Support Vector Machine (SVM) Lab Report

Name : C Bhargav

SRN : PES2UG23CS137

Section : C

Objectives

1. To implement SVM for classification.
2. To analyze results on the Iris dataset.

Results

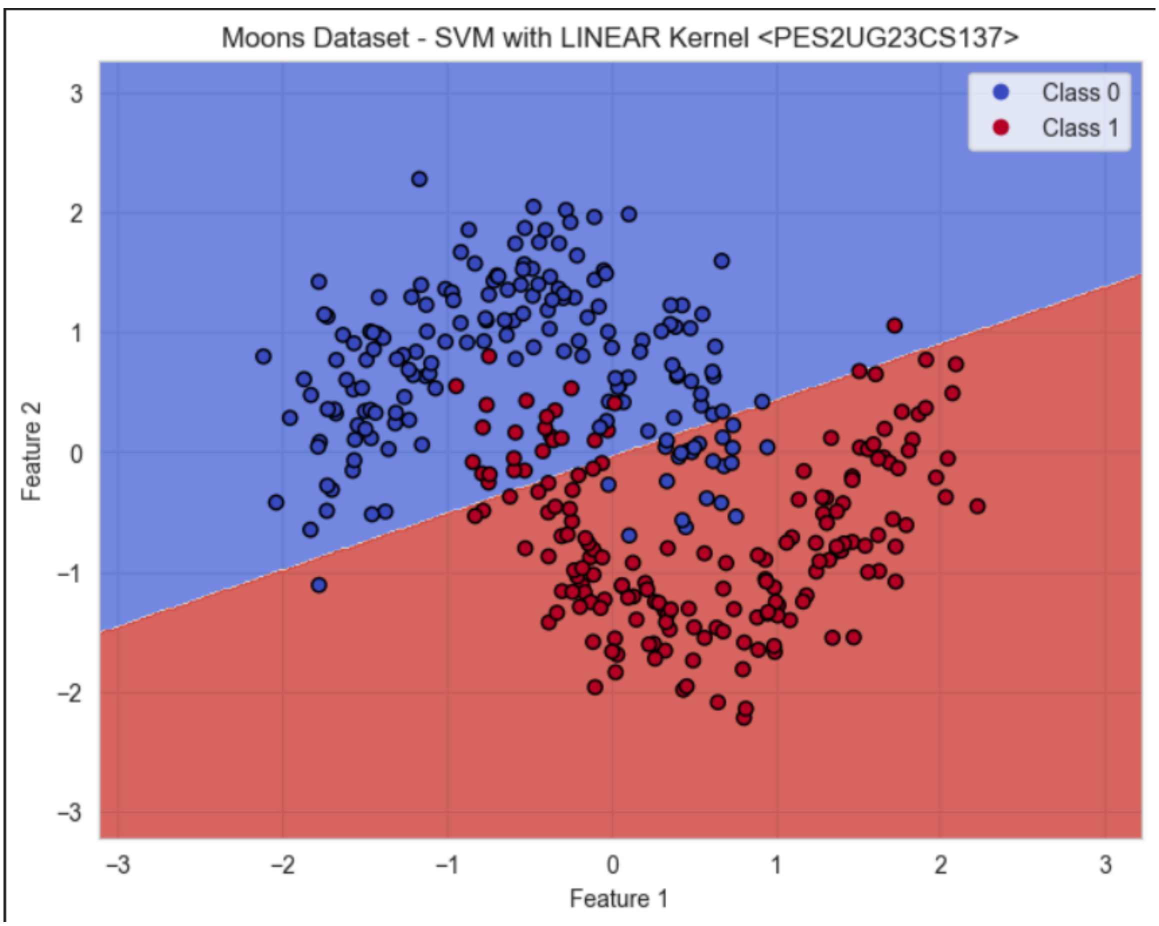
Paste your SVM results here (accuracy, confusion matrix, classification report, etc.).

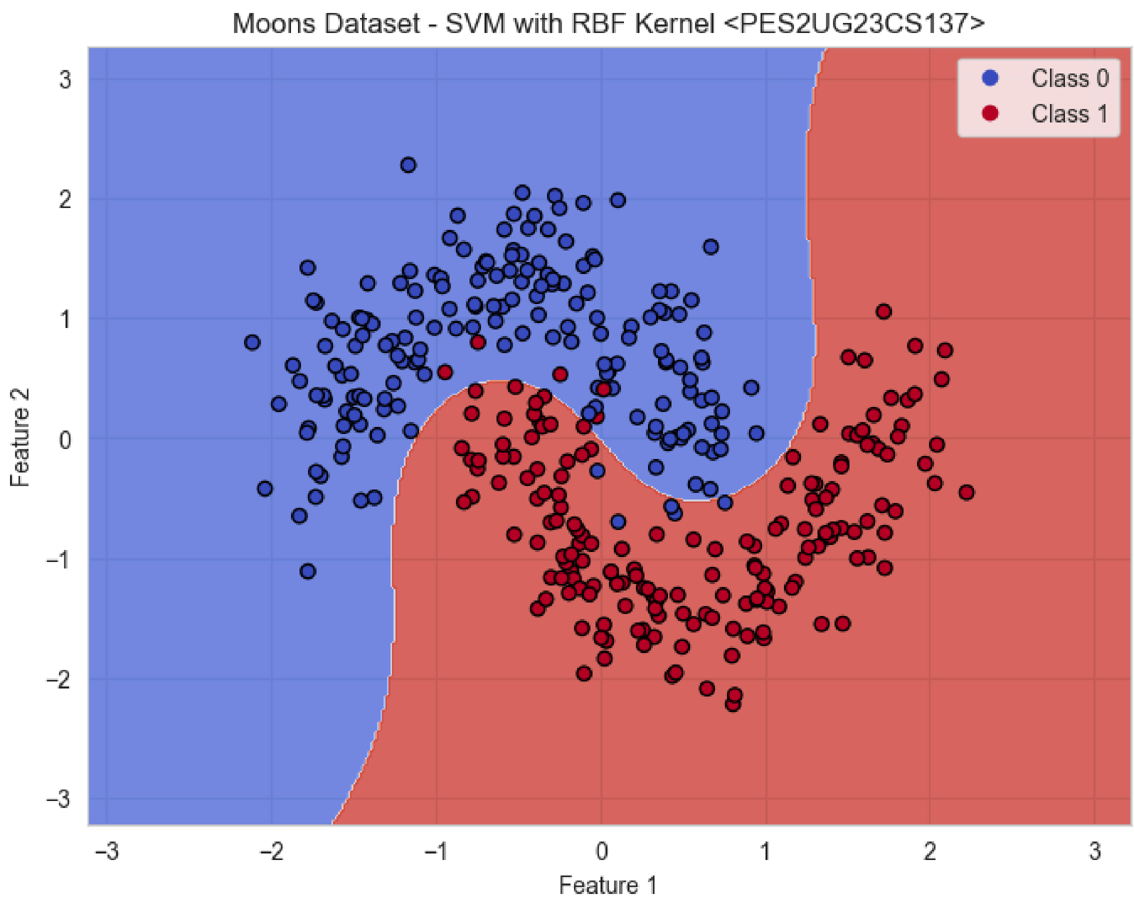
SVMs on Moons

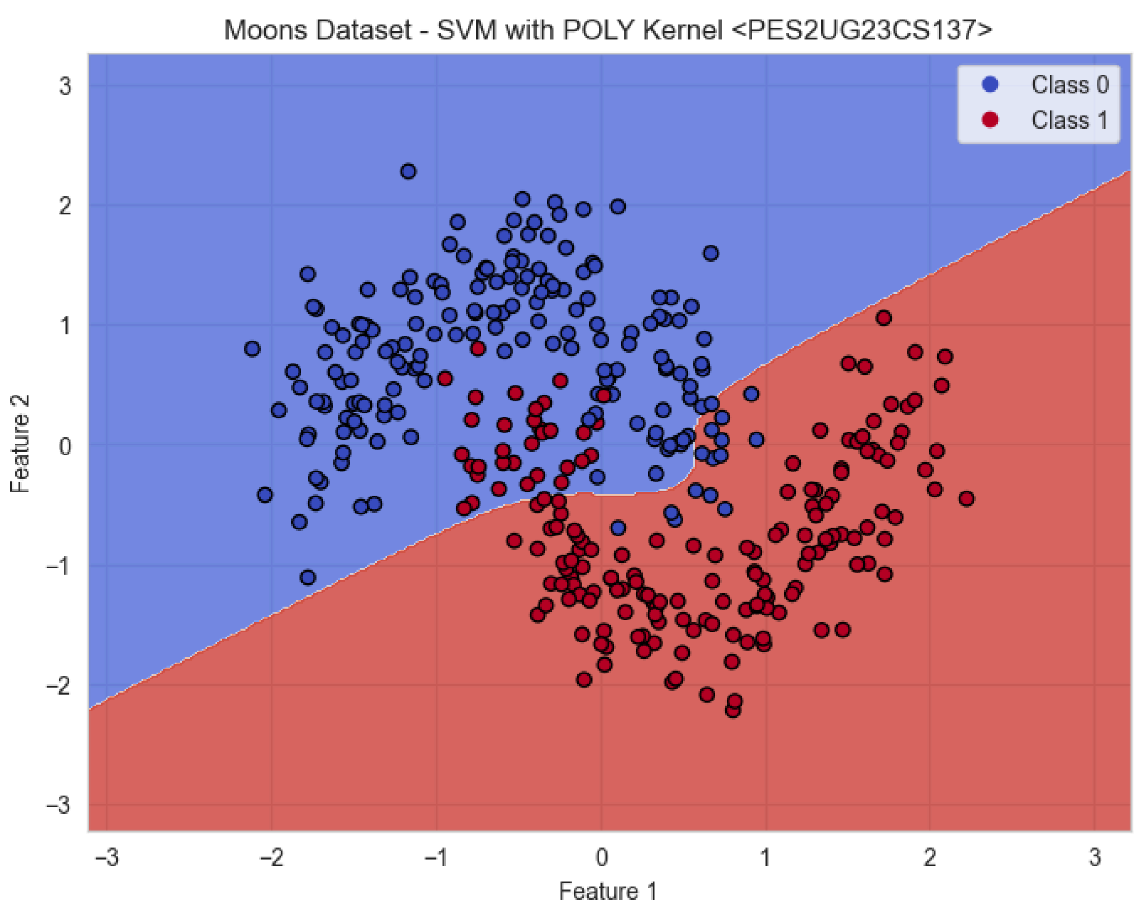
```
SVM with LINEAR Kernel <PES2UG23CS137>
              precision    recall  f1-score   support

           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

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

```
SVM with RBF Kernel <PES2UG23CS137>
              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
```

```
SVM with POLY Kernel <PES2UG23CS137>
              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
```




Moons Dataset - SVM with RBF Kernel <PES2UG23CS137>
Class 0
Class 1
Feature 1
Feature 2
−3
−2
−1
0
1
2
3


Moons Dataset - SVM with POLY Kernel <PES2UG23CS137>
Class 0
Class 1
Feature 1
Feature 2
−3
−2
−1
0
1
2
3

SVMs on Banknote data

SVM with LINEAR Kernel <PES2UG23CS137>

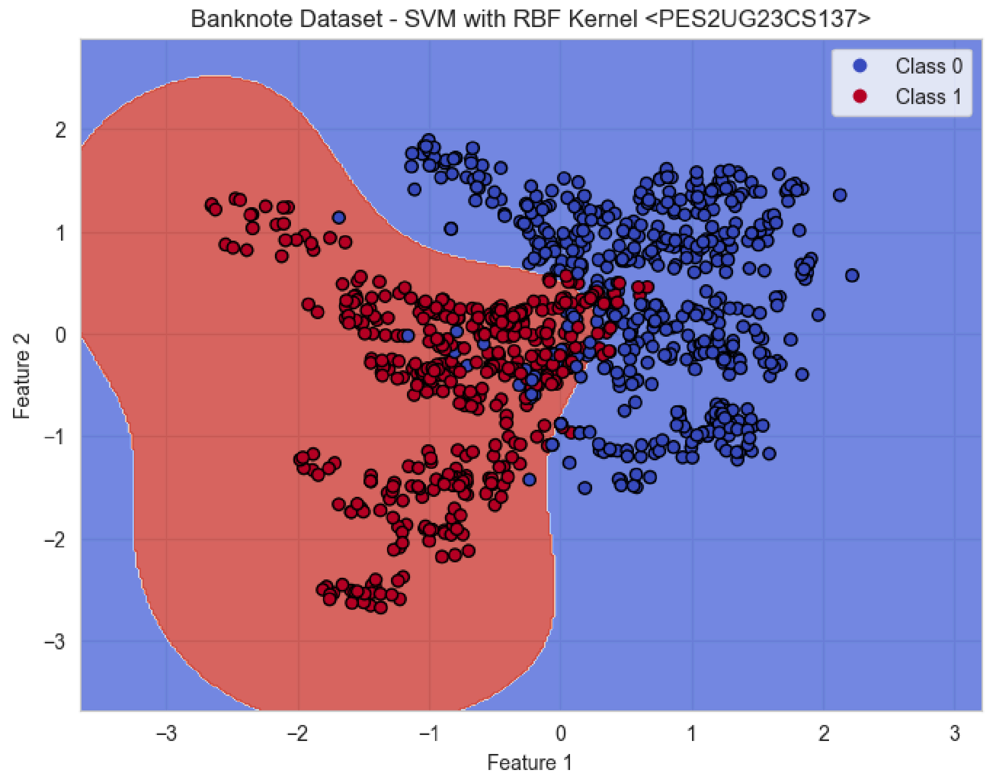
	precision	recall	f1-score	support
Forged	0.90	0.88	0.89	229
Genuine	0.86	0.88	0.87	183
accuracy			0.88	412
macro avg	0.88	0.88	0.88	412
weighted avg	0.88	0.88	0.88	412

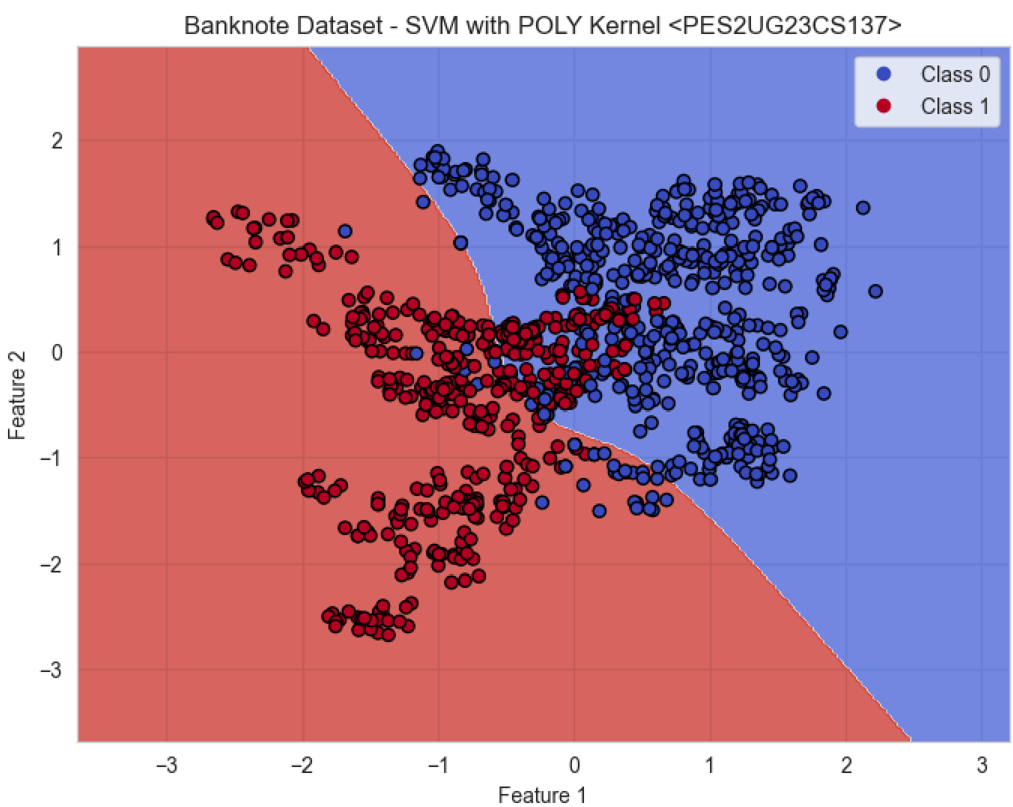
SVM with RBF Kernel <PES2UG23CS137>

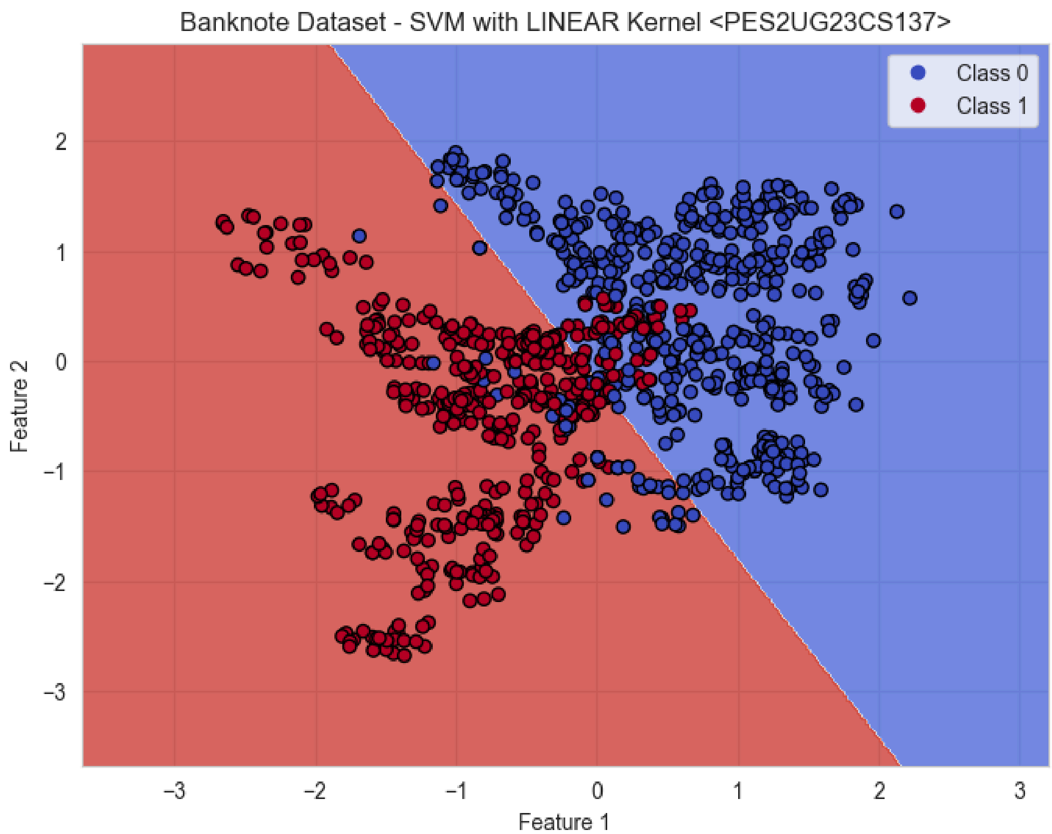
	precision	recall	f1-score	support
Forged	0.96	0.91	0.94	229
Genuine	0.90	0.96	0.93	183
accuracy			0.93	412
macro avg	0.93	0.93	0.93	412
weighted avg	0.93	0.93	0.93	412

SVM with POLY Kernel <PES2UG23CS137>

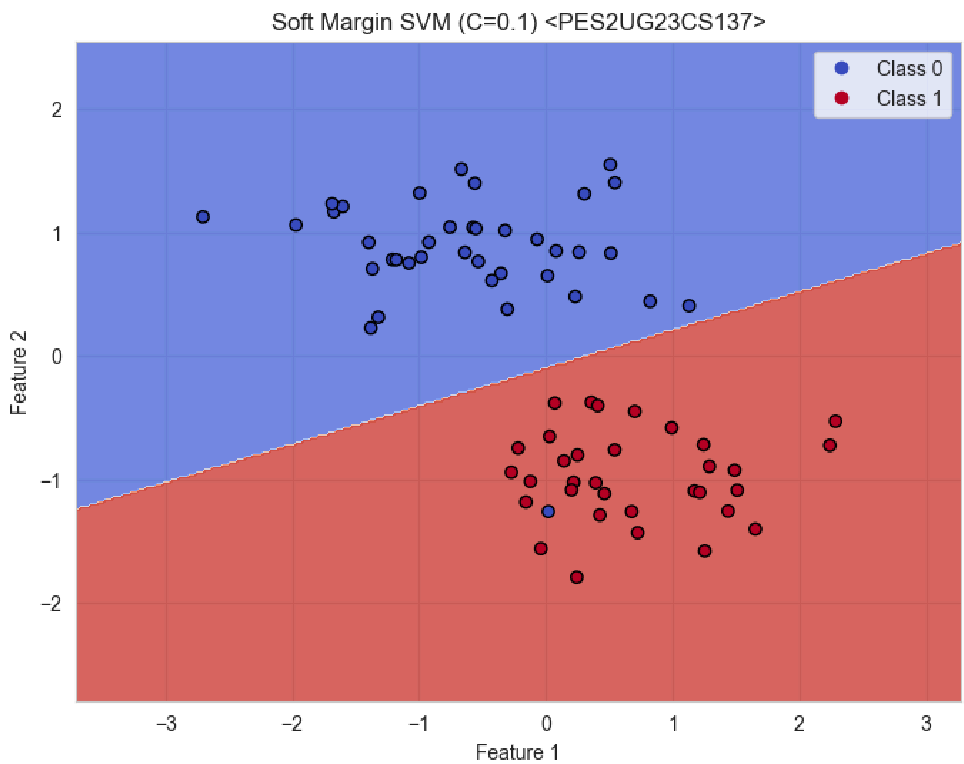
	precision	recall	f1-score	support
Forged	0.82	0.91	0.87	229
Genuine	0.87	0.75	0.81	183
accuracy			0.84	412
macro avg	0.85	0.83	0.84	412
weighted avg	0.85	0.84	0.84	412


Banknote Dataset - SVM with RBF Kernel <PES2UG23CS137>
Class 0
Class 1
Feature 2
Feature 1

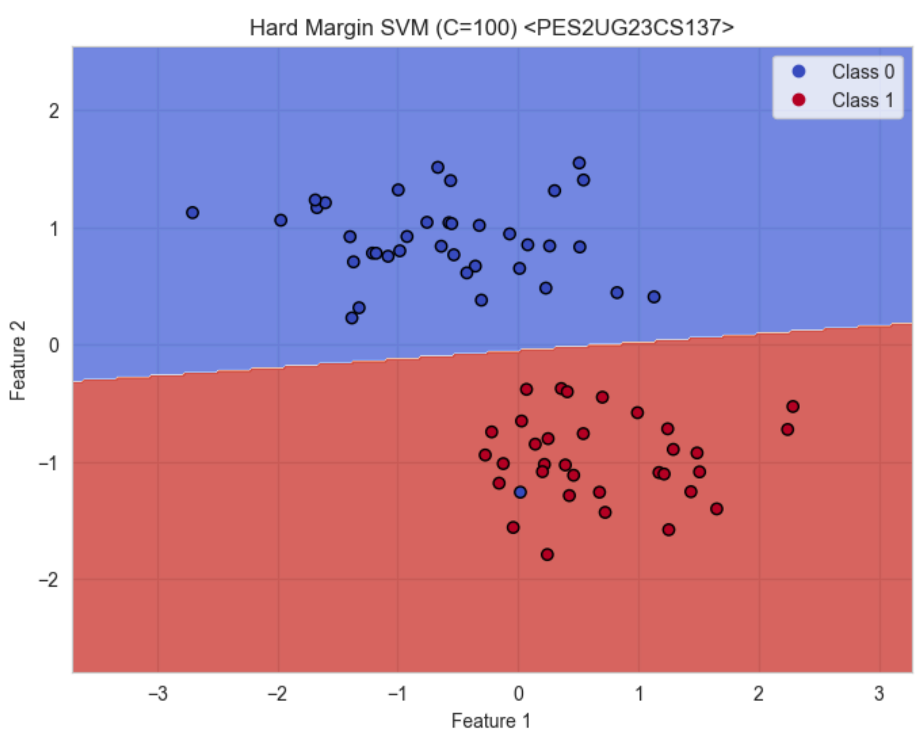

Banknote Dataset - SVM with POLY Kernel <PES2UG23CS137>
Class 0
Class 1
Feature 2
Feature 1



Plot Interpretation — Soft vs Hard Margin (Linear SVM)



Soft Margin SVM (C=0.1) <PES2UG23CS137>
Class 0
Class 1
Feature 2
Feature 1



Hard Margin SVM (C=100) <PES2UG23CS137>
Class 0
Class 1
Feature 2
Feature 1

SECTION 1 – Moons Dataset Questions (2 Questions)

Q1. Which kernel was most effective for this dataset?

Answer: For the Moons dataset, the RBF (Radial Basis Function) kernel performed the best. This is because the dataset is non-linearly separable — the points form crescent shapes that can't be divided by a straight line. The RBF kernel maps data into a higher-dimensional space, allowing the model to form curved boundaries that perfectly capture the shape of the classes.

In short:

•Linear kernel → underfit (too simple, straight boundary)

•Polynomial kernel → can fit but may overcomplicate

•RBF kernel → best trade-off between fit and smoothness.

Q2. Why might the Polynomial kernel have underperformed here?

Answer: The Polynomial kernel may have underperformed because it tries to fit the data using polynomial curves, which can be too rigid or too complex for this dataset's natural circular boundaries. For low degrees, it can't capture enough curvature; for higher degrees, it overfits and forms noisy, irregular boundaries.

Summary: The Moons dataset needs smooth, radial boundaries → Polynomial kernel either overshoots or undershoots, leading to reduced accuracy.

SECTION 2 – Banknote Dataset Questions (2 Questions)

Q3. Which kernel was most effective for this dataset?

Answer: For the Banknote dataset, the Linear kernel performed the best. The banknote features (variance, skewness, kurtosis, entropy) are largely linearly separable, meaning a straight line (or plane in higher dimensions) can distinguish genuine from forged notes effectively.

RBF and Polynomial kernels can model complex patterns but tend to overfit when the separation is already linear. Hence, Linear kernel SVM achieved higher precision and recall while maintaining simplicity.

Q4. Why might the Polynomial kernel have underperformed here?

Answer: The Polynomial kernel adds unnecessary complexity for an already linearly separable dataset. It may try to capture non-existent curvature, fitting noise instead of meaningful structure. This leads to overfitting and poorer generalization on the test data.

In short: Polynomial kernel ≠ beneficial for clean, linearly separable data.

SECTION 3 – Hard vs Soft Margin Questions (4 Questions)

Q5. Which margin (soft or hard) is wider?

Answer: The soft margin (C = 0.1) is wider. A smaller C value allows the SVM to prioritize maximizing the margin width even if some points are misclassified. This creates a more flexible boundary.

Hard margin (C = 100) is much narrower because the model tries to perfectly classify every point.

Q6. Why does the soft margin model allow "mistakes"?

Answer: Because the soft margin SVM introduces slack variables (ξ_i) that permit certain points to lie within or even beyond the margin. This helps the model handle noisy or overlapping data better and improves generalization to unseen samples.

In short: allowing a few mistakes now → performs better later.

Q7. Which model is more likely to overfit and why?

Answer: The hard margin model (C = 100) is more likely to overfit. By forcing perfect classification, it becomes overly sensitive to noise and outliers, adjusting the decision boundary even for minor variations in data.

Overfitting symptom: tight decision boundary hugging training points too closely.

Q8. Which model would you trust more for new data and why?

Answer: The soft margin model ($C = 0.1$) would be more trustworthy for new, unseen data. Its flexible boundary tolerates small errors and thus generalizes better. In real-world datasets (which are rarely perfectly separable), this model yields more stable and robust predictions.