

ML Lab Week 10 SVM

NAME: Nehal.g

SRN: PES2UG23CS380

SECTION:F

Analysis Questions:

Moons Dataset Questions (2 questions):

1. Inferences about the Linear Kernel's performance.

From both the metrics and the plot, it's clear that the Linear Kernel doesn't perform very well on the Moons dataset. Although it achieves around 87% accuracy, the decision boundary is almost a straight line, which isn't suitable for a dataset that's clearly non-linear and curved in shape. Many points near the center are misclassified because the linear boundary cannot properly separate the two moon-shaped clusters. The Linear Kernel underfits the data.

2. Comparison between RBF and Polynomial kernel decision boundaries.

The RBF Kernel produces a smooth, flexible boundary that curves naturally around the data. It adapts to the moon-like shape almost perfectly, giving the highest accuracy of 98%. This shows that it's very effective for non-linear patterns like this one. On the other hand, the Polynomial Kernel performs better than the linear one (around 91% accuracy) but still doesn't capture the shape as well as the RBF. Its boundary is slightly curved but not as precise, leading to a few misclassifications around the edges.

Banknote Dataset Questions (2 questions):

1. Which kernel was most effective for this dataset?

The RBF kernel is the most effective in this case. It achieves the highest accuracy of 93%, along with strong precision and recall for both the forged and genuine classes. This shows that it captures the complex, non-linear relationships in the data much better than the linear or polynomial kernels. The RBF kernel's flexibility allows it to fit better to the boundaries.

2. Why might the Polynomial kernel have underperformed here?

The Polynomial kernel performs worse here mainly due to overfitting. In this forgery dataset, the data is more complex and noisy, so the polynomial boundary becomes too wavy and starts fitting noise instead of real patterns. In contrast, the Moons dataset has a simpler, cleaner structure that the polynomial kernel can model more easily, which is why it performs better there.

Hard vs. Soft Margin Questions (4 questions):

1. Which margin (soft or hard) is wider?

The Soft Margin SVM with $C=0.1$ produces a wider margin compared to the Hard Margin SVM with $C=100$. This is because a smaller value of C allows the model to focus more on maximizing the margin and less on perfectly classifying all training points.

2. Why does the soft margin model allow "mistakes"?

The Soft Margin SVM allows a few points to lie inside the margin or even on the wrong side of the decision boundary to handle noise and overlapping data better. Its primary goal is not to classify every point correctly but to

find a balance between having a large margin and minimizing misclassification errors. The model achieves better generalization, which means it performs well not just on the training data but also on unseen data.

3. Which model is more likely to be overfitting and why?

The Hard Margin SVM ($C=100$) is more likely to overfit the training data. Because of its high C value, the model tries to classify every training point correctly, even if it means forming a very tight decision boundary that closely follows the data points. This behavior makes the model sensitive to small fluctuations or noise in the dataset, which can hurt its performance when exposed to new, unseen data.

4. Which model would you trust more for new data and why?

The Soft Margin SVM ($C=0.1$) would be more trustworthy for classifying new, unseen data. Since it allows a few errors during training, it learns a more general decision boundary that captures the underlying pattern of the data rather than fitting every specific detail. This makes it more robust and reliable when dealing with new or slightly different inputs.

Training Results (6 Screenshots):

Moons Dataset (3 screenshots):

1. Classification Report for SVM with LINEAR Kernel with SRN

SVM with LINEAR Kernel PES2UG23CS380					
	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.

Classification Report for SVM with RBF Kernel with SRN

SVM with RBF Kernel PES2UG23CS380					
	precision	recall	f1-score	support	
0	0.96	1.00	0.98	75	
1	1.00	0.96	0.98	75	
accuracy			0.98	150	
macro avg	0.98	0.98	0.98	150	
weighted avg	0.98	0.98	0.98	150	

Classification Report for SVM with POLY Kernel with SRN

SVM with POLY Kernel PES2UG23CS380					
	precision	recall	f1-score	support	
0	0.93	0.88	0.90	75	
1	0.89	0.93	0.91	75	
accuracy			0.91	150	
macro avg	0.91	0.91	0.91	150	
weighted avg	0.91	0.91	0.91	150	

Banknote Dataset (3 screenshots):

Classification Report for SVM with LINEAR Kernel

SVM with LINEAR Kernel PES2UG23CS380					
	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	

Classification Report for SVM with RBF Kernel

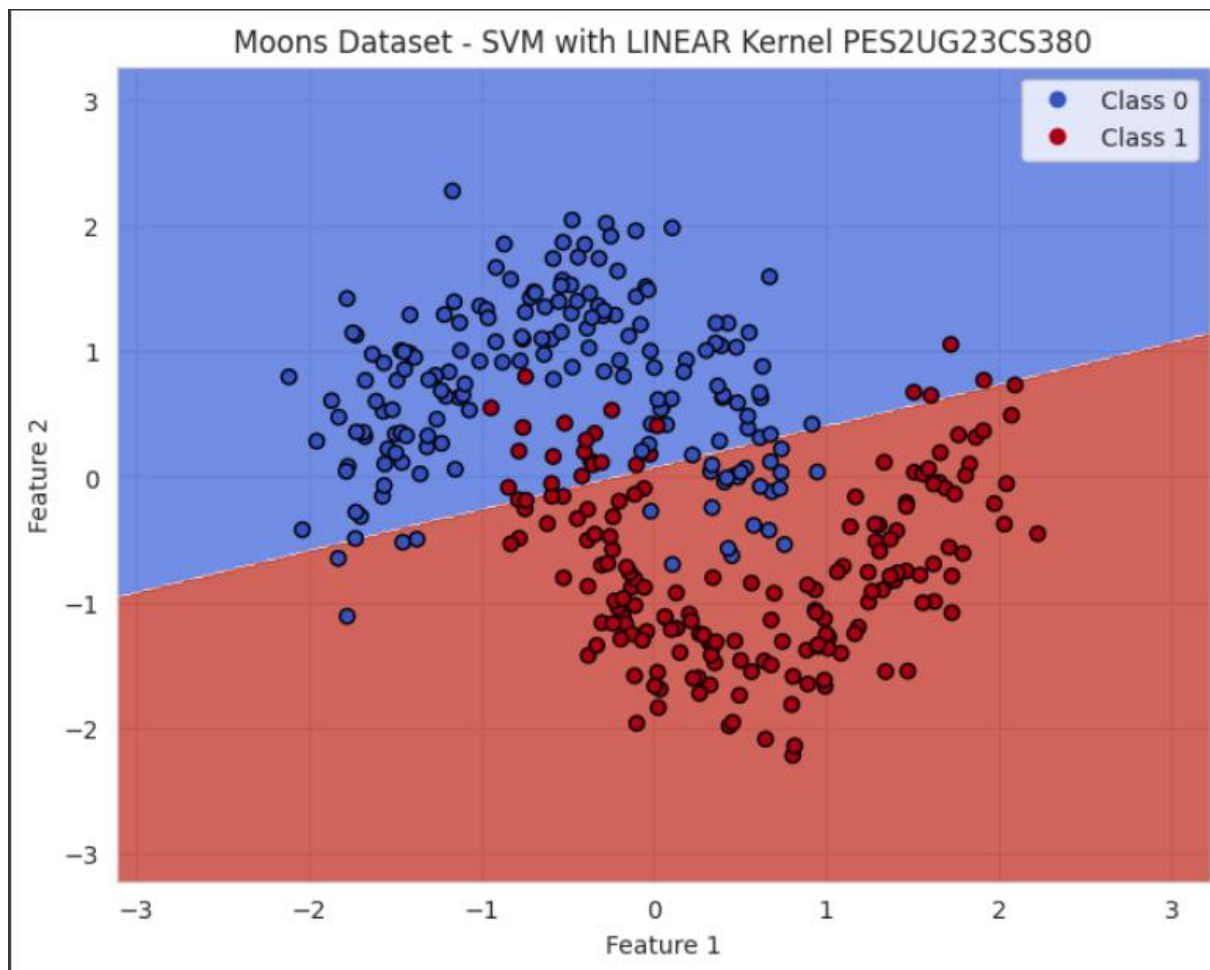
SVM with RBF Kernel PES2UG23CS380					
	precision	recall	f1-score	support	
0	0.96	1.00	0.98	75	
1	1.00	0.96	0.98	75	
accuracy			0.98	150	
macro avg	0.98	0.98	0.98	150	
weighted avg	0.98	0.98	0.98	150	

Classification Report for SVM with POLY Kernel

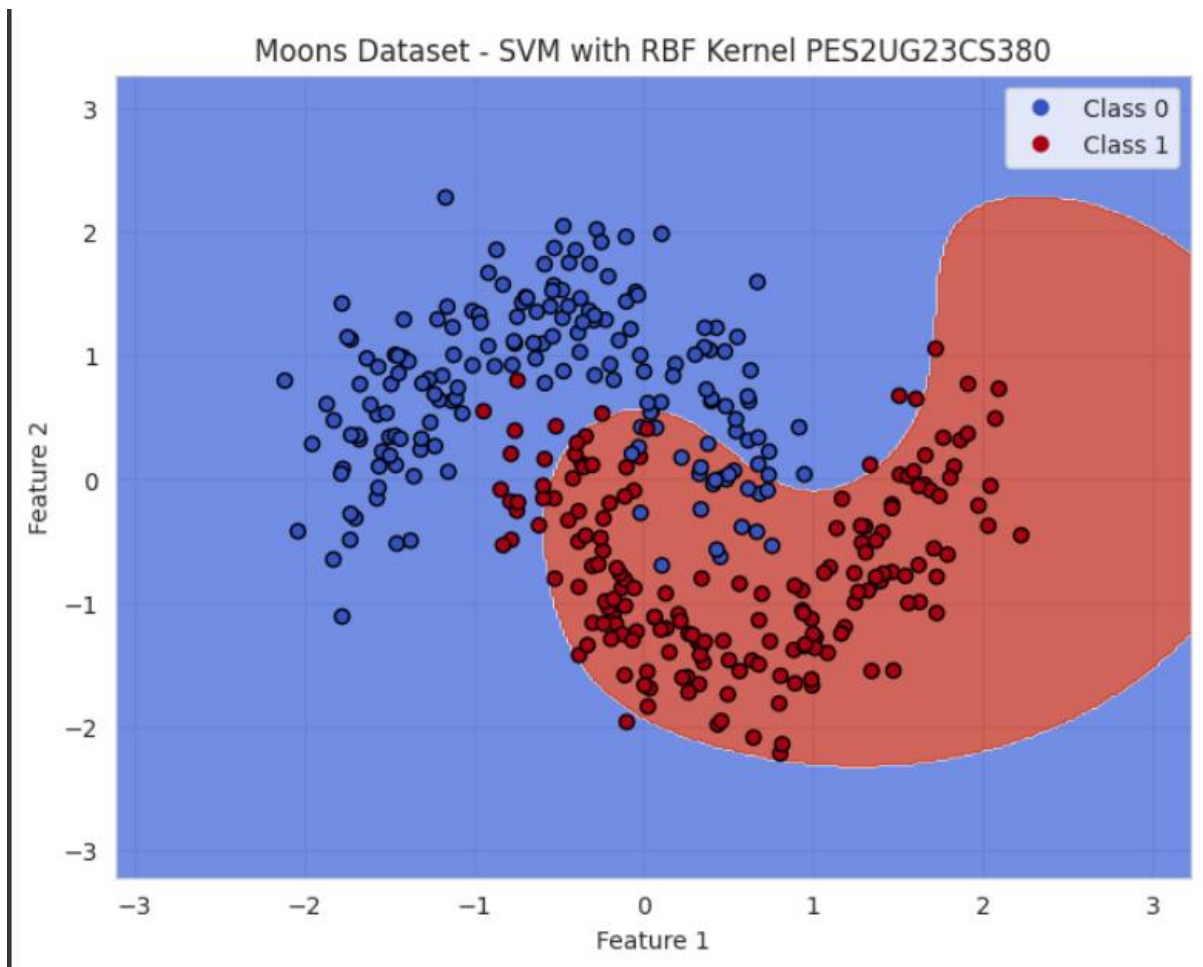
SVM with POLY Kernel PES2UG23CS380					
	precision	recall	f1-score	support	
0	0.93	0.88	0.90	75	
1	0.89	0.93	0.91	75	
accuracy			0.91	150	
macro avg	0.91	0.91	0.91	150	
weighted avg	0.91	0.91	0.91	150	

Decision Boundary Visualizations (8 Screenshots):

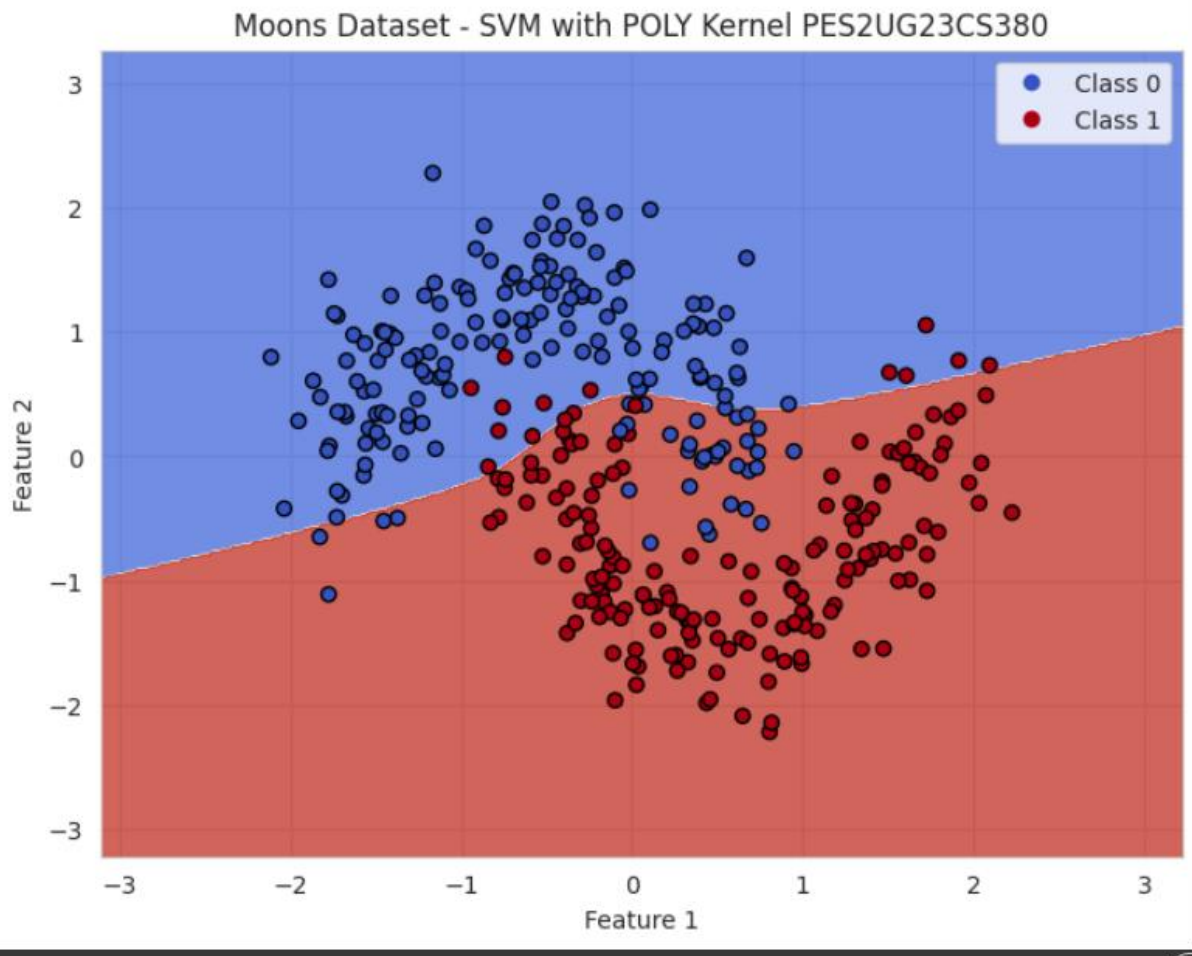
Moons Dataset - SVM with LINEAR Kernel



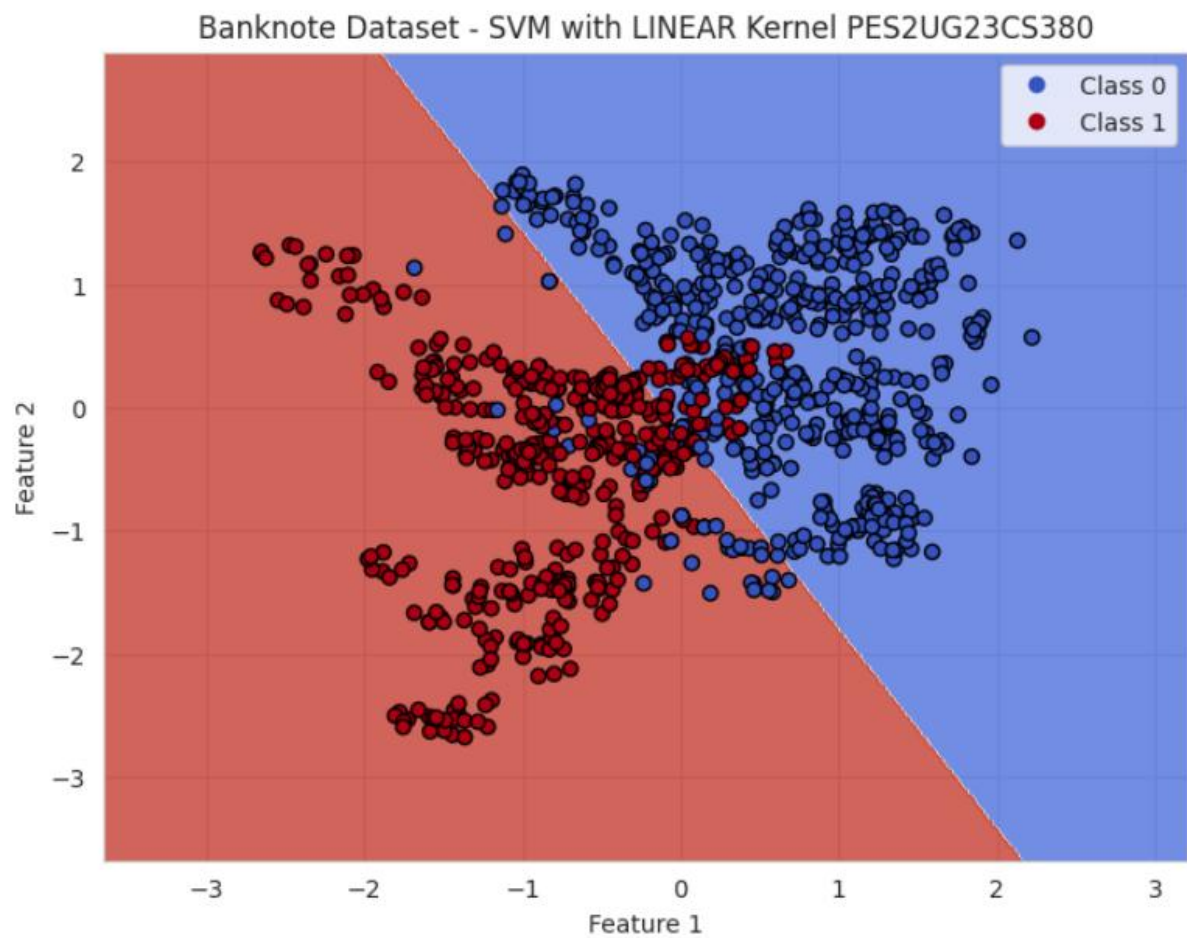
Moons Dataset - SVM with RBF Kernel



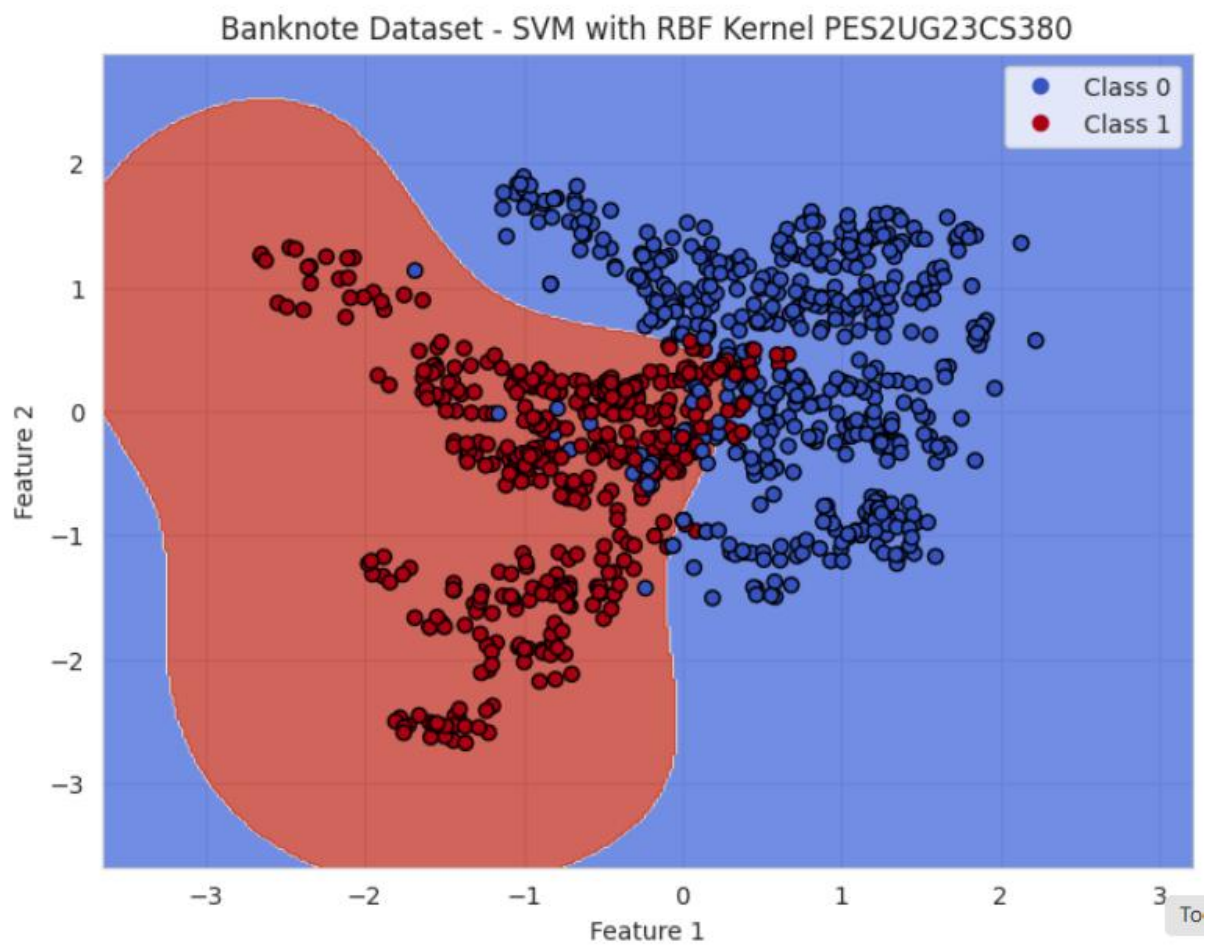
Moons Dataset - SVM with POLY Kernel



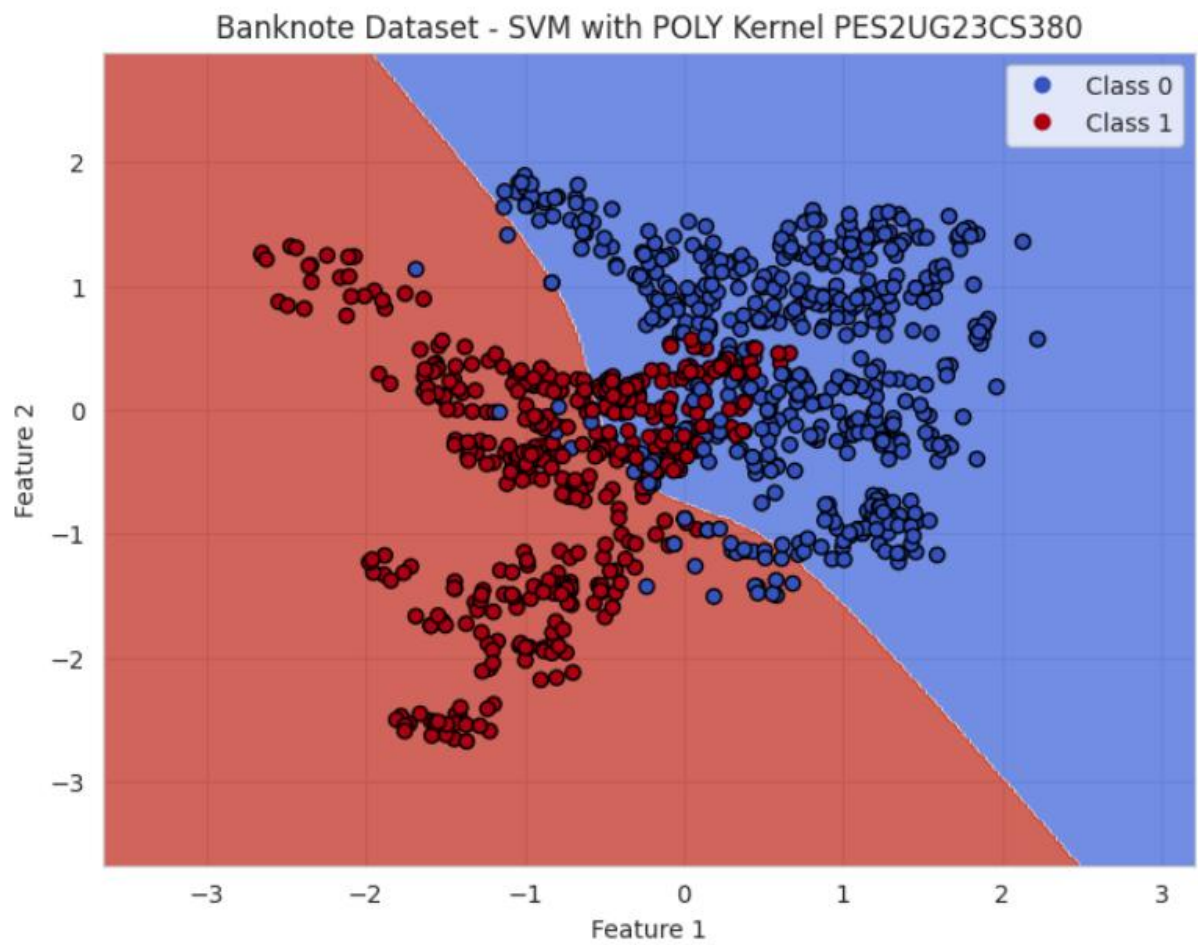
Banknote Dataset - SVM with LINEAR Kernel



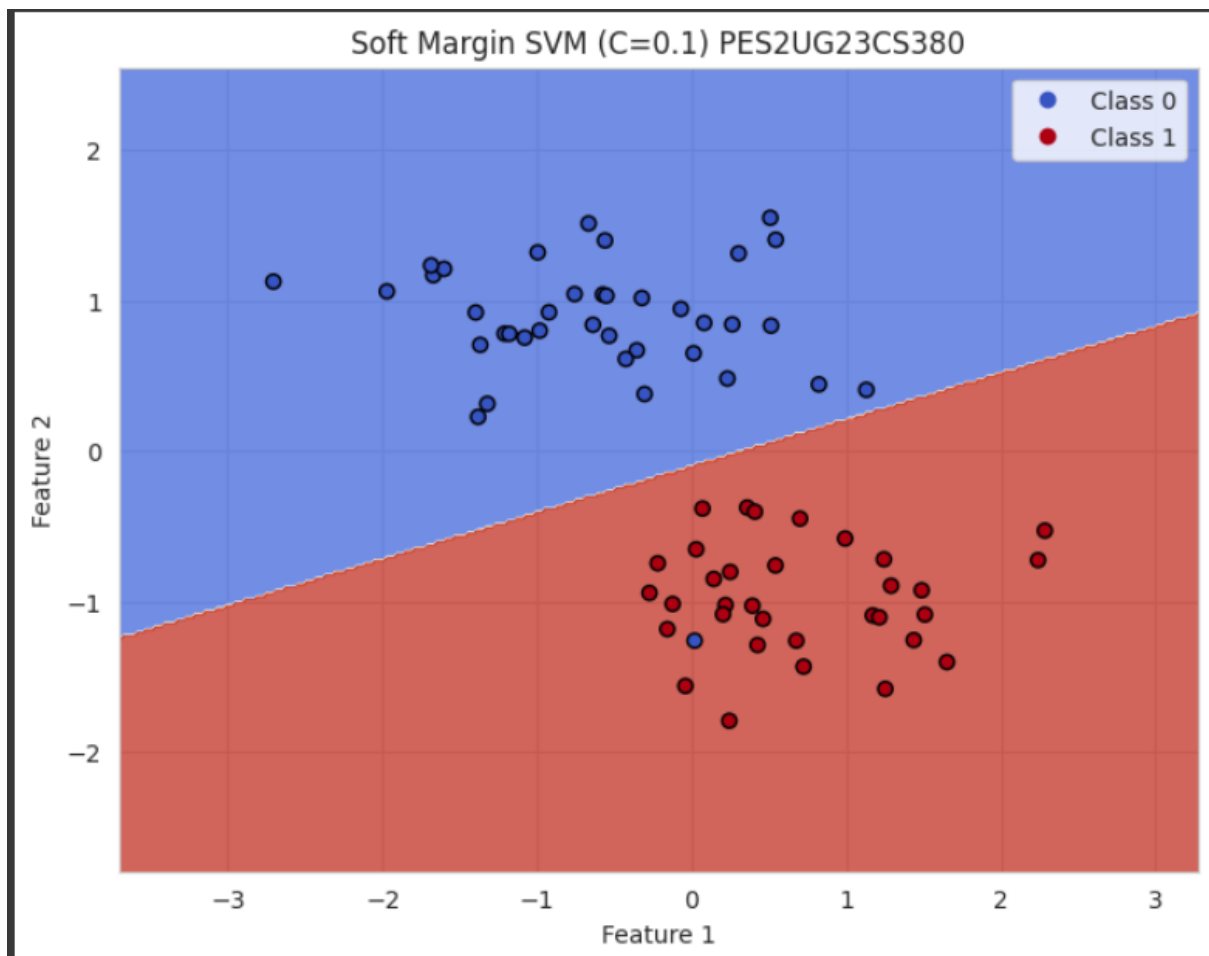
Banknote Dataset - SVM with RBF Kernel



Banknote Dataset - SVM with POLY Kernel



Soft Margin SVM ($C=0.1$)



Hard Margin SVM (C=100)

