ML Lab Week 10 SVM Lab Report

Name: Ninad Chavan

SRN: PES2UG23CS392

Date:15-11-2025

Section: F

<u>Analysis Questions for Moons:</u>

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

The Linear kernel struggled with the Moons dataset due to the data's curved, non-linear structure that cannot be effectively divided by a straight boundary. Consequently, the Linear SVM produced more errors and achieved reduced precision and recall when compared to non-linear kernels. Put simply, it failed to adequately fit the data because it was unable to represent the dataset's curved shape.

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

The RBF and Polynomial kernels are both capable of managing the curved patterns present in the Moons dataset. That said, the RBF kernel typically generates a smoother and more organic decision boundary that adapts better to the data's structure, whereas the Polynomial kernel may produce boundaries that are either too rigid or excessively intricate based on its degree parameter.

<u>Analysis Questions for Banknote:</u>

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

The RBF kernel typically delivers optimal performance on the Banknote dataset. While the data is nearly linearly separable and the linear kernel achieves good results, the RBF kernel's adaptability enables it to create a more accurate decision boundary between classes, frequently resulting in complete separation and superior accuracy along with the best F1-score.

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

The Polynomial kernel's effectiveness tends to be highly dependent on hyperparameter configuration (such as the polynomial's degree). For this particular dataset, a default degree polynomial may fail to generate a decision boundary that aligns with the data distribution as effectively as the linear or RBF kernels. It could produce a boundary that is either overly simplistic or unnecessarily complicated, resulting in classification errors that the alternative kernels manage to avoid.

<u>Analysis Questions for Hard and Soft Margin</u>

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

The Soft Margin SVM using C = 0.1 creates a broader margin. This is evident in the visualization where the decision boundary maintains greater distance from the nearest data points in comparison to the Hard Margin (C = 100).

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?

The Soft Margin model permits certain points to lie within the margin or even be misclassified because its primary objective isn't flawless classification of training data but instead to maximize the margin and enhance generalization capability. It tolerates a small number of classification errors, which ultimately helps the model achieve better performance on new, previously unseen data.

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

The Hard Margin SVM (C = 100) has a greater tendency to overfit the training data. It attempts excessively to correctly classify each individual training point, which can cause it to become overly responsive to noise or outliers and diminish its effectiveness on new data.

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?

It would be preferable to rely on the Soft Margin model (C = 0.1) more for new, unseen data because it prioritizes generalization rather than memorizing the training points. In real-world scenarios where data is frequently noisy and imperfect, it's typically advisable to begin with a low C value since it's more resilient and less prone to overfitting.

SCREENSHOTS

Training Results

Moons Dataset (3 screenshots):

Classification Report for SVM with LINEAR Kernel with SRN

```
SVM with LINEAR Kernel <PES2UG23CS392>
              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 with SRN

```
SVM with RBF Kernel <PES2UG23CS392>
              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

----------------------------------------
```

Classification Report for SVM with POLY Kernel with SRN

```
SVM with POLY Kernel <PES2UG23CS392>
              precision    recall  f1-score   support

           0       0.85      0.95      0.89        75
           1       0.94      0.83      0.88        75

    accuracy                           0.89       150
   macro avg       0.89      0.89      0.89       150
weighted avg       0.89      0.89      0.89       150

----------------------------------------
```

Banknote Dataset (3 screenshots):

Classification Report for SVM with LINEAR Kernel

```
SVM with LINEAR Kernel <PES2UG23CS392>
              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

----------------------------------------
```

Classification Report for SVM with RBF Kernel

```
SVM with RBF Kernel <PES2UG23CS392>
              precision    recall  f1-score   support

      Forged       0.96      0.91      0.94       229
     Genuine       0.90      0.96      0.93       183

    accuracy                           0.93       412
   macro avg       0.93      0.93      0.93       412
weighted avg       0.93      0.93      0.93       412

----------------------------------------
```

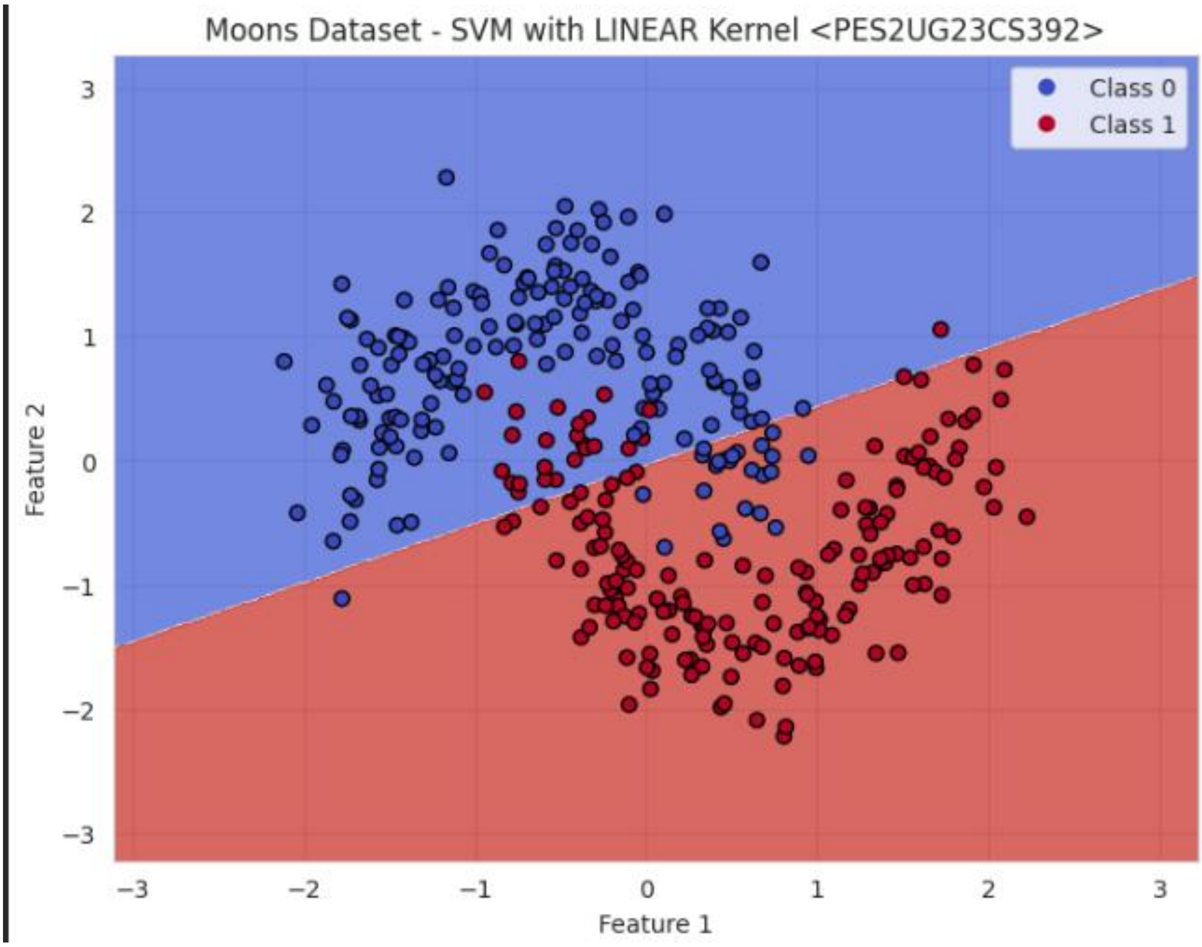
Classification Report for SVM with POLY Kernel

```
SVM with POLY Kernel <PES2UG23CS392>
              precision    recall  f1-score   support

      Forged       0.82      0.91      0.87       229
     Genuine       0.87      0.75      0.81       183

    accuracy                           0.84       412
   macro avg       0.85      0.83      0.84       412
weighted avg       0.85      0.84      0.84       412

----------------------------------------
```

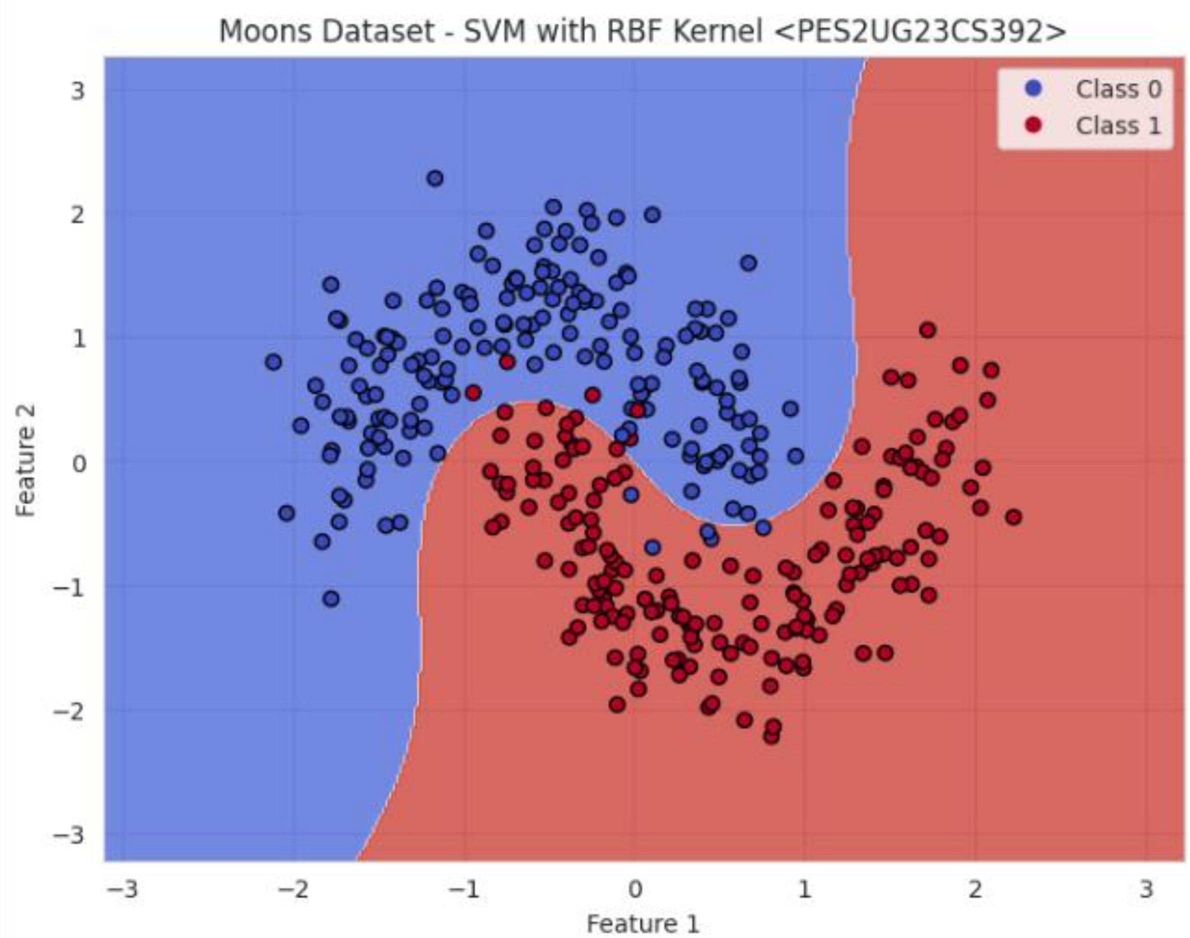
Decision Boundary Visualizations Moons

Dataset (3 plots):

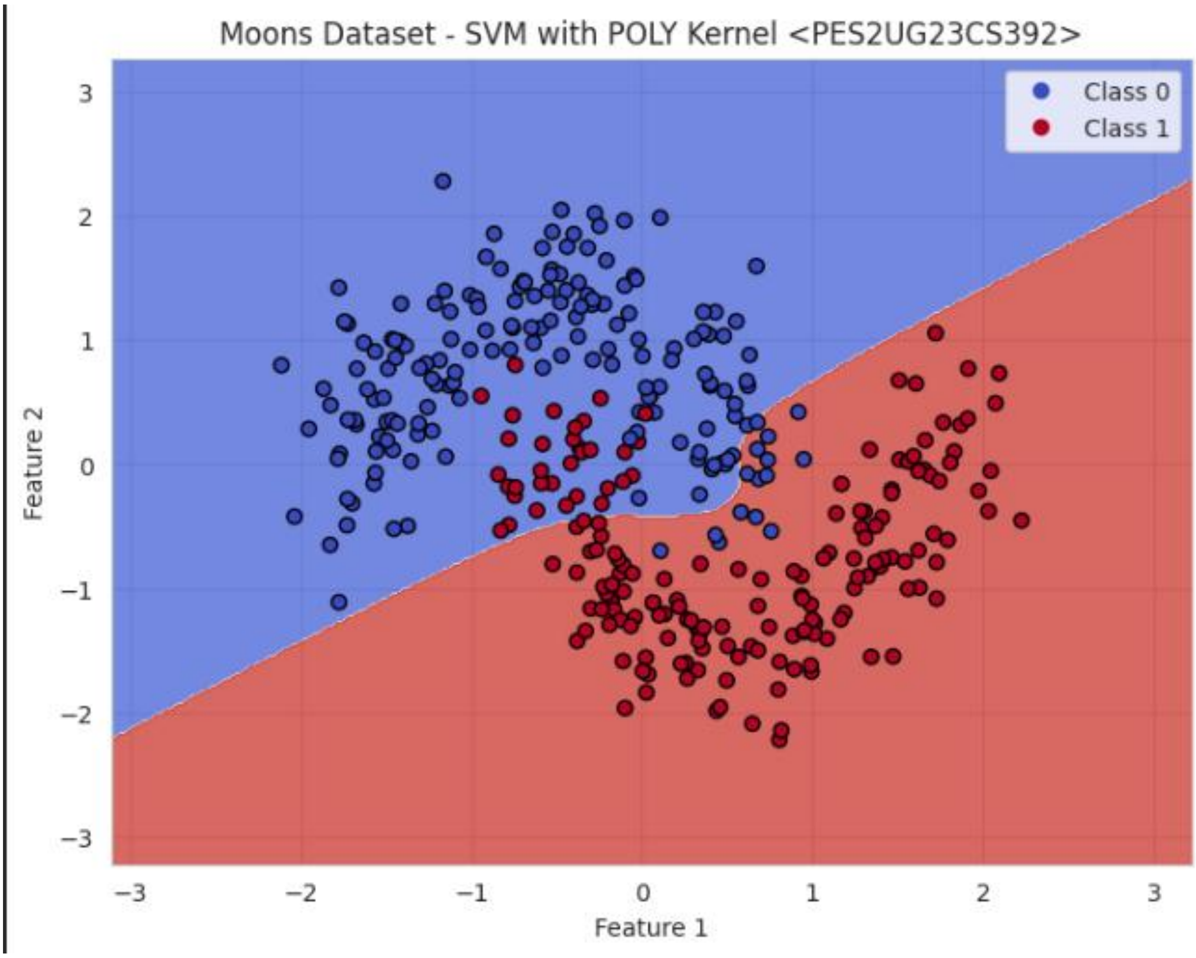
Moons Dataset - SVM with LINEAR Kernel



Moons Dataset - SVM with RBF Kernel

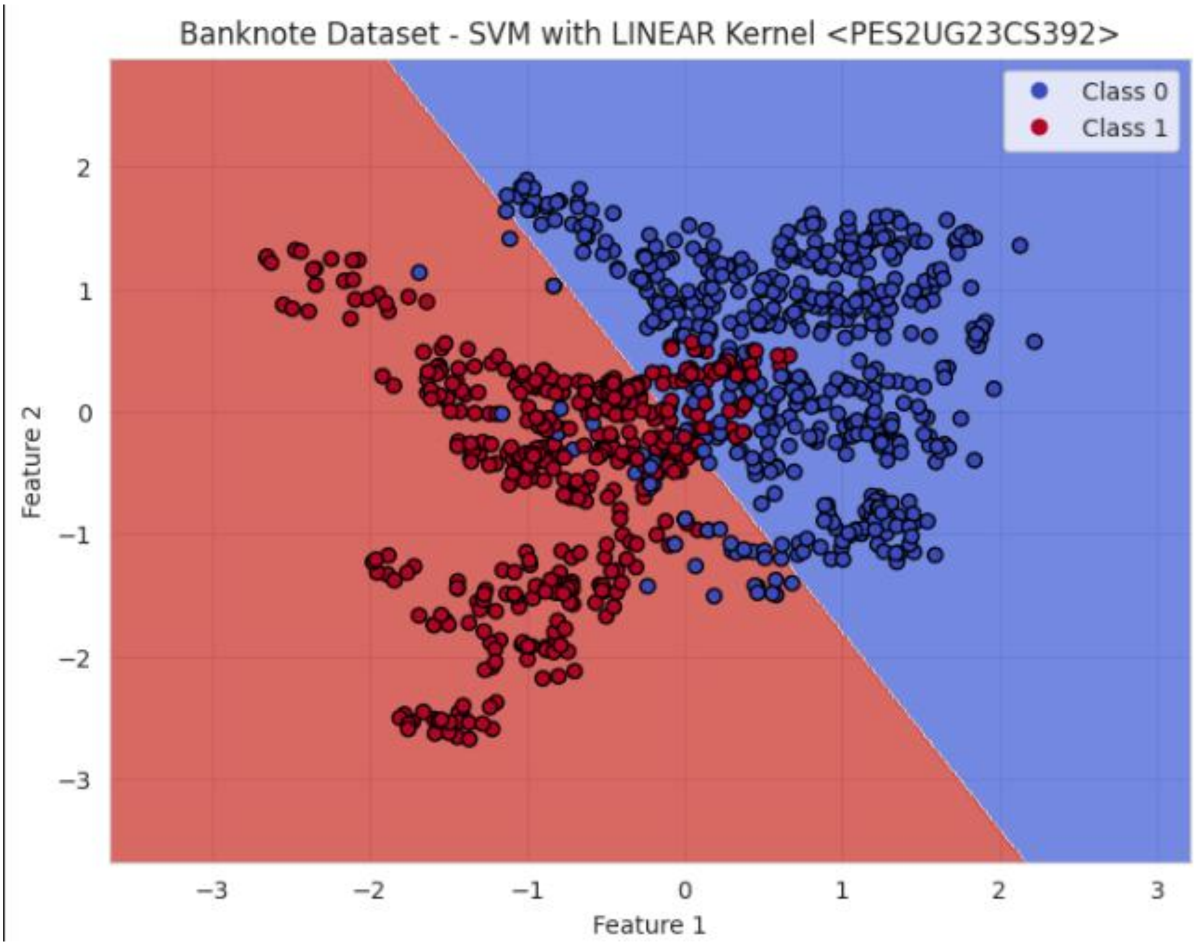


Moons Dataset - SVM with POLY Kernel

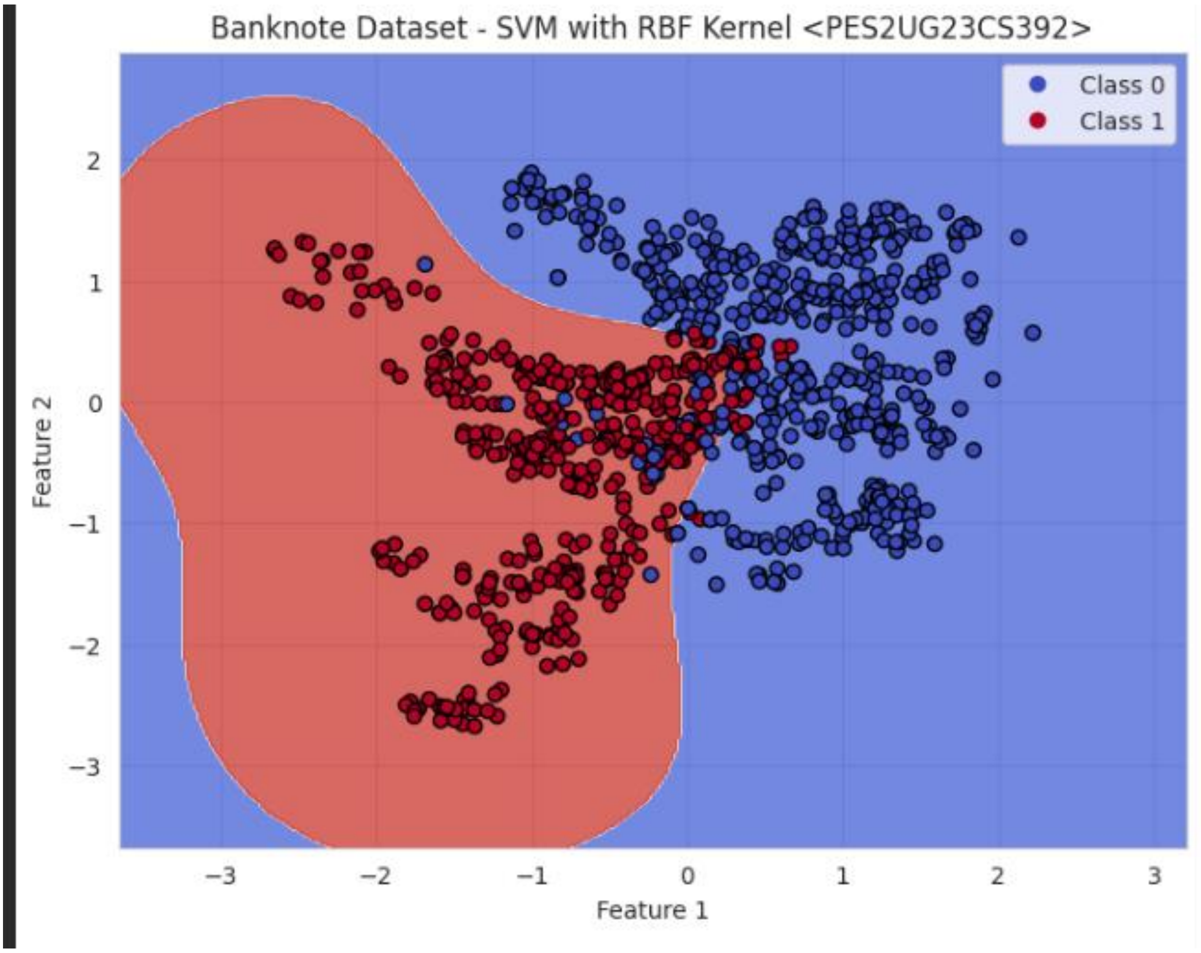


Banknote Dataset (3 plots):

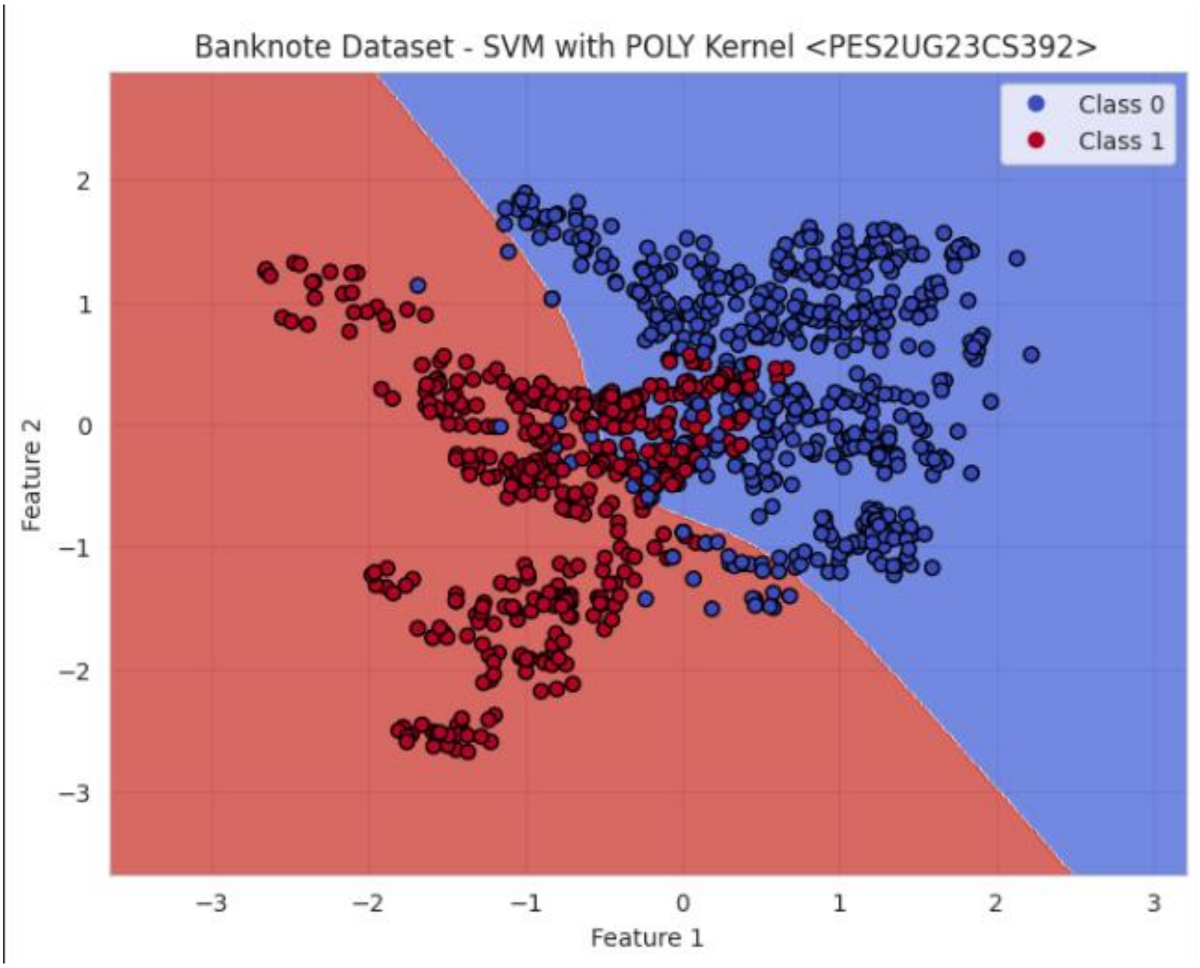
Banknote Dataset - SVM with LINEAR Kernel



Banknote Dataset - SVM with RBF Kernel

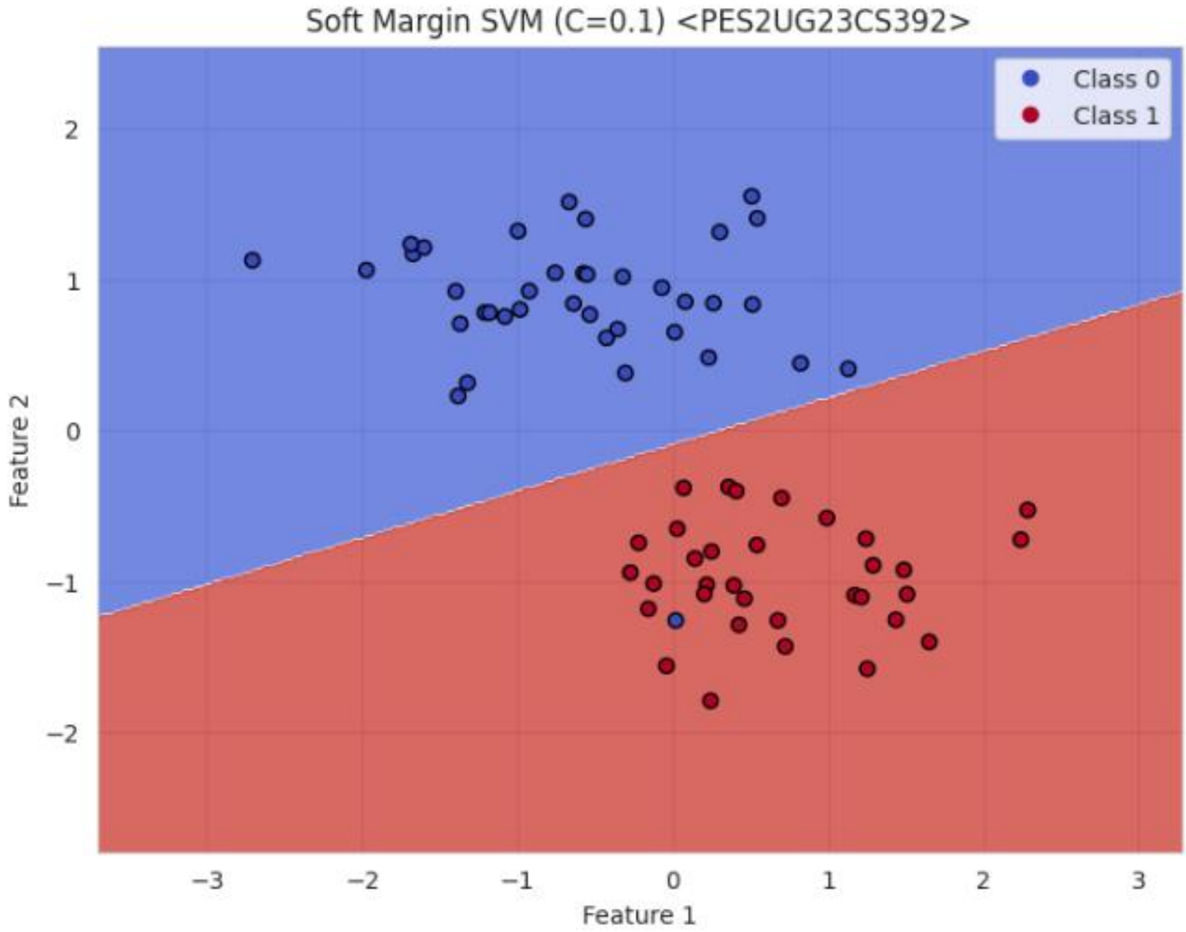


Banknote Dataset - SVM with POLY Kernel

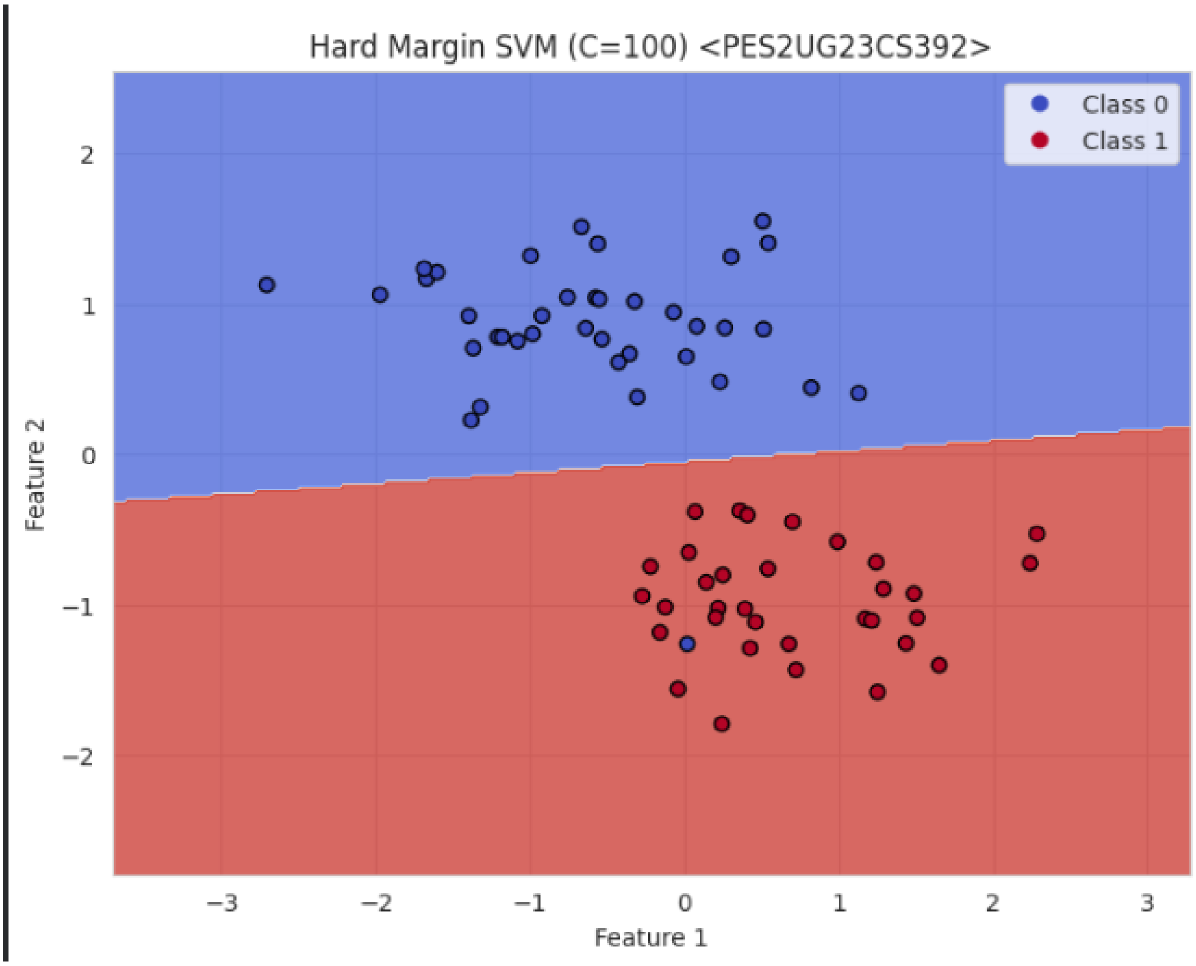


Margin Analysis (2 plots):

Soft Margin SVM (C=0.1)



Hard Margin SVM (C=100)



Hard Margin SVM (C=100) <PES2UG23CS392>
Class 0
Class 1
Feature 1
Feature 2