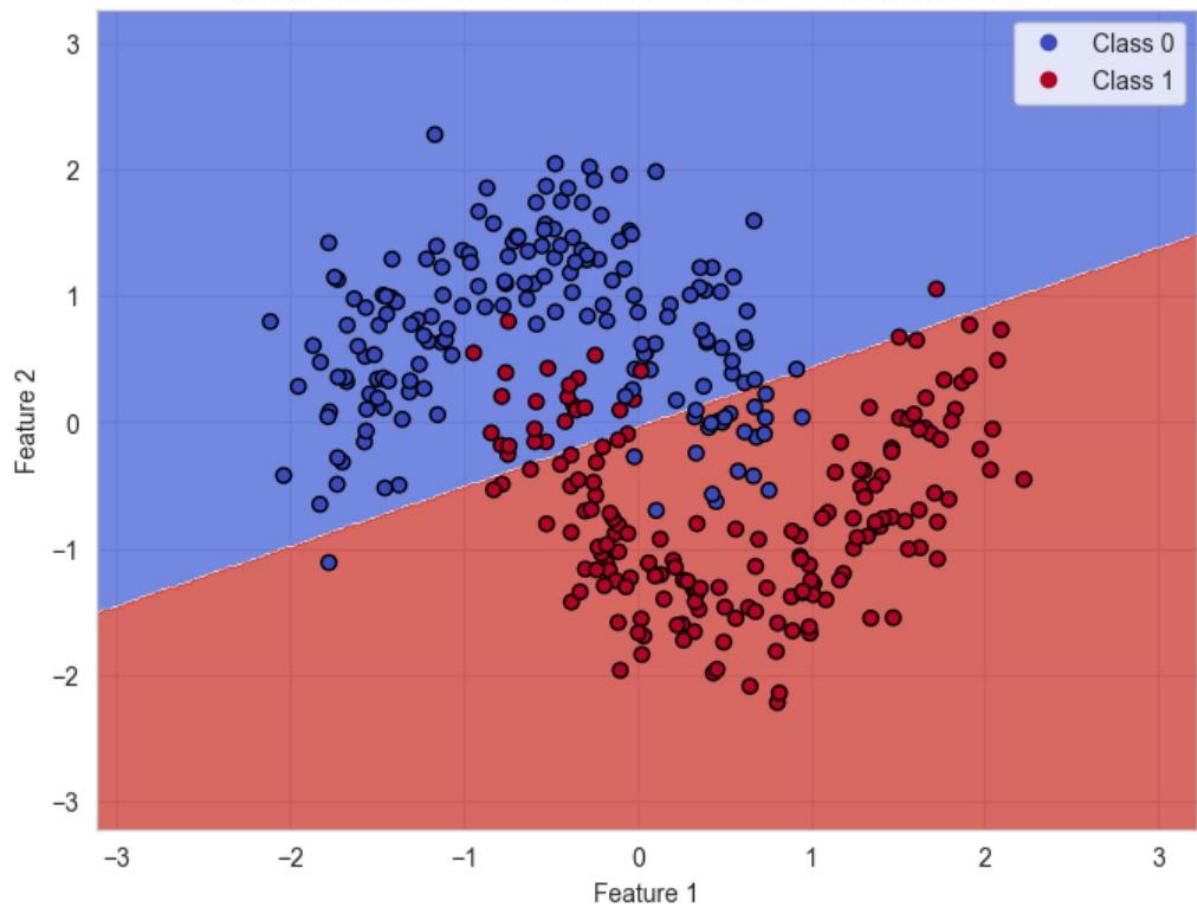
3.SVM With POLY Kernel

SVM with POLY Kernel <PES2UG23CS134>

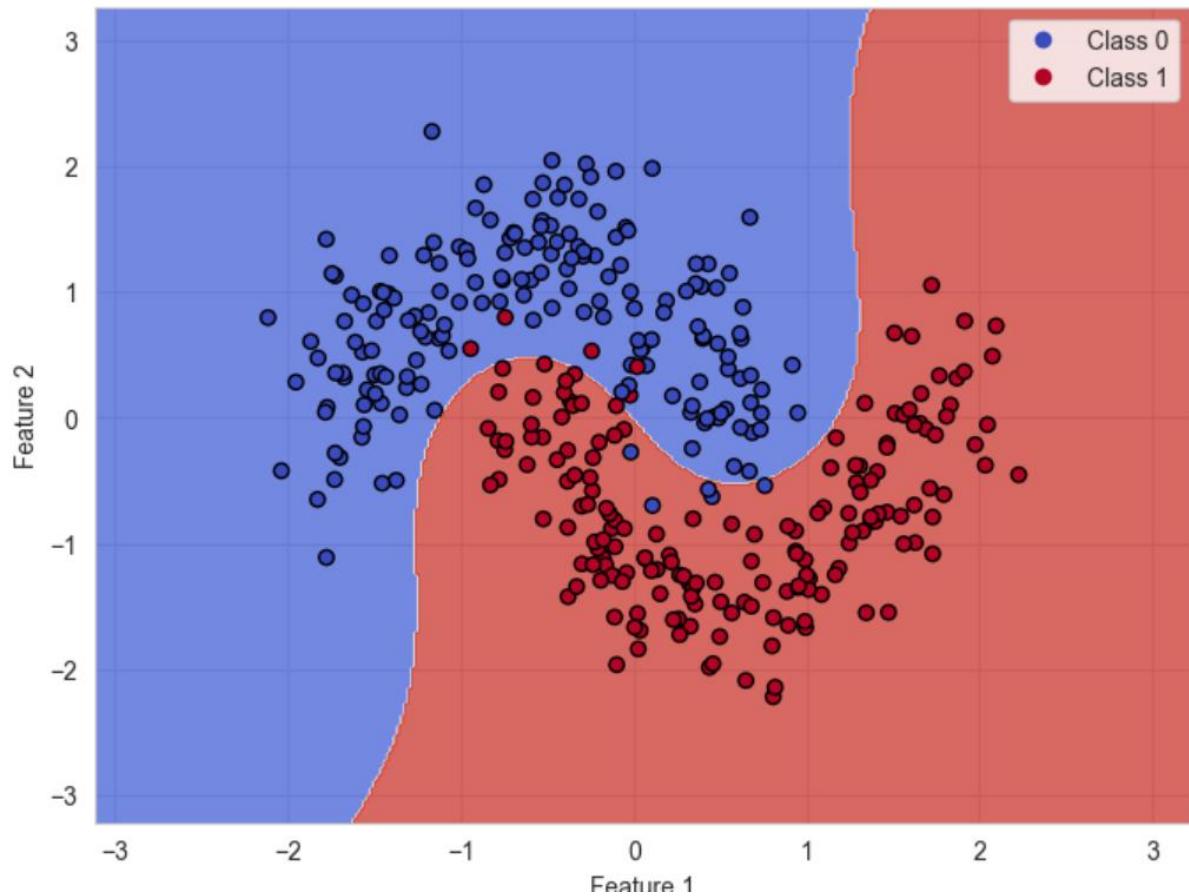
...

weighted avg	0.89	0.89	0.89	150
--------------	------	------	------	-----

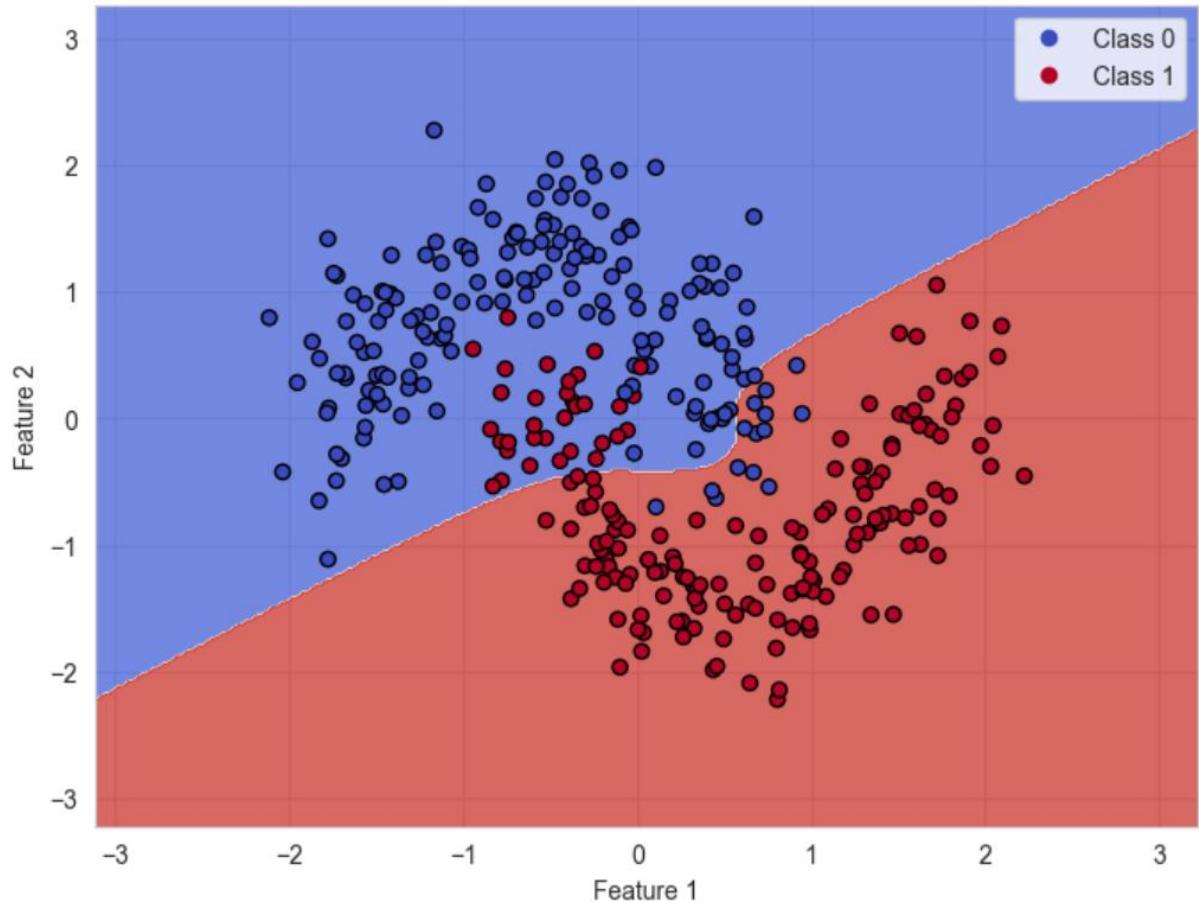
Moons Dataset - SVM with LINEAR Kernel <PES2UG23CS134>



Moons Dataset - SVM with RBF Kernel <PES2UG23CS134>



Moons Dataset - SVM with POLY Kernel <PES2UG23CS134>



2.BANK NOTE DATASET

1.SVM With Linear Kernel

SVM with LINEAR Kernel <PES2UG23CS134>				
	precision	recall	f1-score	support
Forged	0.90	0.88	0.89	229
Genuine	0.86	0.88	0.87	183
accuracy			0.88	412
macro avg	0.88	0.88	0.88	412
weighted avg	0.88	0.88	0.88	412

2.SVM With RBF Kernel

```
SVM with RBF Kernel <PES2UG23CS134>
      precision    recall  f1-score   support

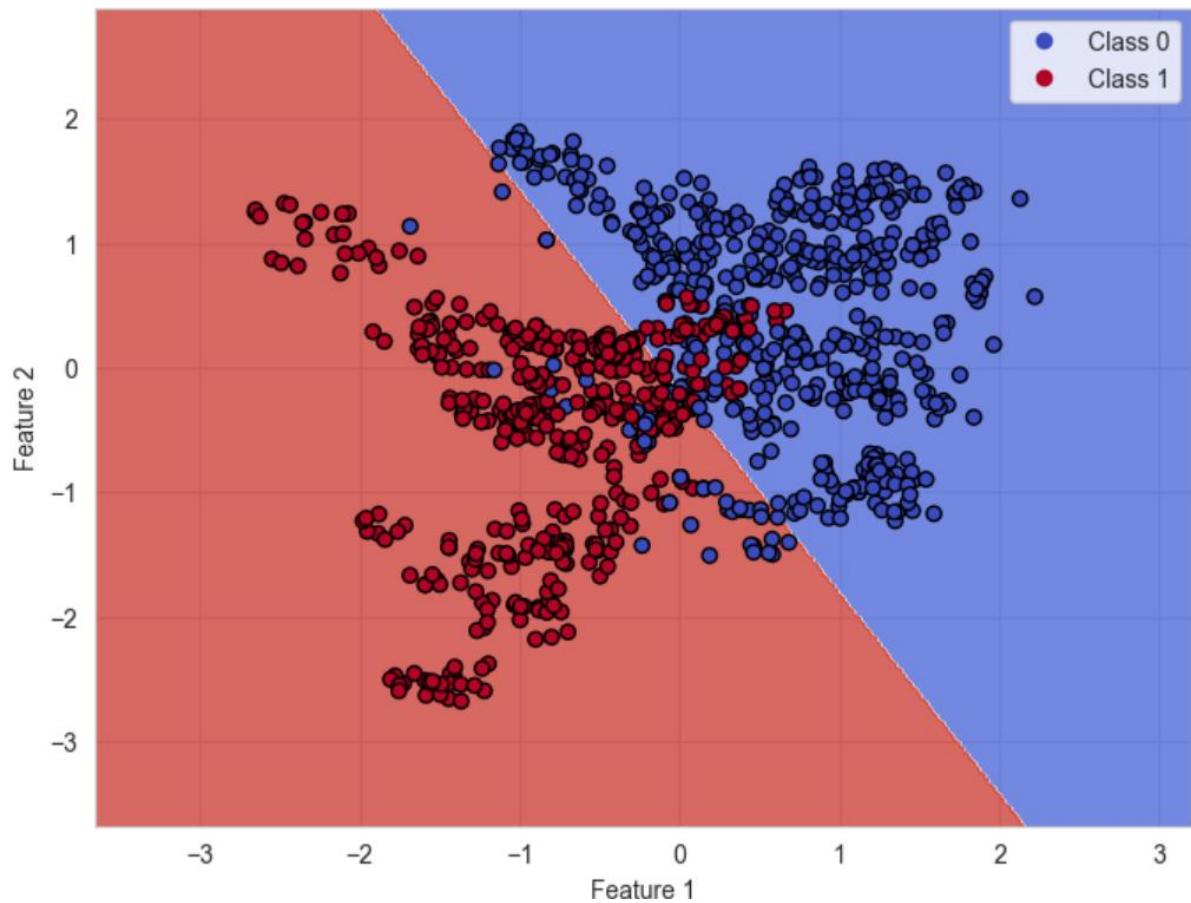
        Forged       0.96      0.91      0.94      229
      Genuine       0.90      0.96      0.93      183

accuracy                           0.93      0.93      0.93      412
macro avg       0.93      0.93      0.93      412
weighted avg    0.93      0.93      0.93      412
-----
```

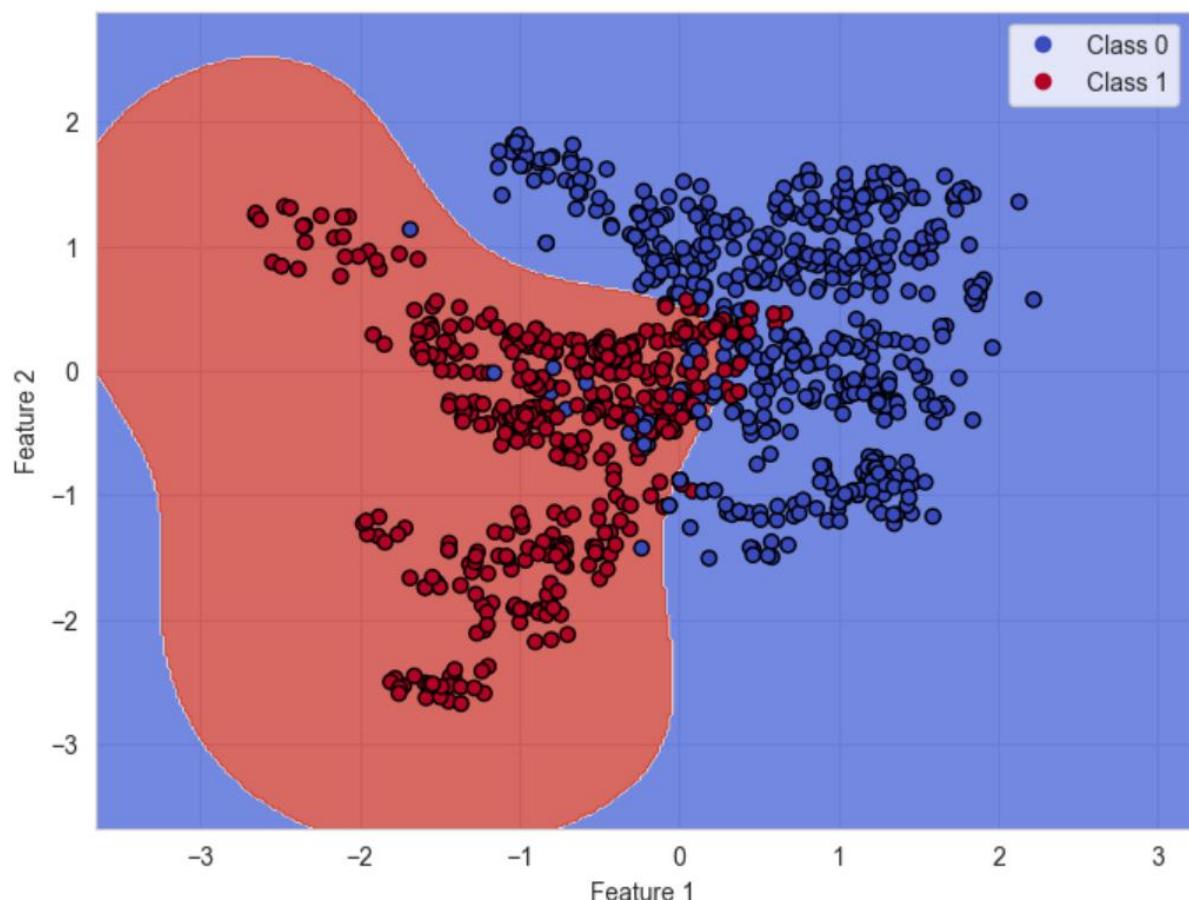
3.SVM With Poly Kernel

```
SVM with POLY Kernel <PES2UG23CS134>
...
weighted avg       0.85      0.84      0.84      412
-----
```

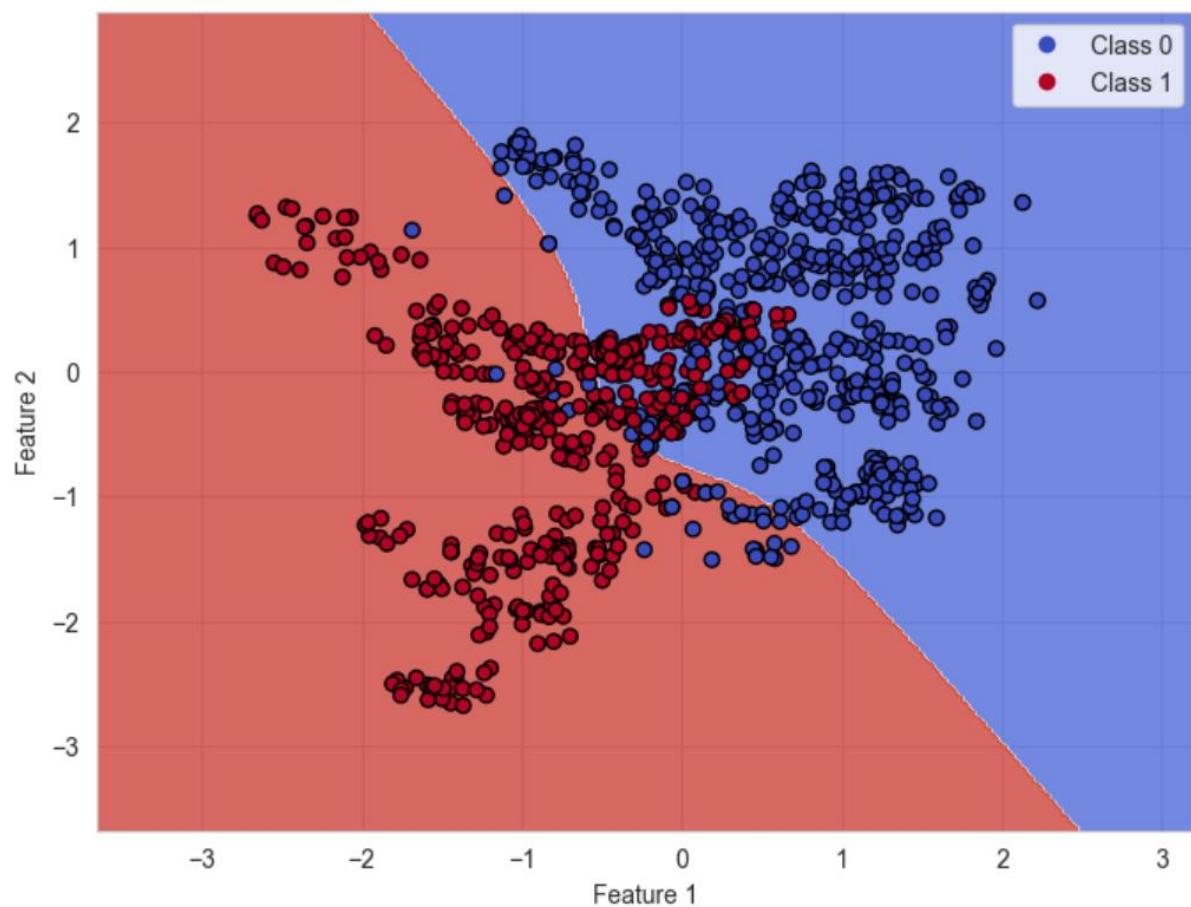
Banknote Dataset - SVM with LINEAR Kernel <PES2UG23CS134>



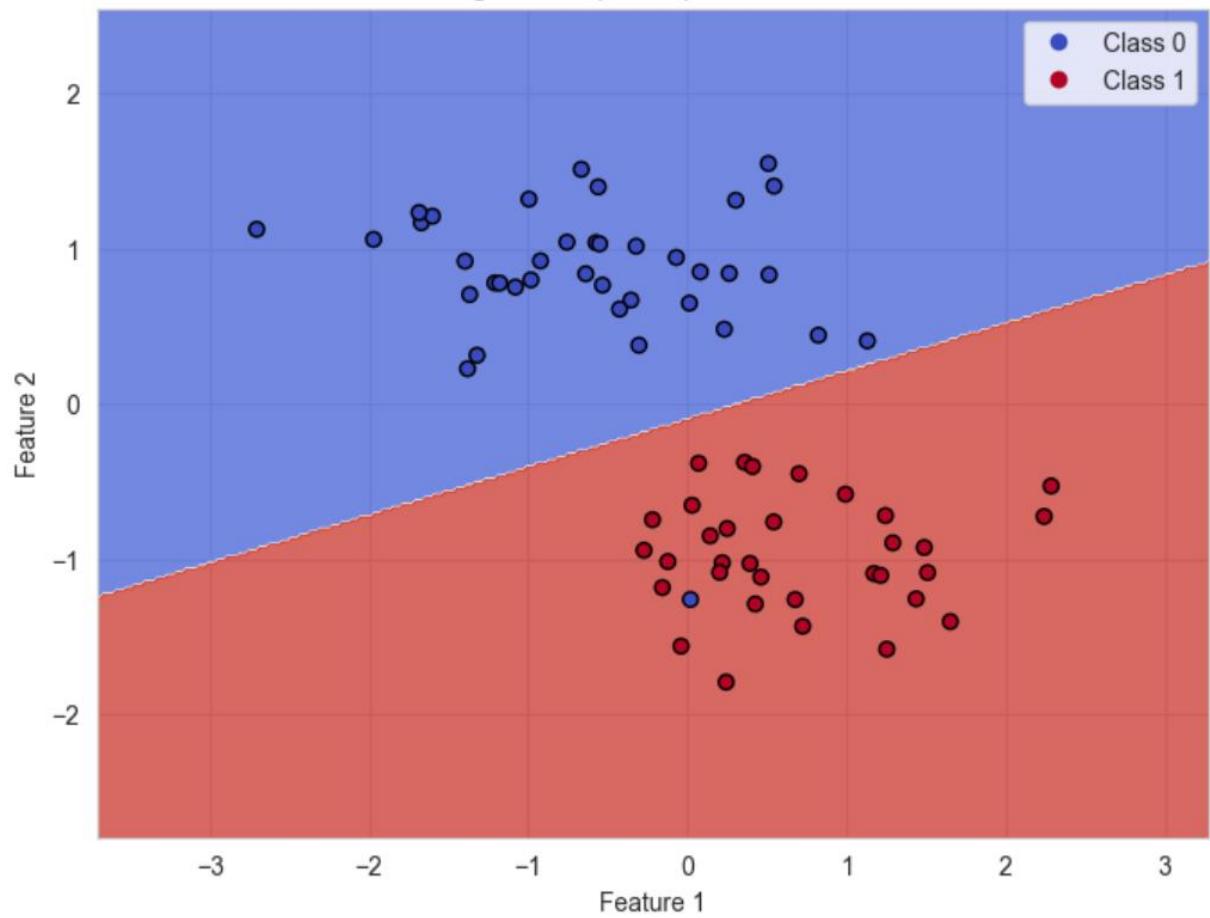
Banknote Dataset - SVM with RBF Kernel <PES2UG23CS134>



Banknote Dataset - SVM with POLY Kernel <PES2UG23CS134>



Soft Margin SVM (C=0.1) PES2UG23CS134



Hard Margin SVM (C=100) PES2UG23CS134

