ML LAB WEEK - 10

NAME : CHETAN NADICHAGI

SRN : PES2UG23CS149

SEC : C

MOONS DATASET :

Classification Reports :

1) **LINEAR SVM :**

```
SVM with LINEAR Kernel PES2UG23CS149
              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) **RBF SVM :**

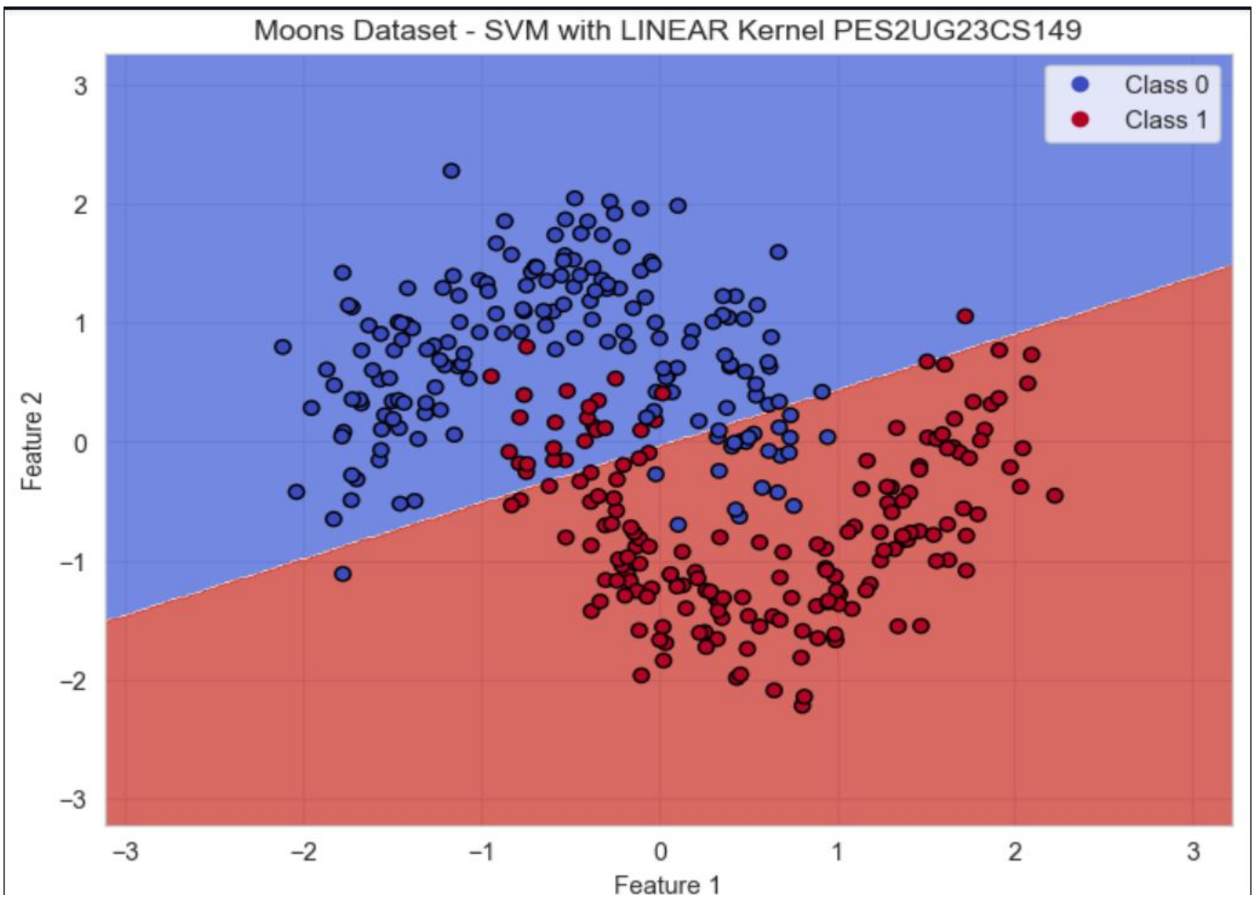
```
SVM with RBF Kernel PES2UG23CS149
              precision    recall  f1-score   support

           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) **POLY SVM :**

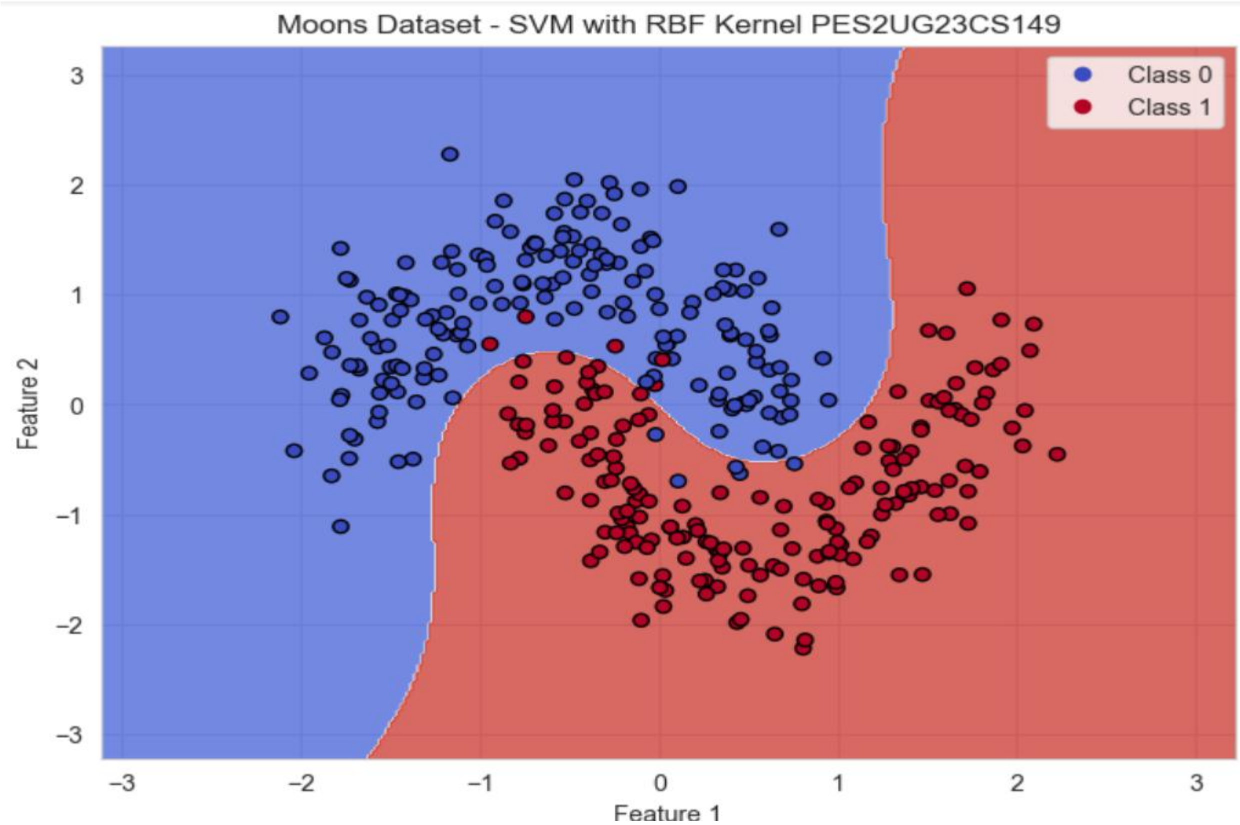
```
SVM with POLY Kernel PES2UG23CS149
...
weighted avg       0.89      0.89      0.89       150

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

Decision Boundaries :

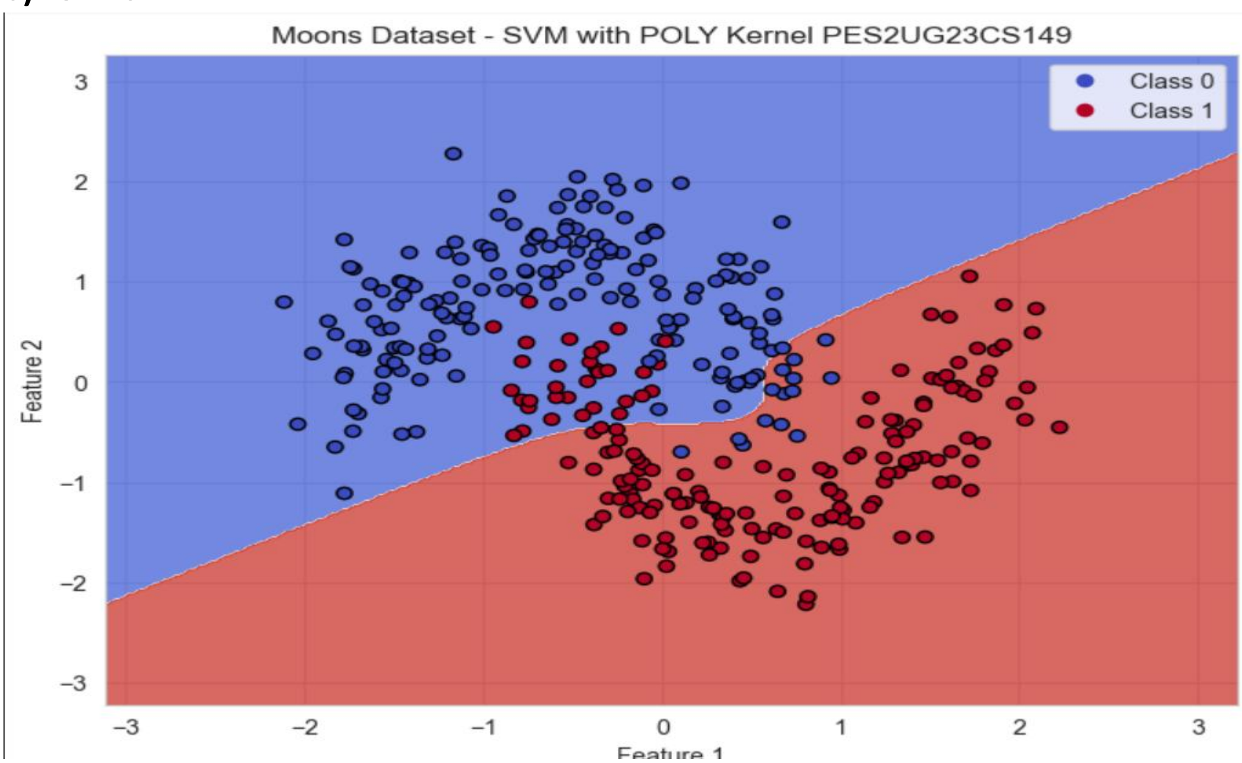
4)LINEAR SVM :



5)RBF SVM :



6)POLY SVM :



BANKNOTE DATASET :

Classification reports :

1) **LINEAR SVM :**

```
SVM with LINEAR Kernel PES2UG23CS149
              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) **RBF SVM :**

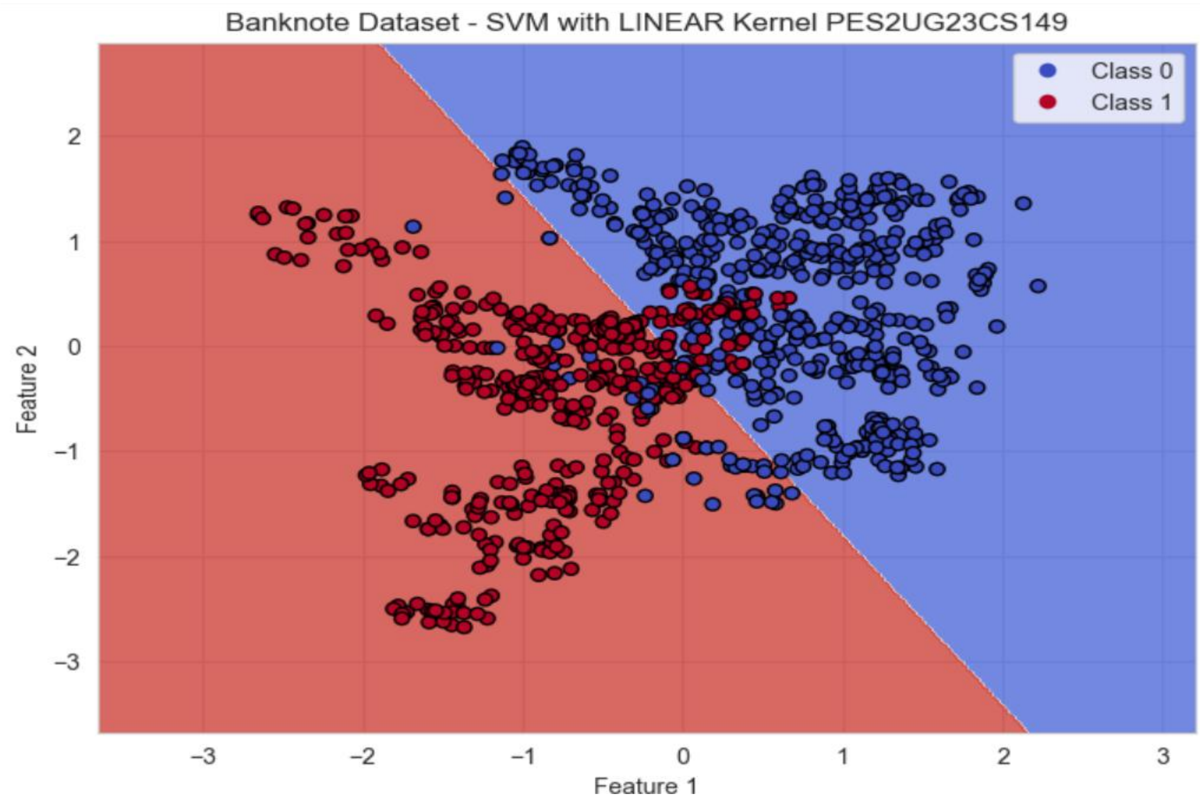
```
SVM with RBF Kernel PES2UG23CS149
              precision    recall  f1-score   support

      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
```

3) **POLY SVM :**

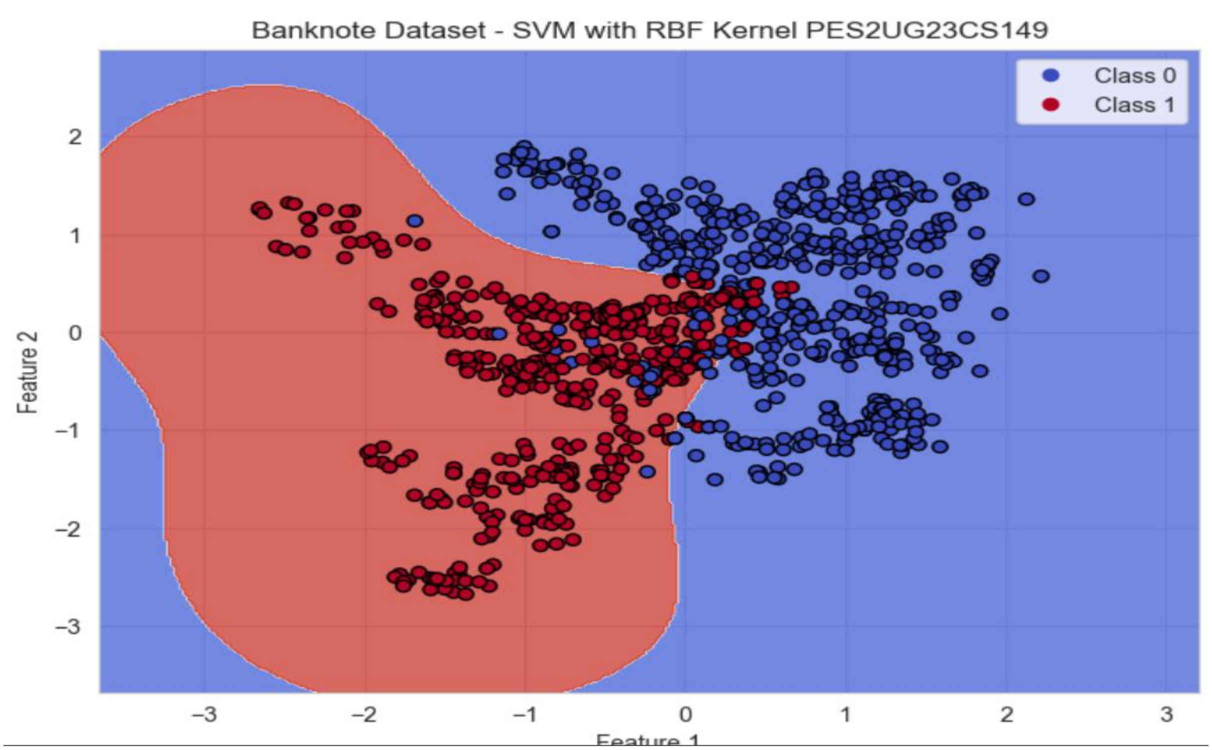
```
SVM with POLY Kernel PES2UG23CS149
...
weighted avg       0.85      0.84      0.84       412


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

Decision Boundaries :

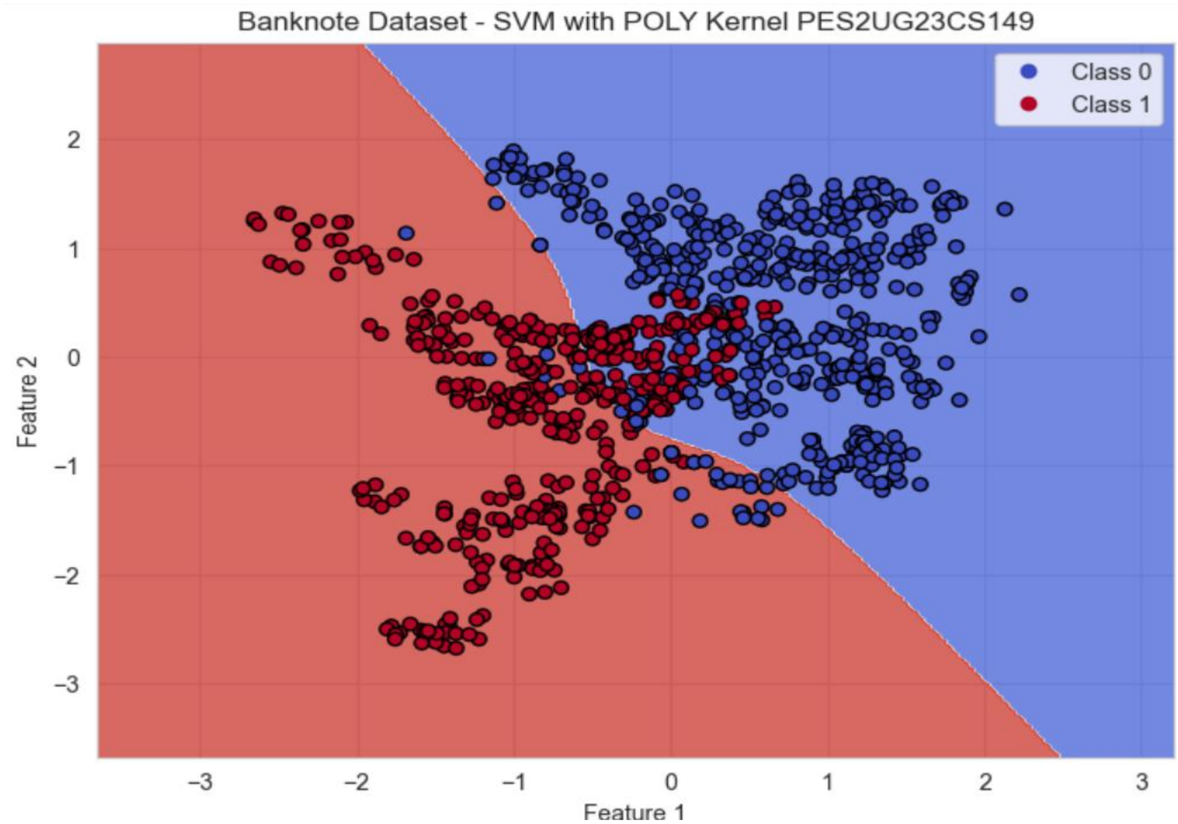
4) **LINEAR SVM :**



5) **RBF SVM :**

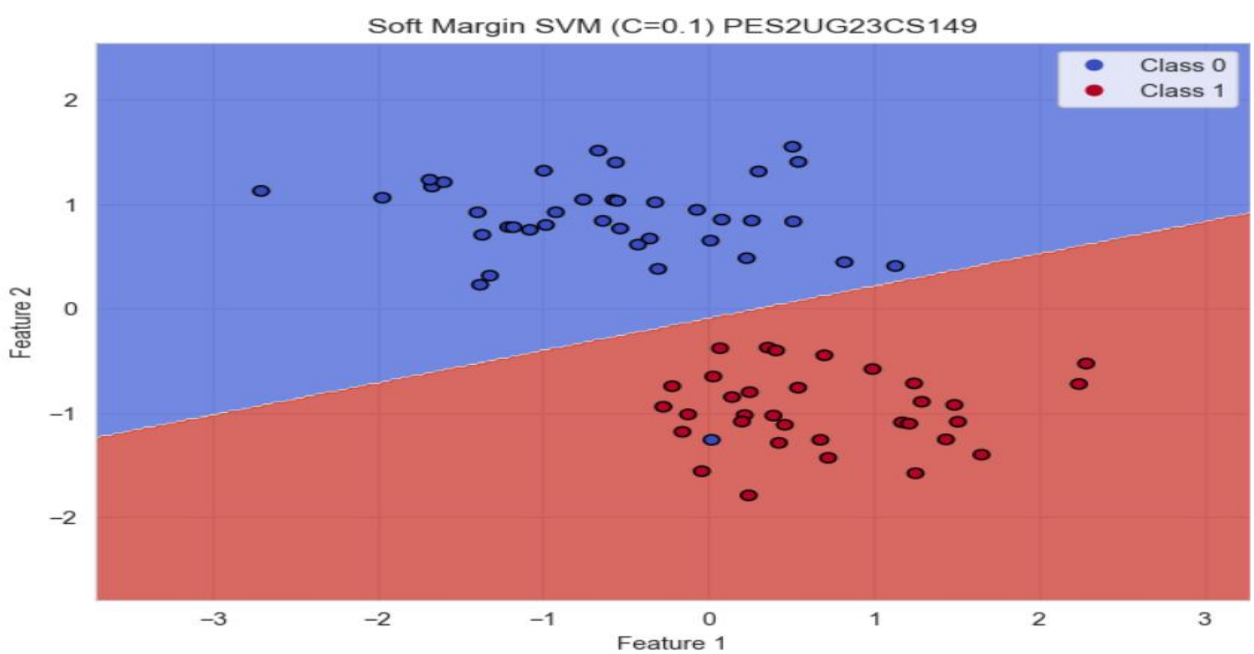


6) POLY SVM :

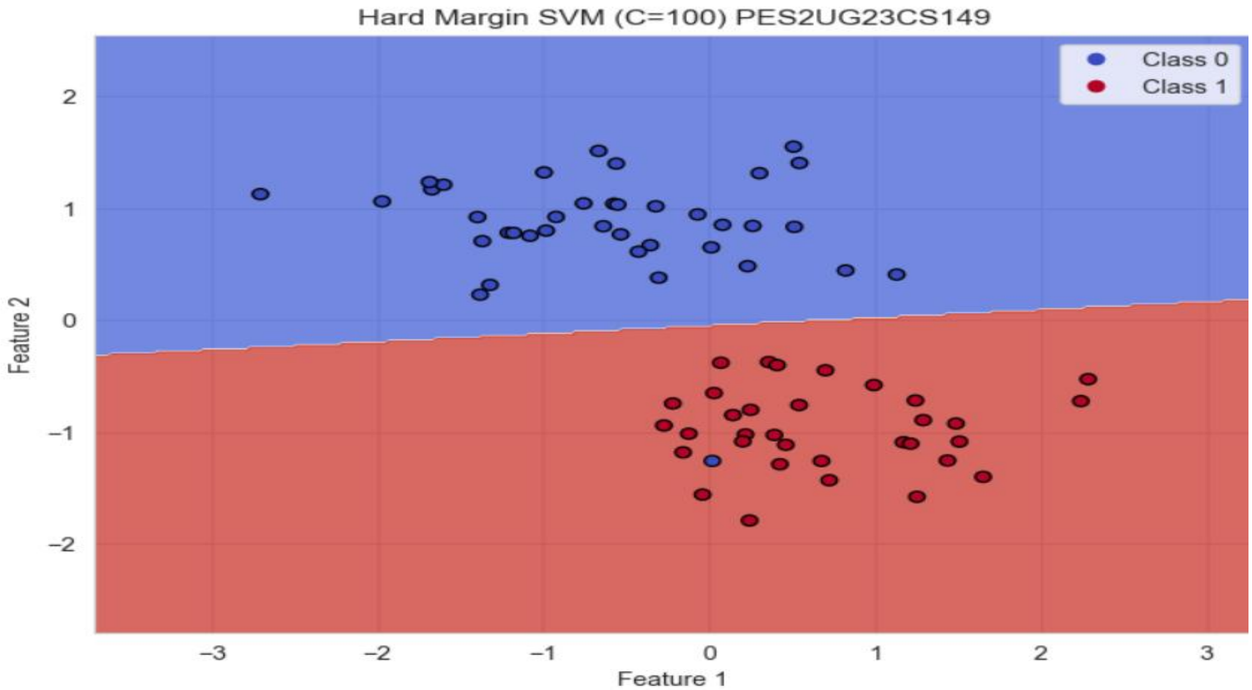


Margin Analysis :

Soft Margin (C=0.1)



Hard Margin SVM (C=100)



Analysis Answers

Moonset Dataset :

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

The Linear Kernel performs poorly on the Moons dataset because it cannot model the curved boundary needed to separate the two classes. This confirms that the data is non-linearly separable, and linear decision surfaces are inadequate for such patterns.

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 Kernel captures the natural shape of the Moons data more effectively and generalizes better. The Polynomial Kernel can model non-linearity but is more sensitive to parameter tuning (degree, coefficient), leading to either underfitting or overfitting.

Banknote Dataset :

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

The Linear Kernel is the most effective for the Banknote dataset.
Since the data is nearly linearly separable, complex transformations (like RBF or Polynomial) add no advantage and may even slightly degrade performance due to overfitting or extra complexity.

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

The Polynomial kernel performs worse here because the Banknote dataset doesn’t require non-linear transformations. Its added complexity overfits the data, while the simpler Linear kernel captures the true separation effectively.

Hard vs Soft Margin Analysis :

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

C controls the penalty for misclassification/slack. A small C makes slack cheap, so the optimizer prioritizes a larger margin even if some training points are inside or misclassified. A large C forces the model to avoid errors, which typically yields a narrower margin that tightly hugs the training points.

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?

SVM with slack (soft margin) allows such "mistakes" because it trades off between margin width and training error.

- The optimization objective is: maximize margin (minimize ||w||) while penalizing misclassifications/slack (weighted by C).
- With a low C, the penalty for slack is small, so the solver accepts some points inside/on the wrong side to achieve a larger margin.

Primary goal**:** find a decision boundary that generalizes well to unseen data — not to perfectly label every training point. The model prefers a simpler (wider-margin) separator if that improves expected performance on new data (structural risk minimization).

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 (C = 100) is more likely to overfit**.**
Why:

- High C strongly penalizes training errors, so the SVM will try to classify every training point correctly, even at the cost of a very narrow, complex margin that follows idiosyncrasies (noise) in the training set.
- That tight fit reduces margin-based regularization and increases sensitivity to noise/outliers → worse generalization.

The soft margin (low C) is more robust because it tolerates some training error to keep the margin wide.

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?

   For a new, unseen point (especially when data is noisy**),** trust the Soft Margin (low **C**) more. It's less likely to have fit noise in the training set and typically generalizes better.

In practice**:** start with a low-to-moderate C (softer margin) and use cross-validation to tune C. This gives a good bias–variance starting point: prevents overfitting while still allowing the model to learn structure.

- If CV shows underfitting, gradually increase C.
- If CV shows overfitting, decrease C.

THANK YOU