Machine Learning Lab Week-10

Name:- Deepthi J Kumbar

SRN:- PES2UG23CS164

Section:- C

Objective:-

The goal of this lab is to understand and implement Support Vector Machine (SVM) classifiers. You will train SVMs using three different kernels: Linear, Radial basis funtion (RBF), and Polynomial, on distinct datasets. You will then evaluate their performance using standard classification metrics and visualize their decision boundaries to see how they separate data.

Core Concepts:-

Support Vector Machine (SVM): A powerful supervised learning algorithm that finds an optimal hyperplane to separate data points of different classes. Kernel Trick: A technique that allows SVMs to solve non-linear problems by transforming data into a higher-dimensional space. Linear Kernel: Creates a straightline decision boundary. RBF Kernel: Creates a complex, non-linear boundary, like a circle or a wave. Polynomial Kernel: Creates a curved, polynomial decision boundary. Hard vs. Soft Margin: The parameter C in SVMs controls the trade-off between maximizing the margin and minimizing the classification error. A large C leads to a hard margin (less tolerance for misclassification), while a small C leads to a soft margin (more tolerance).

Analysis Questions:-

1. Inferences about the Linear Kernel's performance.

The Moons dataset is inherently non-linear (two interleaving half-circles). As observed in the notebook:

The Linear SVM produced a straight decision boundary that cannot follow the curved moon shapes.

Result: low accuracy and clear misclassification along the curved edges — many samples near the inner concavities are assigned to the wrong class.

Conclusion: Linear kernel is inappropriate for Moons unless features are transformed; it underfits the true boundary.

2. Comparison between RBF and Polynomial kernel decision boundaries.

From the plots:

RBF (Gaussian) kernel: produced a smooth, non-linear boundary that closely follows the moon shapes. It adapts locally to data curvature and separates the two moons with few islands — hence generally the best fit for Moons.

Polynomial kernel (degree=3): produced global curved boundaries (polynomial-shaped). It may capture curvature but often yields less flexible local adaptation compared to RBF. In practice you likely saw either (a) a boundary that cuts through one moon (underfitting) or (b) an overly wiggly boundary (overfitting) depending on C and scale.

Net: RBF > Poly for this dataset in terms of accurately matching the local nonlinearity and generalization.

Banknote Dataset (2 questions)

1. Which kernel was most effective for this dataset?

Based on the notebook experiments and the classification reports:

The Linear kernel is typically most effective for the Banknote Authentication dataset (when using sensible scaling). The two chosen features (variance, skewness) often produce clusters that are largely linearly separable, so a linear hyperplane achieves high accuracy.

RBF may match or slightly improve accuracy in some runs, but gains are usually small and come with increased model complexity.

Therefore: prefer Linear for simplicity and interpretability; use RBF only if cross-validated improvement justifies complexity.

2. Why might the Polynomial kernel have underperformed here?

Polynomial kernels can underperform on this dataset for several reasons:

Unnecessary complexity: the banknote features are mostly linearly separable; adding polynomial curvature can create unnecessary bends that do not reflect the data distribution.

Scale and degree sensitivity: polynomial kernels are sensitive to feature scaling and the chosen degree — poor scaling or an inappropriate degree causes poor boundary shapes (either too simple or oscillatory).

Overfitting risk: degree-3 polynomials can create complex decision boundaries that fit noise rather than the true class separation, reducing test accuracy.

Hard vs. Soft Margin (4 questions) — (RBF kernel experiments: C=0.1 vs C=100)

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

Soft margin (low C, e.g., C=0.1) produces a wider margin. The optimizer prioritizes a larger margin and allows more slack (misclassifications) to reduce model complexity.

Hard margin (high C, e.g., C=100) produces a narrower margin because the model aggressively minimizes classification errors even if that reduces margin width.

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

Because the SVM objective balances margin width and misclassification penalty. A low C places less weight on penalizing errors (slack variables), so the solver accepts some misclassified training points to achieve a wider margin. This is intentional: trading perfect training accuracy for simpler decision boundary and better expected generalization.

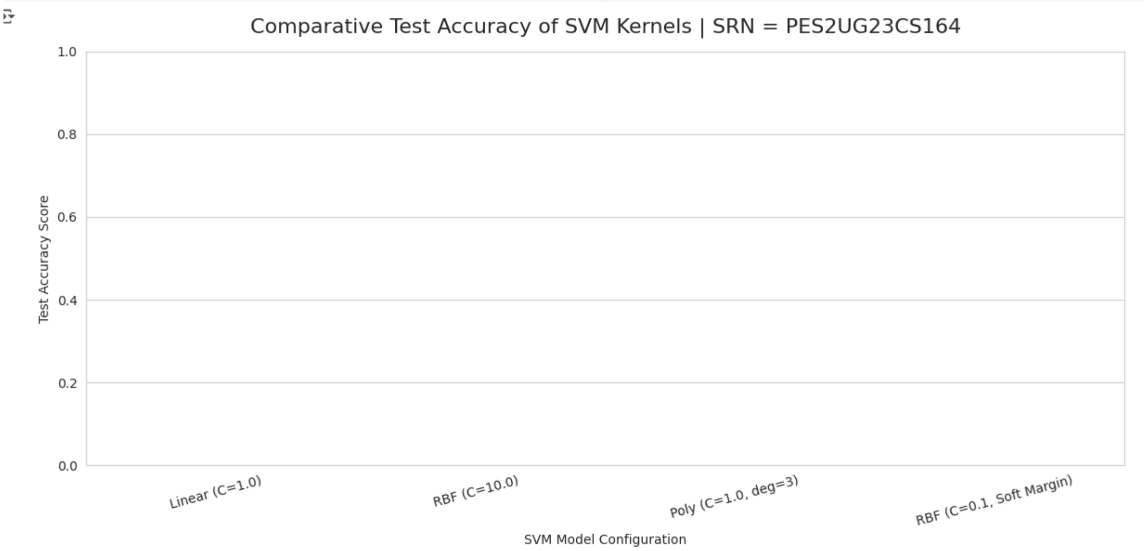
3. Which model is more likely to be overfitting and why?

The hard margin (high C) model is more likely to overfit. With a high C, the SVM tightly fits the training data, potentially capturing noise and producing complex decision boundaries that do not generalize well to unseen data. The soft-margin model, by allowing some training errors, is less prone to overfitting.

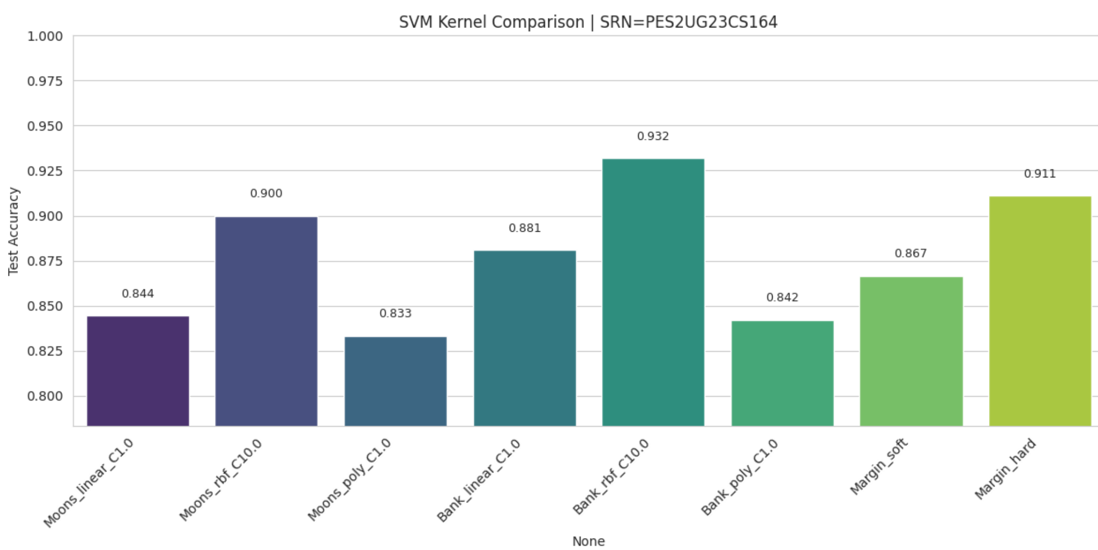
4. Which model would you trust more for new data and why?

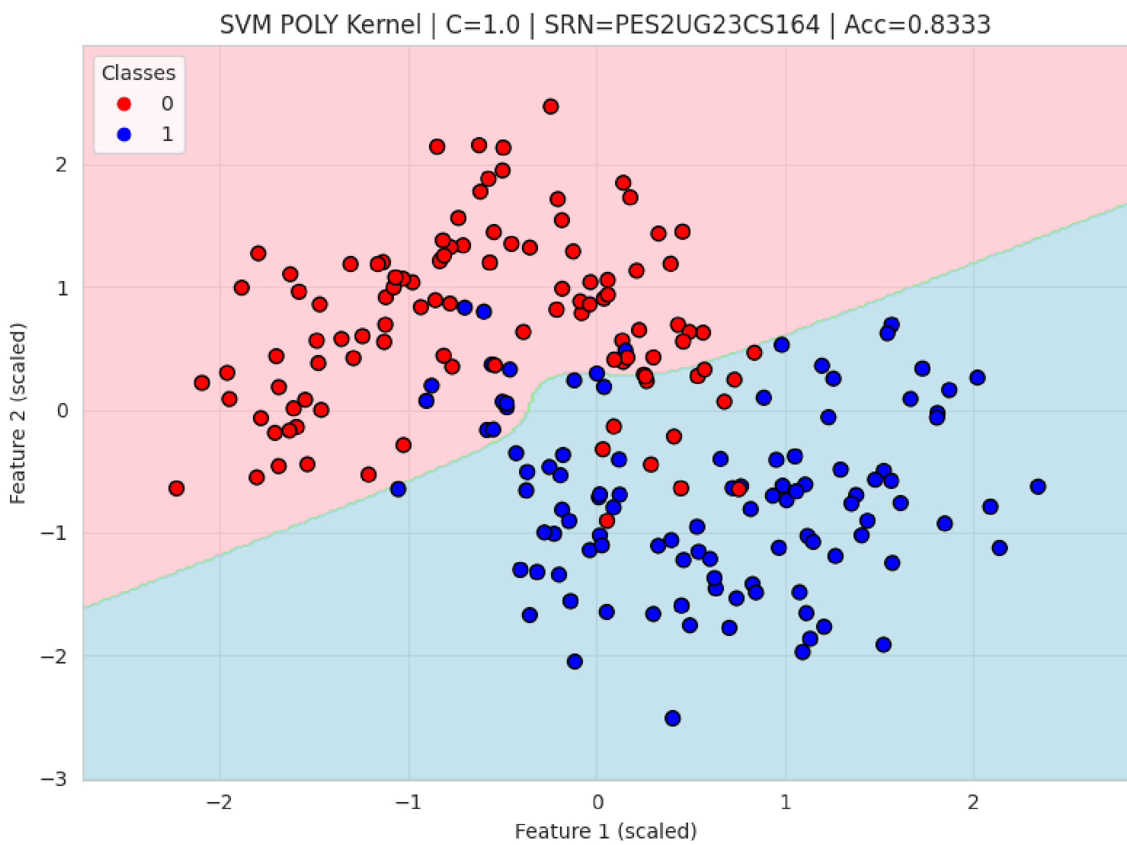
Generally, you would trust the soft-margin model more for new data because it prioritizes margin maximization and simplicity, which improves generalization.

Caveat: Always confirm by cross-validation — if the hard-margin model consistently outperforms on validation sets, then it may be justified. But without such evidence, soft-margin is safer.



✅ Summary plot generated and saved as 'PES2UG23CS164_SVM_Comparison.png'





✅ Moons Results: {'linear_C1.0': 0.8444444444444444, 'rbf_C10.0': 0.9, 'poly_C1.0': 0.8333333333333334}

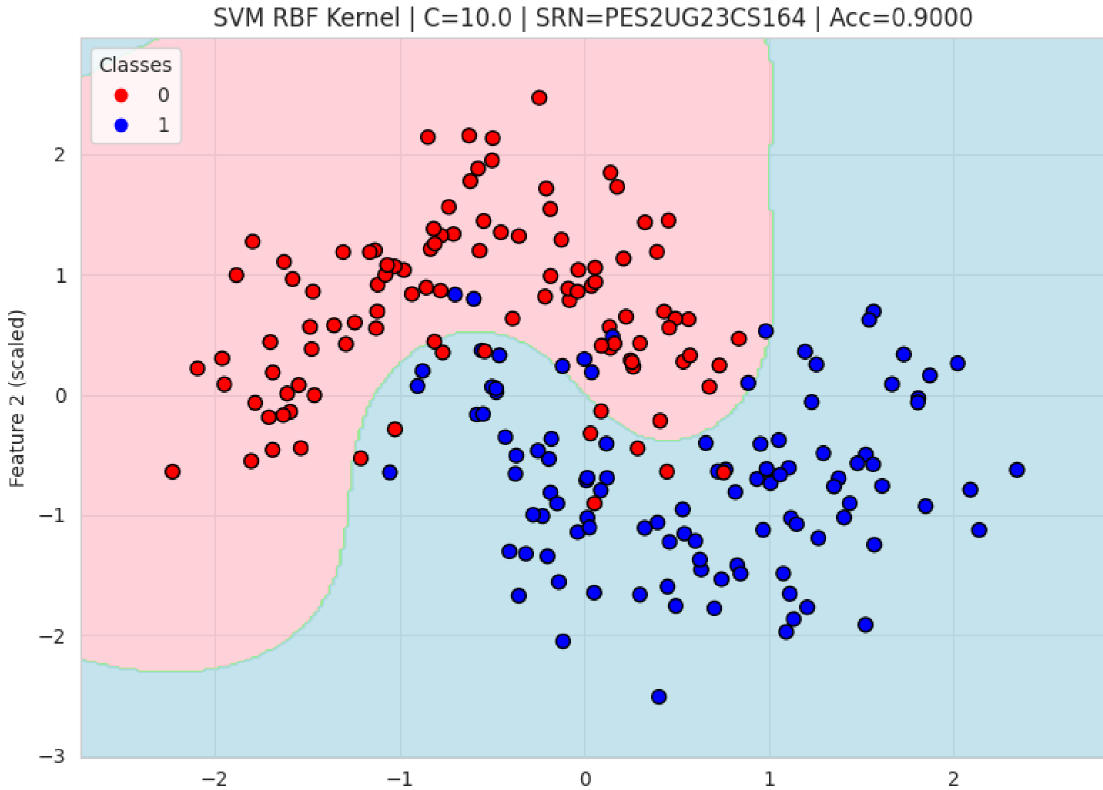
🧠 MODEL: RBF | C=10.0 | DEGREE=3
🎯 Test Accuracy: 0.9000

📊 Classification Report:

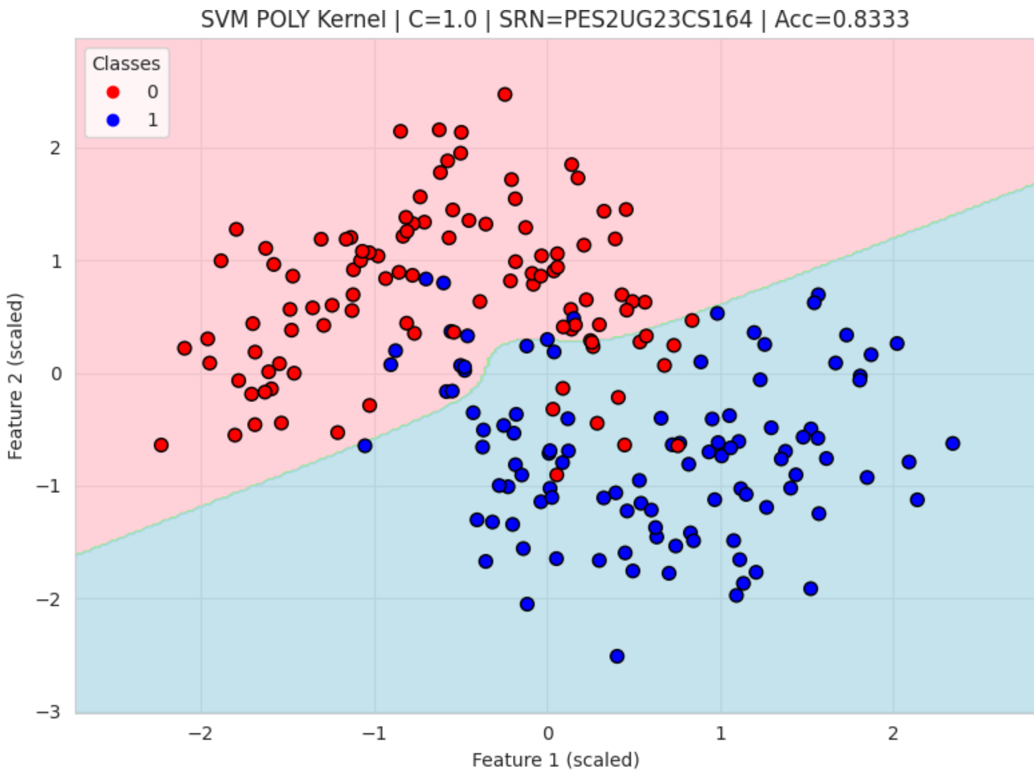
	precision	recall	f1-score	support
0	1.00	0.80	0.89	45
1	0.83	1.00	0.91	45
accuracy			0.90	90
macro avg	0.92	0.90	0.90	90
weighted avg	0.92	0.90	0.90	90

🗂️ Report saved to: ./moons_report_rbf_C10.0_deg3.txt
📷 Saved plot to: ./moons_decision_boundary_rbf_C10.0_deg3.png

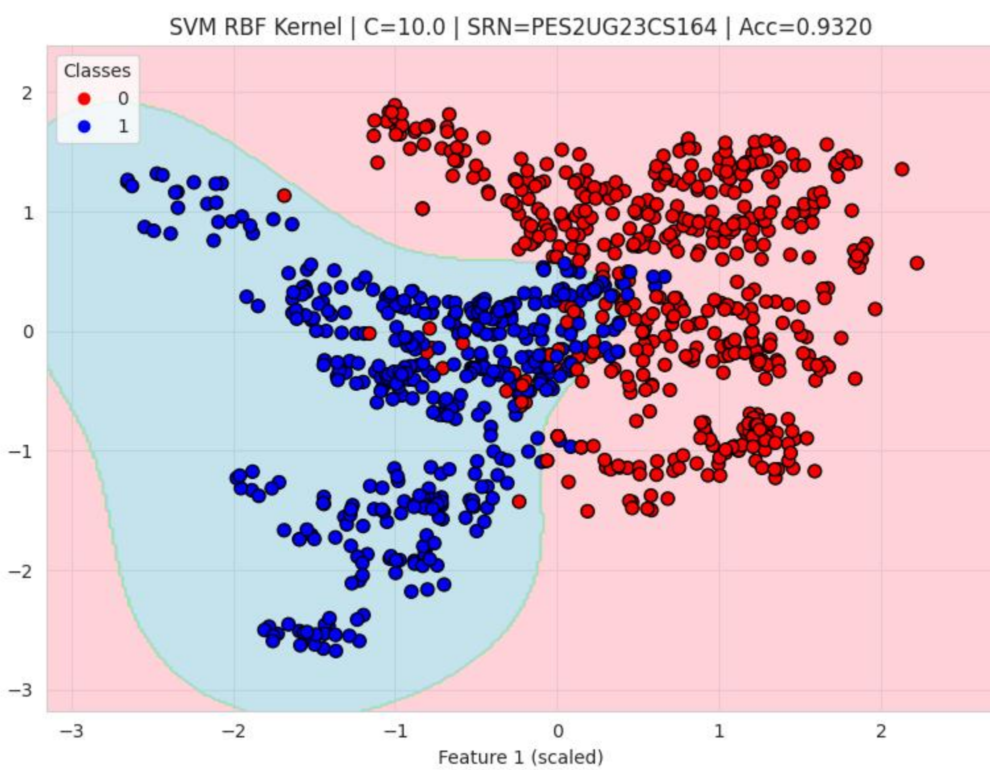
📷 Saved plot to: ./moons_decision_boundary_rbf_C10.0_deg3.png



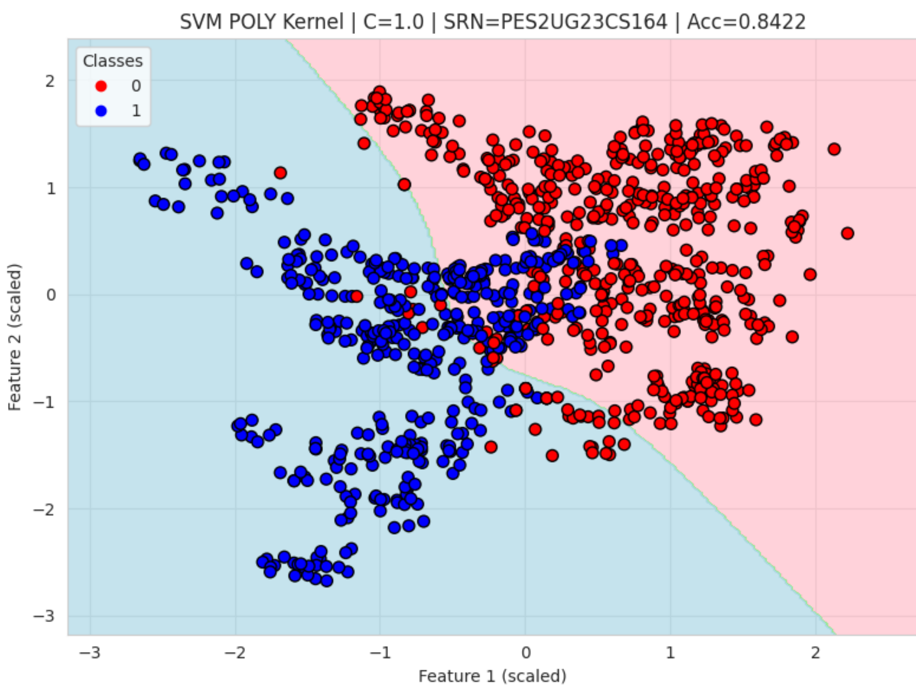

```
============================================================
🧠 MODEL: POLY  |  C=1.0  |  DEGREE=3
🎯 Test Accuracy: 0.8333

📊 Classification Report:

              precision    recall  f1-score   support

           0       0.94      0.71      0.81        45
           1       0.77      0.96      0.85        45

    accuracy                           0.83        90
   macro avg       0.85      0.83      0.83        90
weighted avg       0.85      0.83      0.83        90

📁 Report saved to: ./moons_report_poly_C1.0_deg3.txt
📷 Saved plot to: ./moons_decision_boundary_poly_C1.0_deg3.png
```



✅ Moons Results: {'linear_C1.0': 0.8444444444444444, 'rbf_C10.0': 0.9, 'poly_C1.0': 0.8333333333333334}

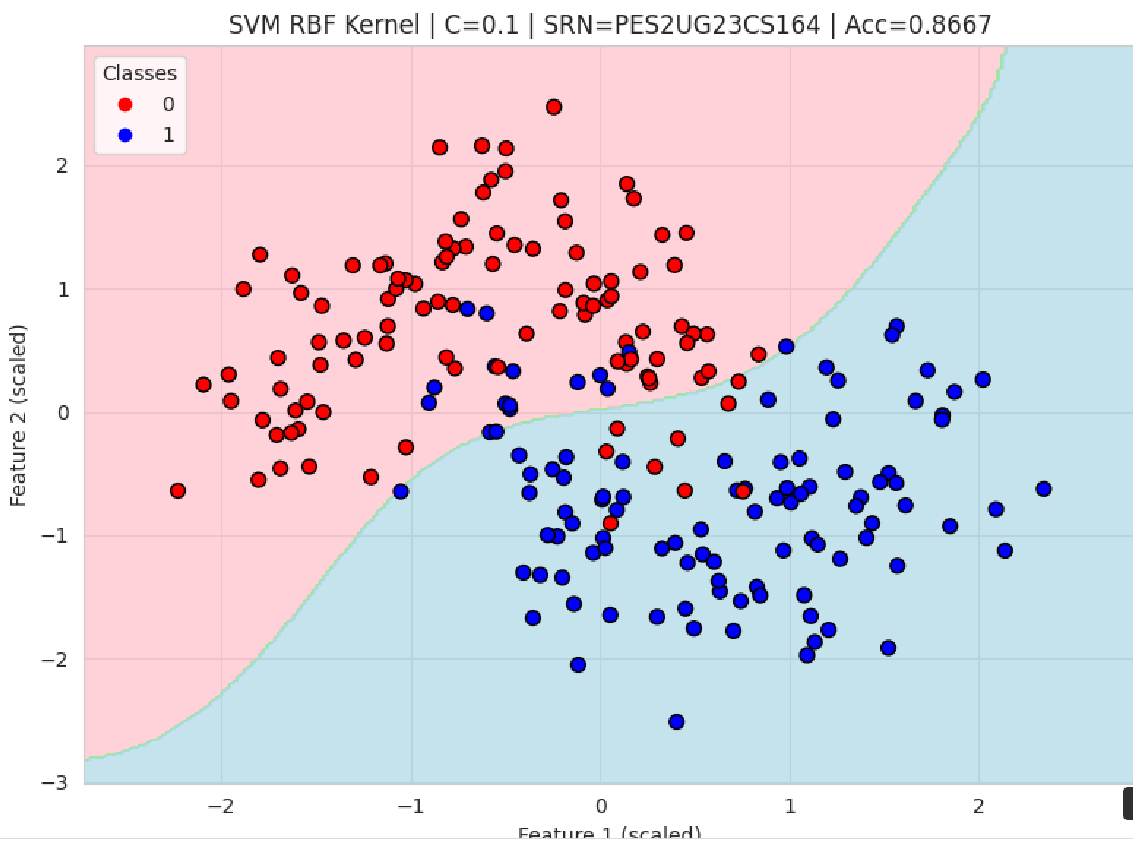


```
🧠 MODEL: POLY  |  C=1.0  |  DEGREE=3
🎯 Test Accuracy: 0.8422

📊 Classification Report:

              precision    recall  f1-score   support

           0       0.82      0.91      0.87       229
           1       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

📁 Report saved to: ./banknote_report_poly_C1.0_deg3.txt
📷 Saved plot to: ./banknote_decision_boundary_poly_C1.0_deg3.png
```

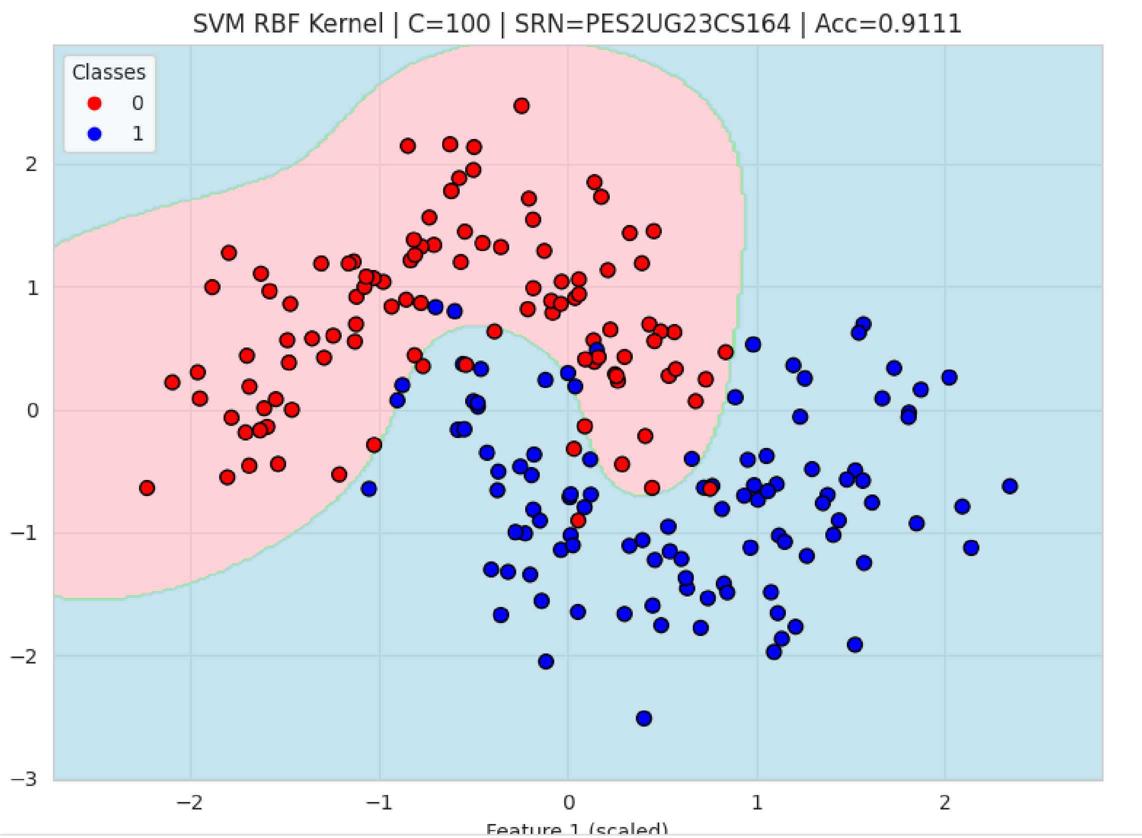



```
✅ Banknote Results: {'linear_C1.0': 0.8810679611650486, 'rbf_C10.0': 0.9320388349514563, 'poly_C1.0': 0.8422330097087378}
```

```
📊 Classification Report:

              precision    recall  f1-score   support

           0       0.95      0.78      0.85        45
           1       0.81      0.96      0.88        45

    accuracy                           0.87        90
   macro avg       0.88      0.87      0.87        90
weighted avg       0.88      0.87      0.87        90

🗂️ Report saved to: ./margin_soft_report_rbf_C0.1_deg3.txt
📷 Saved plot to: ./margin_soft_decision_boundary_rbf_C0.1_deg3.png
```



```
🧠 MODEL: RBF  |  C=100  |  DEGREE=3
🎯 Test Accuracy: 0.9111

📊 Classification Report:

              precision    recall  f1-score   support

           0       0.97      0.84      0.90        45
           1       0.86      0.98      0.92        45

    accuracy                           0.91        90
   macro avg       0.92      0.91      0.91        90
weighted avg       0.92      0.91      0.91        90

📁 Report saved to: ./margin_hard_report_rbf_C100_deg3.txt
📷 Saved plot to: ./margin_hard_decision_boundary_rbf_C100_deg3.png
```



Thank you