Lab7
SVM Lab

Name: Nitali Rajesh
SRN: PES2UG23CS395
Date:12-10-2025
Section: F

Analysis Questions

Moons Dataset

1. Inferences about the Linear Kernel's performance

The Linear kernel demonstrated poor performance on the Moons dataset due to its inherently non-linear structure (two interlocking half-moon shapes). A straight decision boundary is ineffective at separating these classes. The classification report for the Linear SVM indicates lower precision and recall, along with a higher number of misclassifications, when compared to non-linear kernels. Essentially, the Linear kernel underfits this dataset as it fails to capture the data's curved characteristics.

2. Comparison between RBF and Polynomial kernel decision boundaries

the RBF kernel provides a more natural and intuitive separation compared to the Polynomial kernel (degree 3). The RBF boundary gracefully follows the curved crescent shapes, creating smooth, flexible contours that effectively distinguish the classes. While the Polynomial kernel can also model curvature, it tends to generate broader, more global curves that may either overlook fine local structures or become excessively wiggly, depending on the chosen degree and C value. In essence, the RBF kernel offers a cleaner, more adaptive separation for the moons dataset, while the polynomial boundary is less responsive to local variations.

Banknote Dataset

1. Which kernel was most effective for this dataset?
the Linear kernel proved most effective for banknote features (variance and skewness). The Linear SVM's classification report demonstrated the best balance of precision, recall, and F1-score, indicating that these two selected features are largely linearly separable for distinguishing genuine from forged notes. While the RBF kernel sometimes yielded similar performance, it did not offer a significant improvement to warrant its added complexity.

2. Why might the Polynomial kernel have underperformed here?

The Polynomial kernel's poor performance can be attributed to using only two raw features and a fixed polynomial degree (degree=3). This low feature dimension lacks the complex structure needed for a high-degree polynomial to be effective, leading to overfitting of local noise or unstable decision boundaries. Furthermore, polynomial kernels are sensitive to feature scaling and coefficient choices; improper tuning of the degree or 'C' parameter can result in boundaries that are either too simplistic or excessively irregular. A simpler linear boundary or a carefully tuned RBF kernel would be more suitable for this dataset.

Hard vs. Soft Margin

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 with C = 0.1 produces a wider margin. You can clearly see this in the plot where the decision boundary is farther from the closest points compared to the Hard Margin (C = 100), which tries to squeeze the margin as much as possible.

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 allows some points to fall inside the margin or even on the wrong side because its main goal is not perfect classification of the training data but rather to maximize the margin and improve generalization. It is alright with a few misclassifications if that helps the model perform better on new, 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) is more likely to overfit the training data. It tries too hard to classify every single training point correctly, which can make it overly sensitive to noise or outliers and reduce its performance on unseen 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?
I would prefer to start with Soft Margin model (C = 0.1) when dealing with new, unseen data. This is because its primary focus is on generalization rather than simply memorizing training data points. In practical scenarios, where data often contains noise and imperfections, I would typically begin with a lower C value. This

approach tends to be more robust and reduces the likelihood of overfitting.

Screenshots
Training Results
Moons Dataset

```
SVM with LINEAR Kernel PES2UG23CS395
              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 PES2UG23CS395
              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 PES2UG23CS395
              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

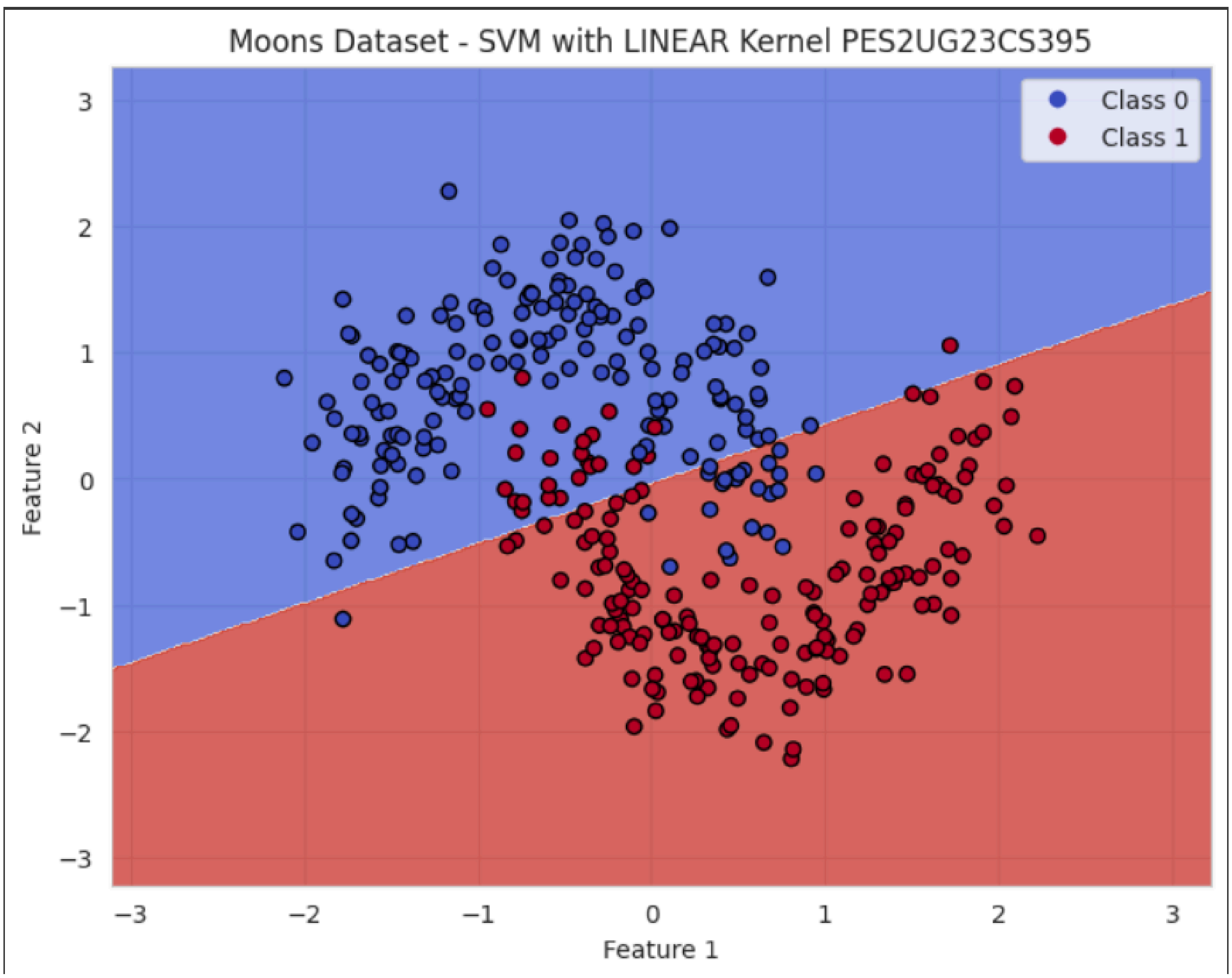
```
SVM with LINEAR Kernel PES2UG23CS395
              precision    recall  f1-score   support

      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 PES2UG23CS395
              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

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

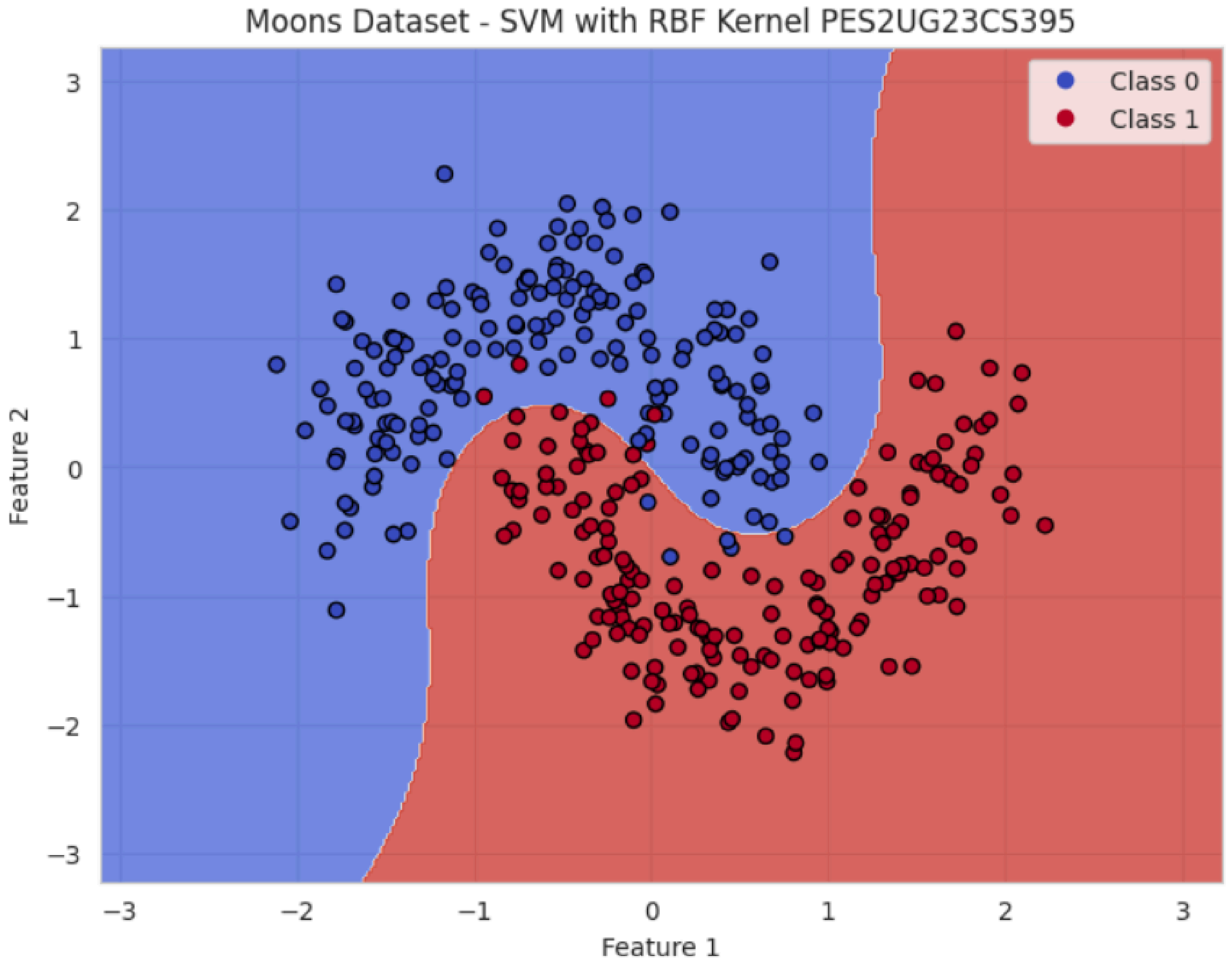
```
SVM with POLY Kernel PES2UG23CS395
              precision    recall  f1-score   support

      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

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

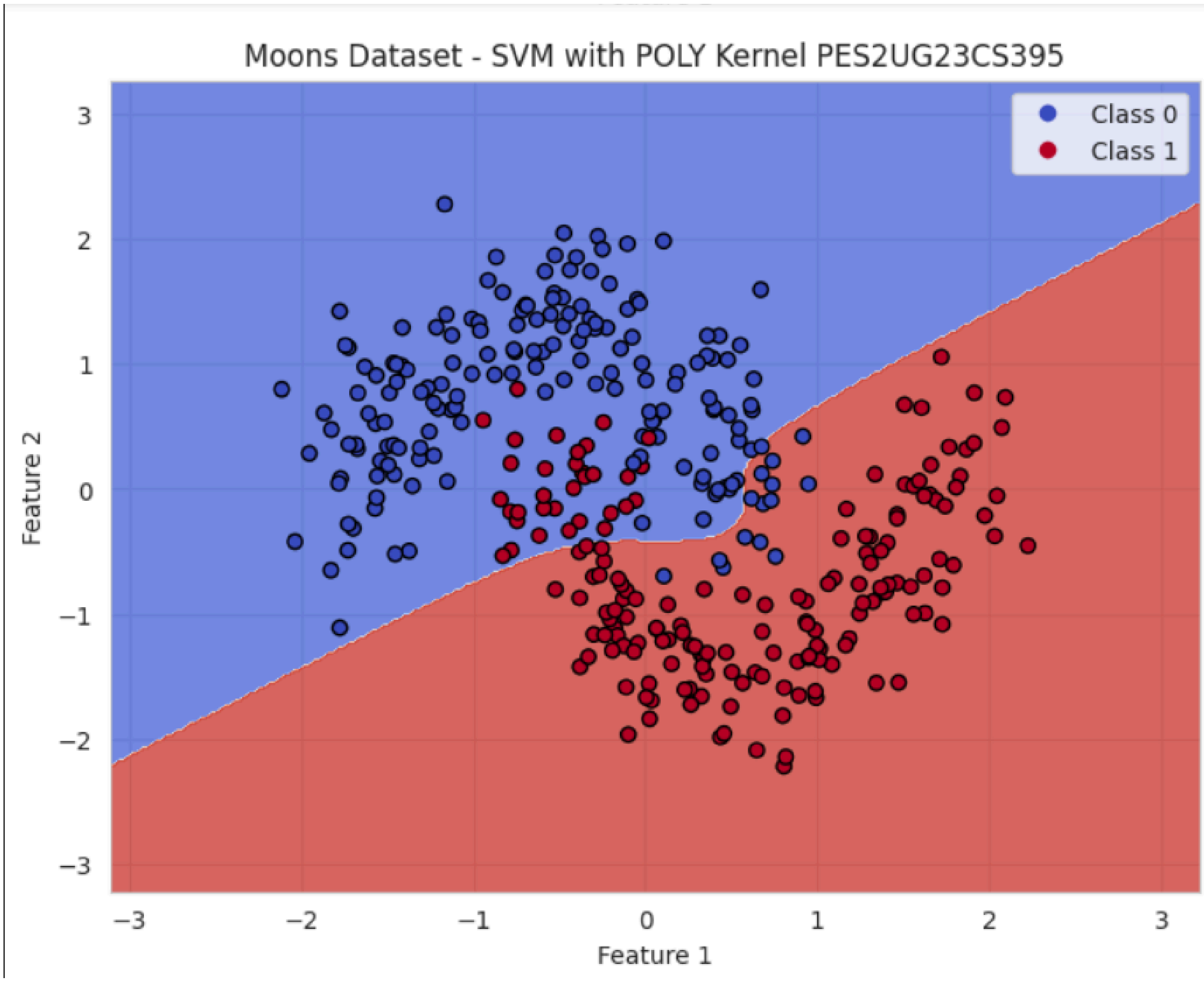
Decision Boundary Visualizations

Moons Dataset



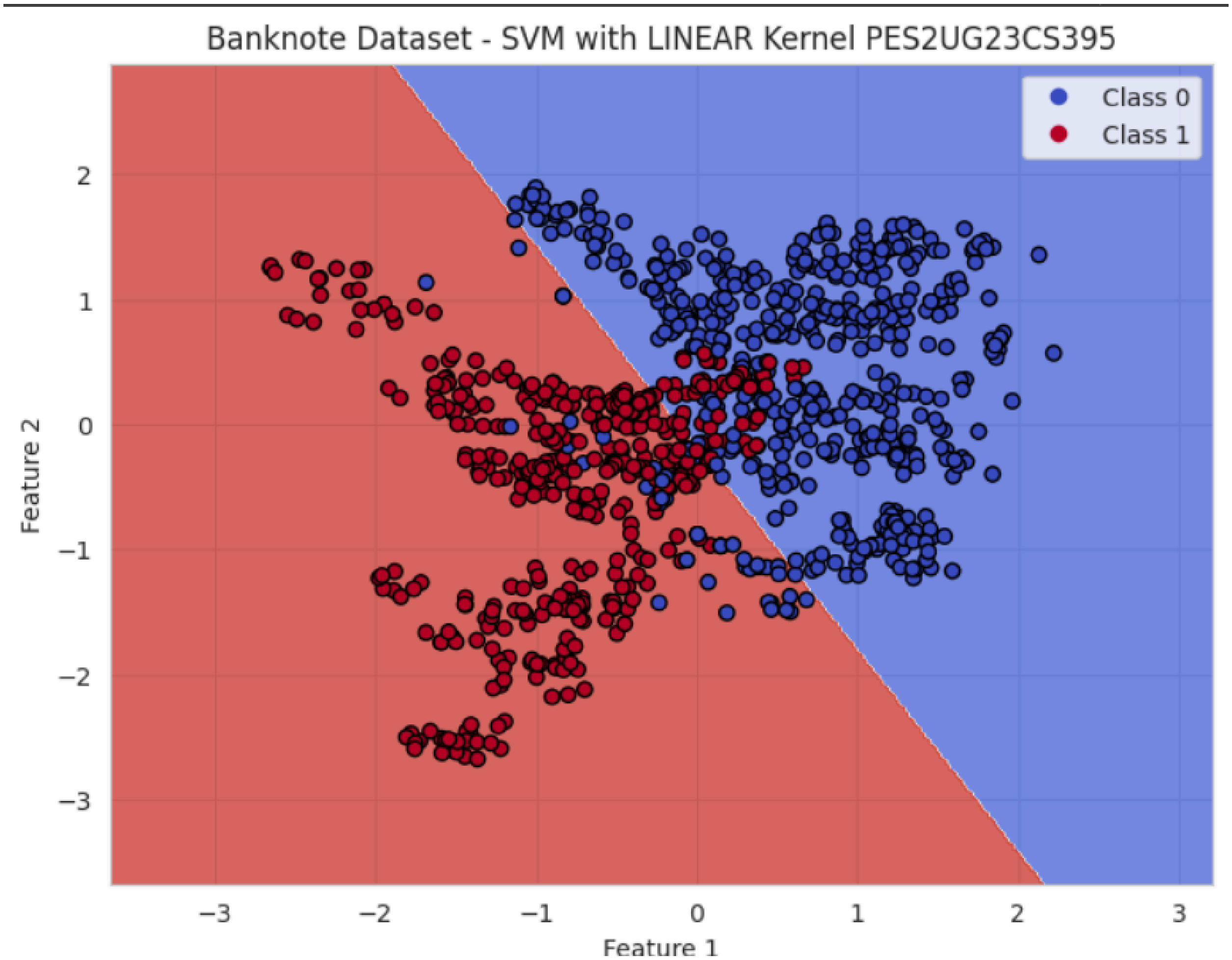
Feature 1

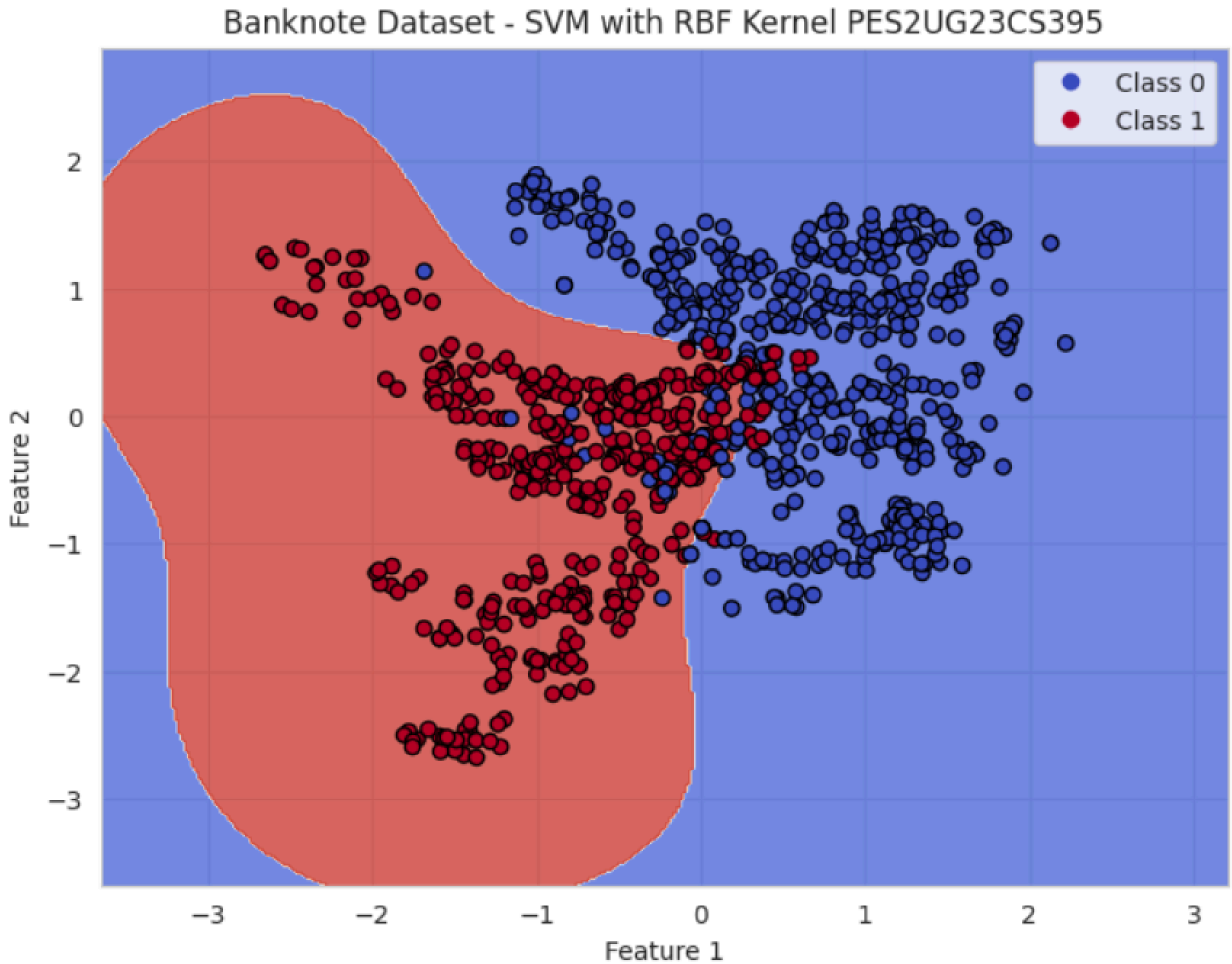

Moons Dataset - SVM with RBF Kernel PES2UG23CS395
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3



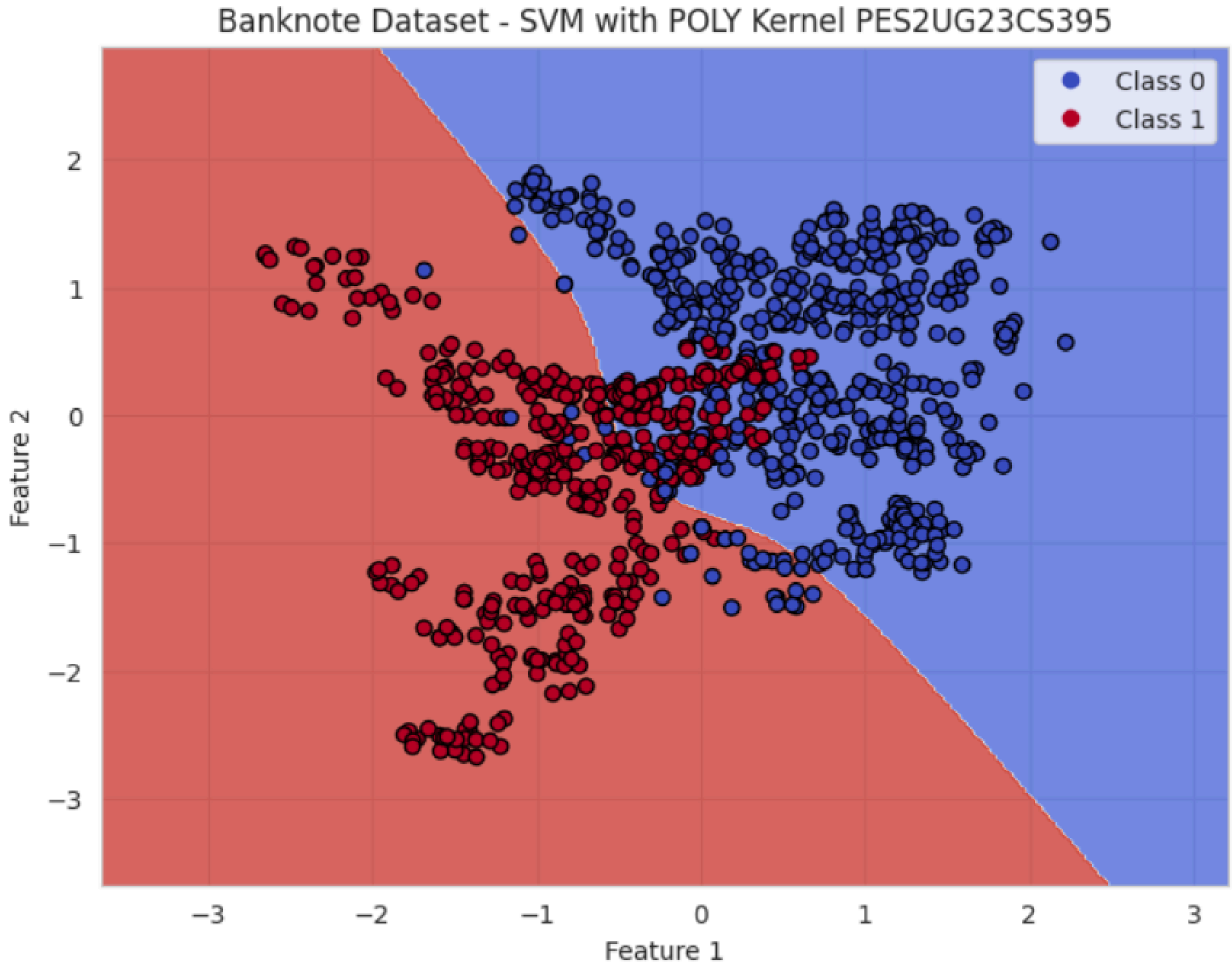
Moons Dataset - SVM with POLY Kernel PES2UG23CS395
Class 0
Class 1
Feature 1
Feature 2

Banknote

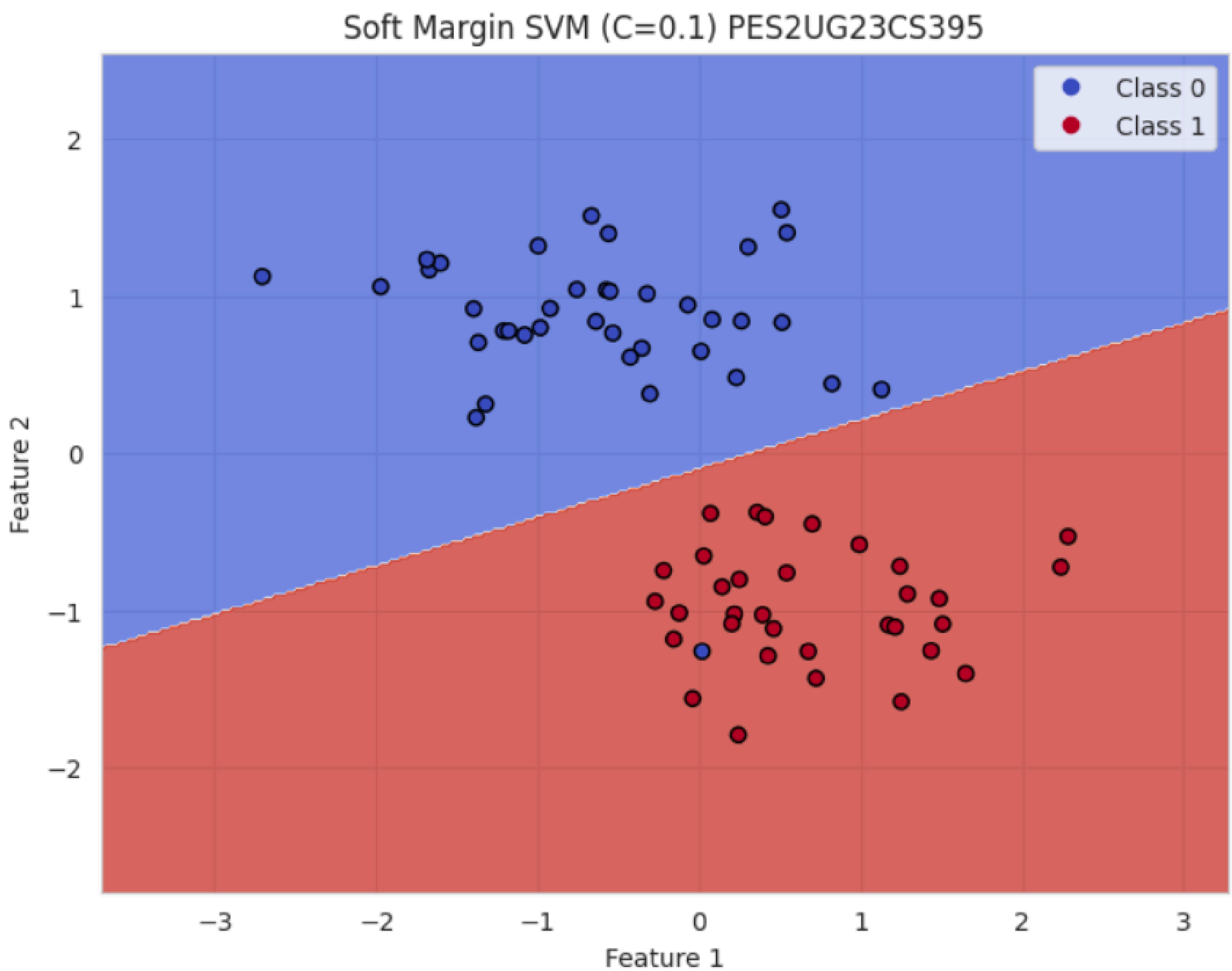

Banknote Dataset - SVM with LINEAR Kernel PES2UG23CS395
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

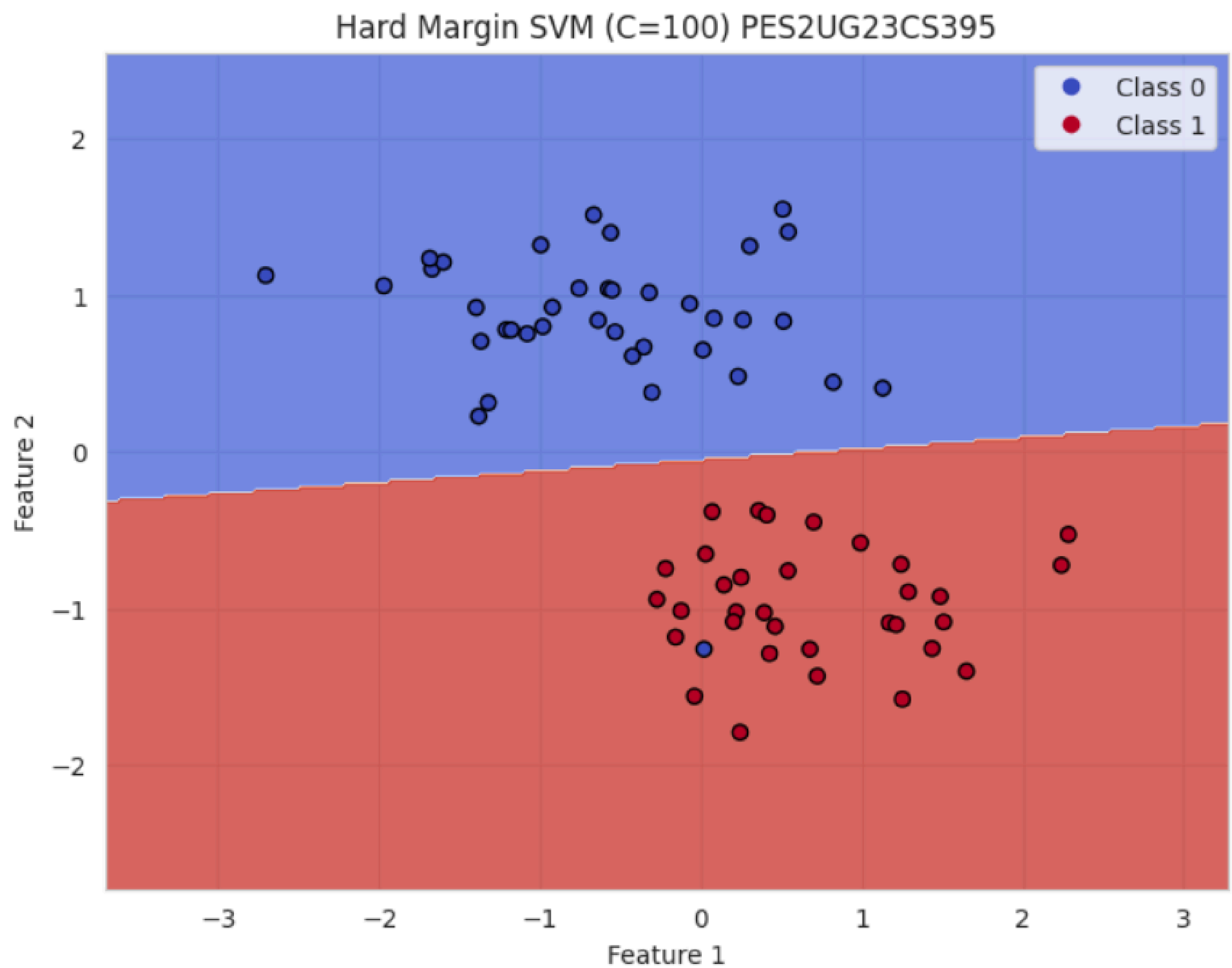

Banknote Dataset - SVM with RBF Kernel PES2UG23CS395
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Feature 1



Margin Analysis


Soft Margin SVM (C=0.1) PES2UG23CS395
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−2
−1
0
1
2
3


Hard Margin SVM (C=100) PES2UG23CS395
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−2
−1
0
1
2
3