LAB10

NAME: NIDHI N| SRN:PES2UG23CS384 | SEC:F

Moons Dataset Questions (2 questions):

1. Inferences aboutthe Linear Kernel's performance.

   The Linear kernel performed poorly on the Moons dataset. Because this dataset is not linearly separable, a straight line decision boundary is unable to capture the underlying pattern. This results in a high number of misclassifications and consequently low scores for accuracy, precision, and recall.

2. Comparison between RBF and Polynomial kernel decision boundaries.
   Both the RBF and Polynomial kernels create non-linear boundaries capable of separating the moon shapes. However, the RBF kernel's decision boundary appears more natural and smooth, closely following the curve of the data. The Polynomial kernel can also separate the data, but its boundary might be less regular or overly complex. The RBF kernel generally provides a more flexible and fitting solution for this type of intricate data distribution.

Banknote Dataset Questions (2 questions):

1. Which kernel was most effective for this dataset?
   The Linear kernel was most effective for the Banknote dataset. The visualization of this data shows that the two classes are largely linearly separable. A simple, straight line boundary is sufficient to classify the data with high accuracy. It provides a simple and robust solution without the unnecessary complexity of non-linear kernels.
2. Why might the Polynomial kernel have underperformed here?
   The Polynomial kernel may have underperformed because it creates a more complex boundary than necessary. Since the data is already well separated by a straight line, introducing a curve can lead to overfitting. The model might try to perfectly fit the noise and minor variations in the training data, resulting in a boundary that doesn't generalize as well to the unseen test data compared to the simpler linear kernel.

Hard vs. Soft Margin Questions (4 questions):

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

   The soft margin model (C=0.1) produces a wider margin. Its goal is to maximize the distance between the hyperplane and the support vectors, even if it means misclassifying a few training points.

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

   The soft margin model allows mistakes to achieve better generalization. It assumes that some data points might be outliers or noise. By tolerating a few misclassifications, it avoids creating a narrow, overly complex boundary that is too sensitive to these noisy points, making it more robust and likely to perform better on new, unseen data.

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

   The hard margin model (C=100) is more likely to be overfitting. The large C value imposes a heavy penalty on misclassification, forcing the model to fit every single training point perfectly. This causes the model to learn the noise in the training data rather than the underlying pattern, which harms its performance on new data.

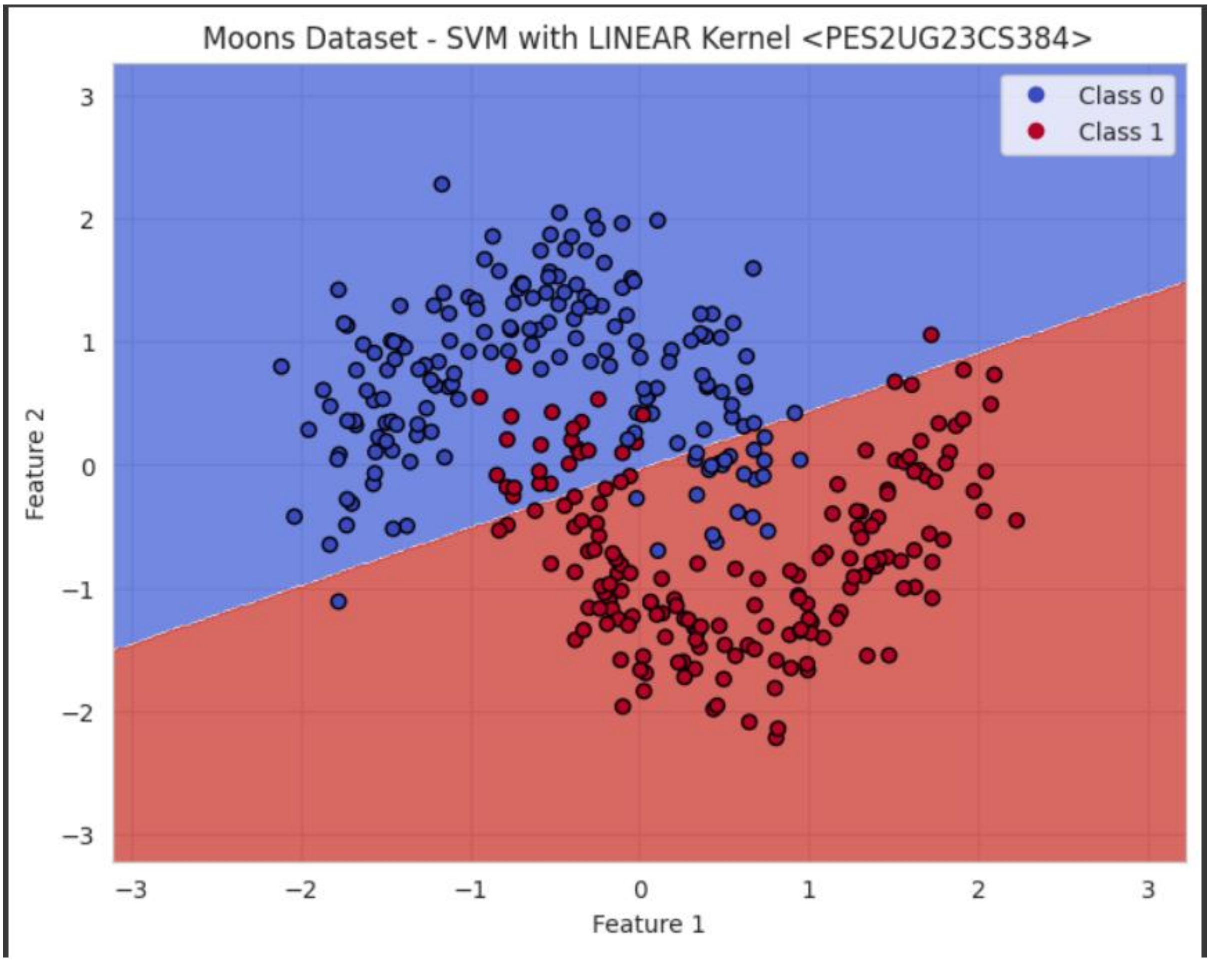
4. Which model would you trust more for new data and why?

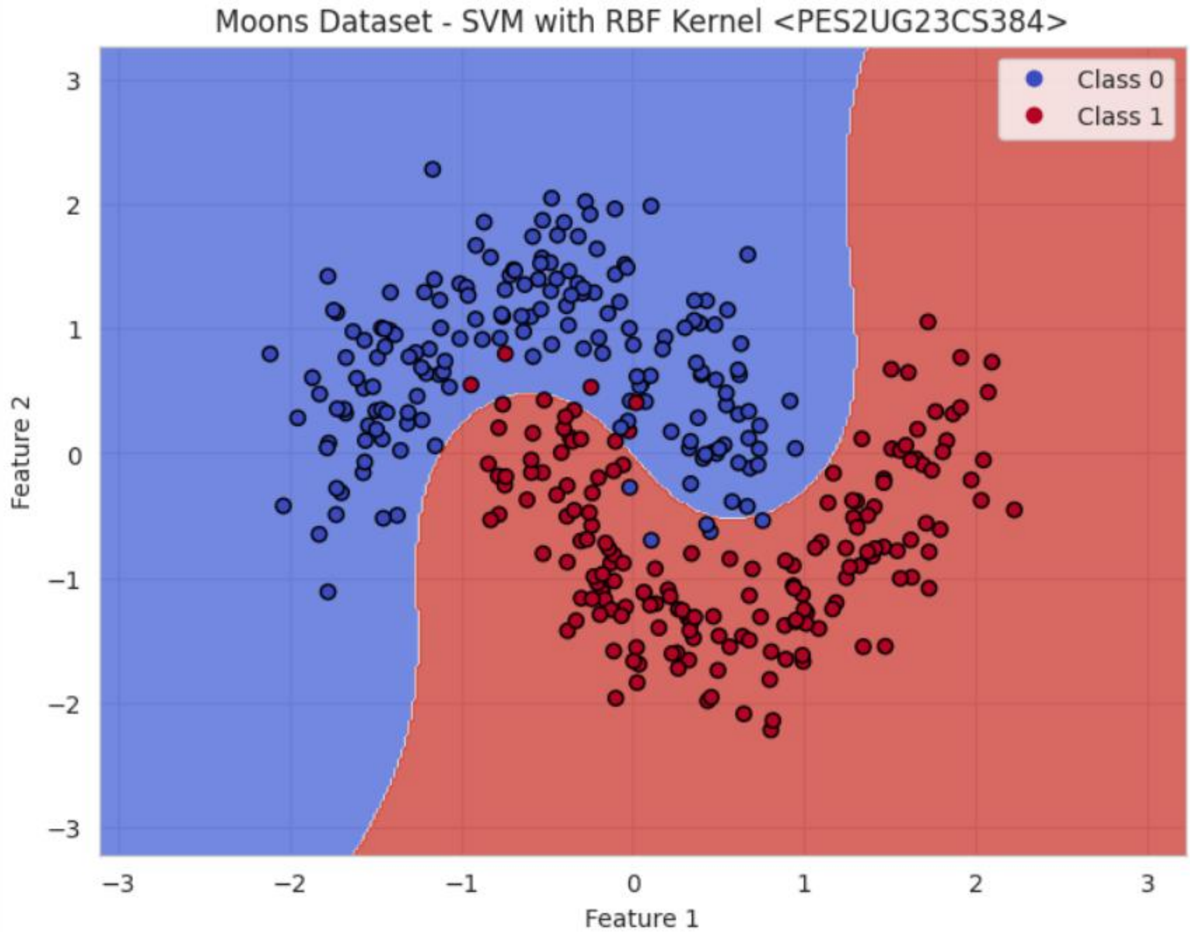
   I would trust the soft margin model (C=0.1) more for new data. Because real world data is often noisy, a model that is robust to outliers is preferable. The soft margin's wider boundary represents a more generalized solution that is less affected by the specific noise in the training set, making it more reliable for classifying new data points accurately.

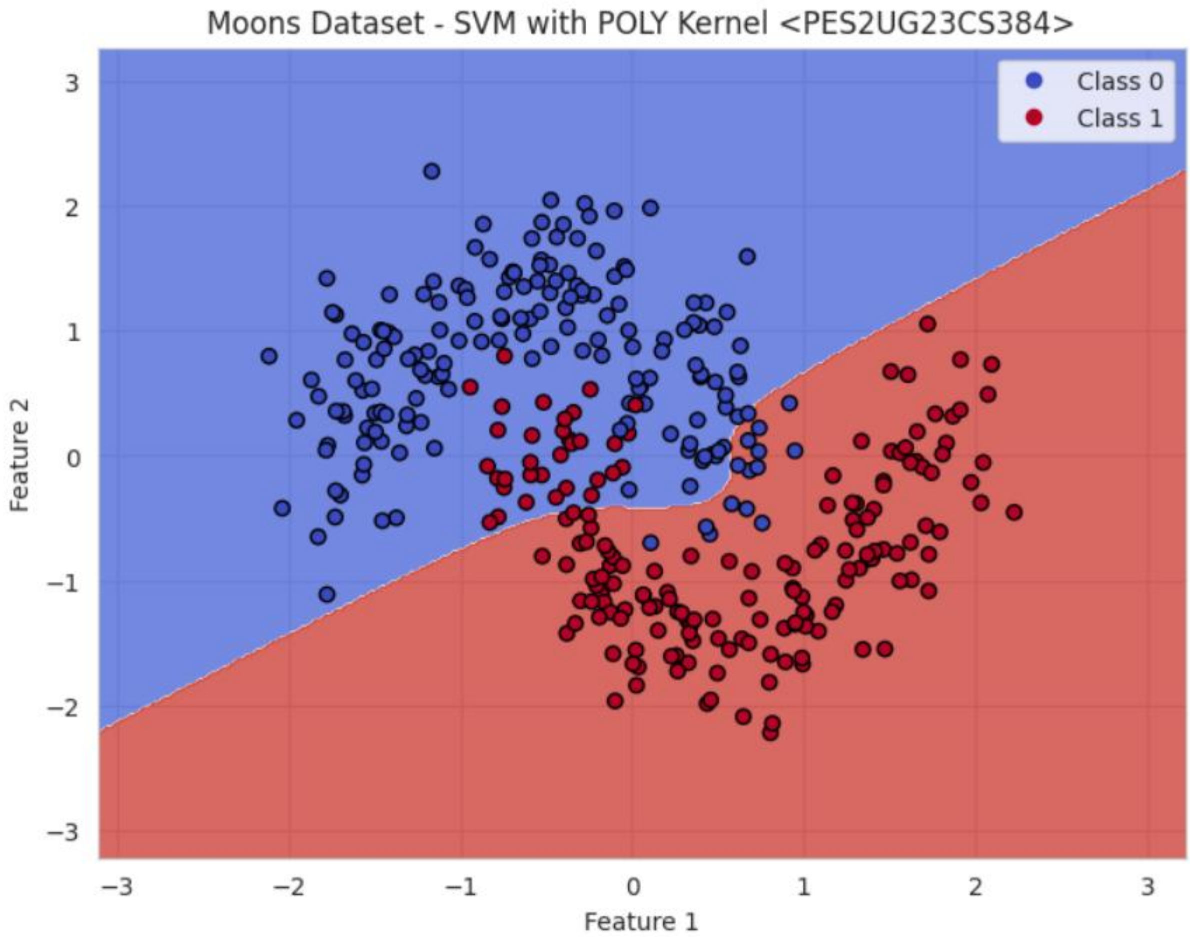
Screenshots:

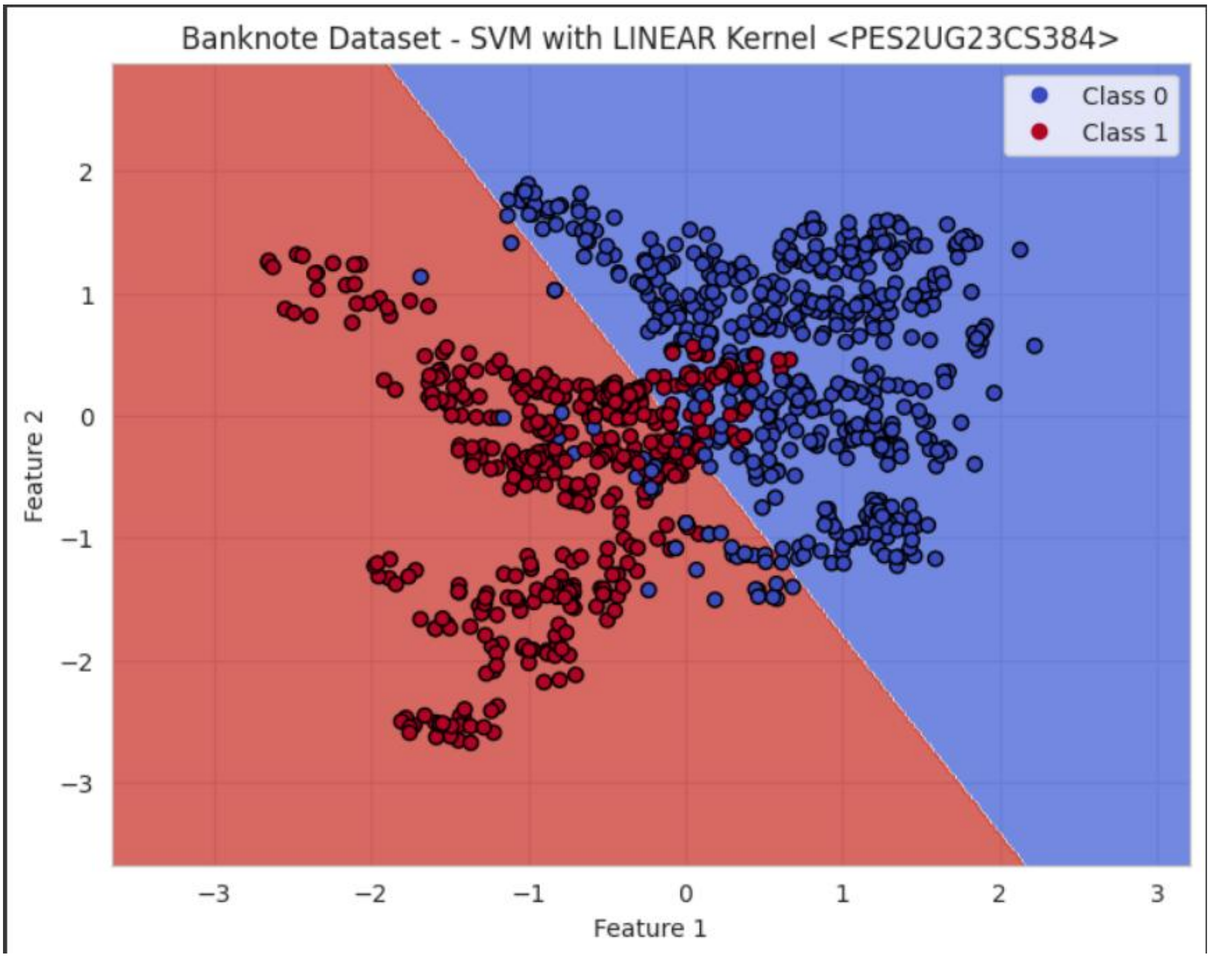
```
SVM with LINEAR Kernel <PES2UG23CS384>
              precision    recall  f1-score   support

           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 <PES2UG23CS384>
              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 <PES2UG23CS384>
              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

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

```
SVM with LINEAR Kernel <PES2UG23CS384>
              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 <PES2UG23CS384>
              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 <PES2UG23CS384>
              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

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

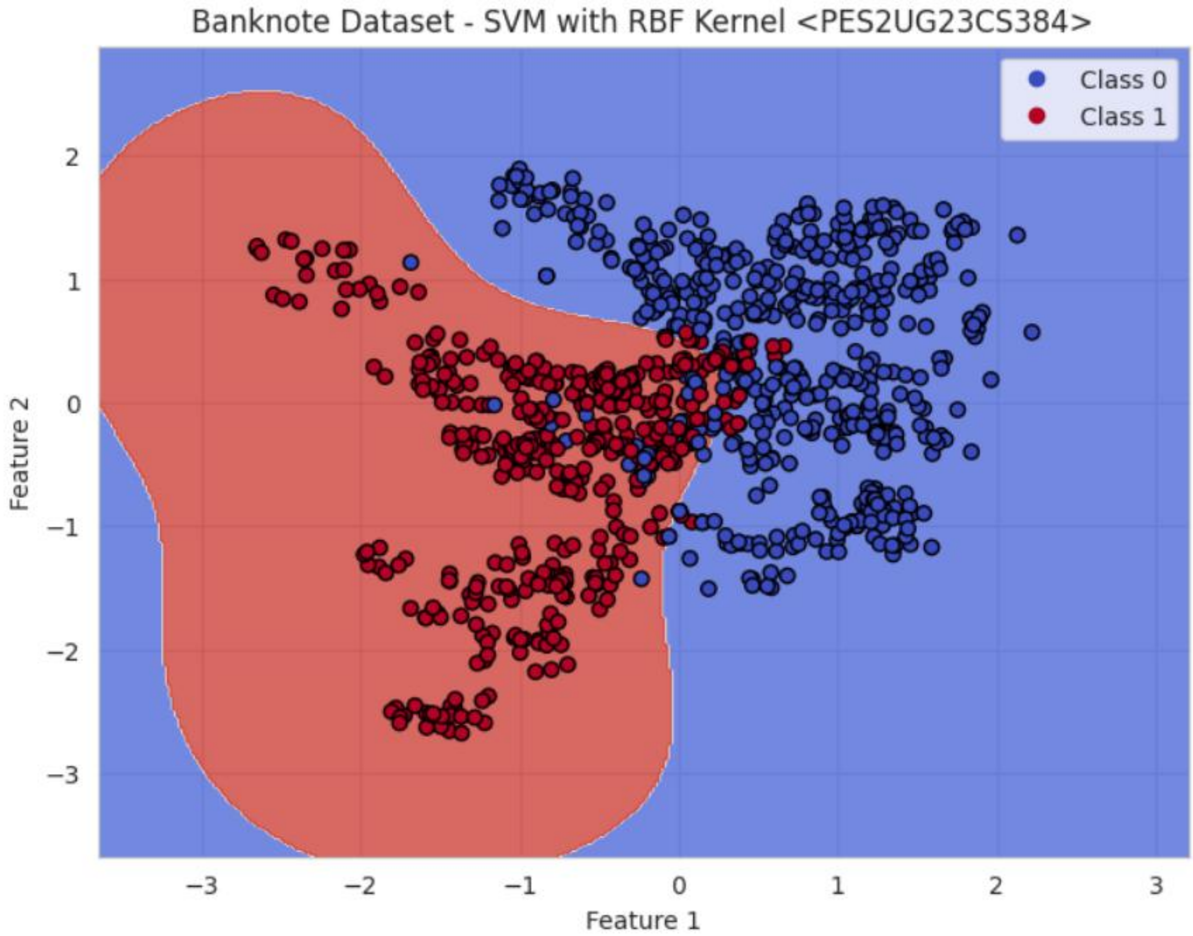

Moons Dataset - SVM with LINEAR Kernel <PES2UG23CS384>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

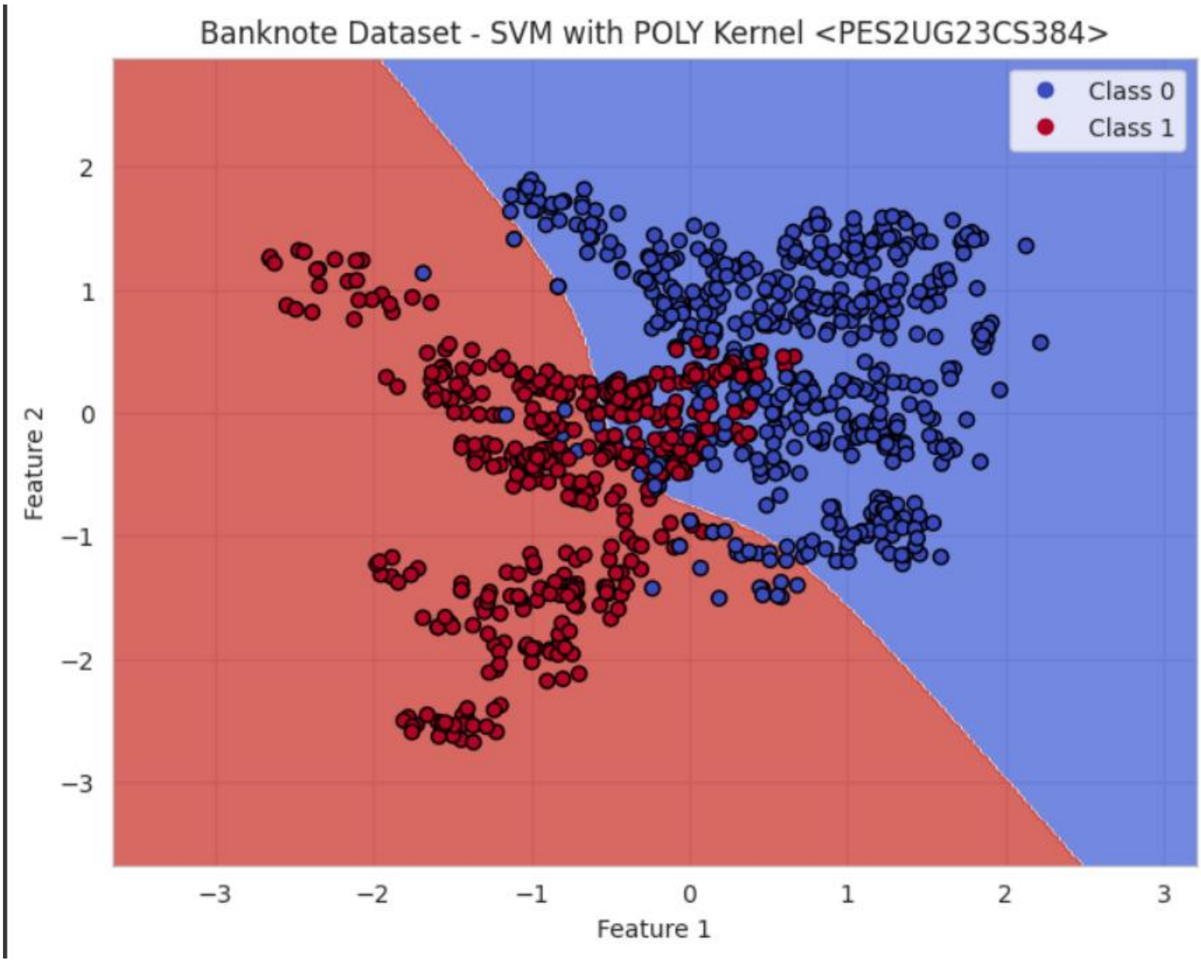

Moons Dataset - SVM with RBF Kernel <PES2UG23CS384>
Class 0
Class 1
Feature 1
Feature 2
−3
−2
−1
0
1
2
3


Moons Dataset - SVM with POLY Kernel <PES2UG23CS384>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

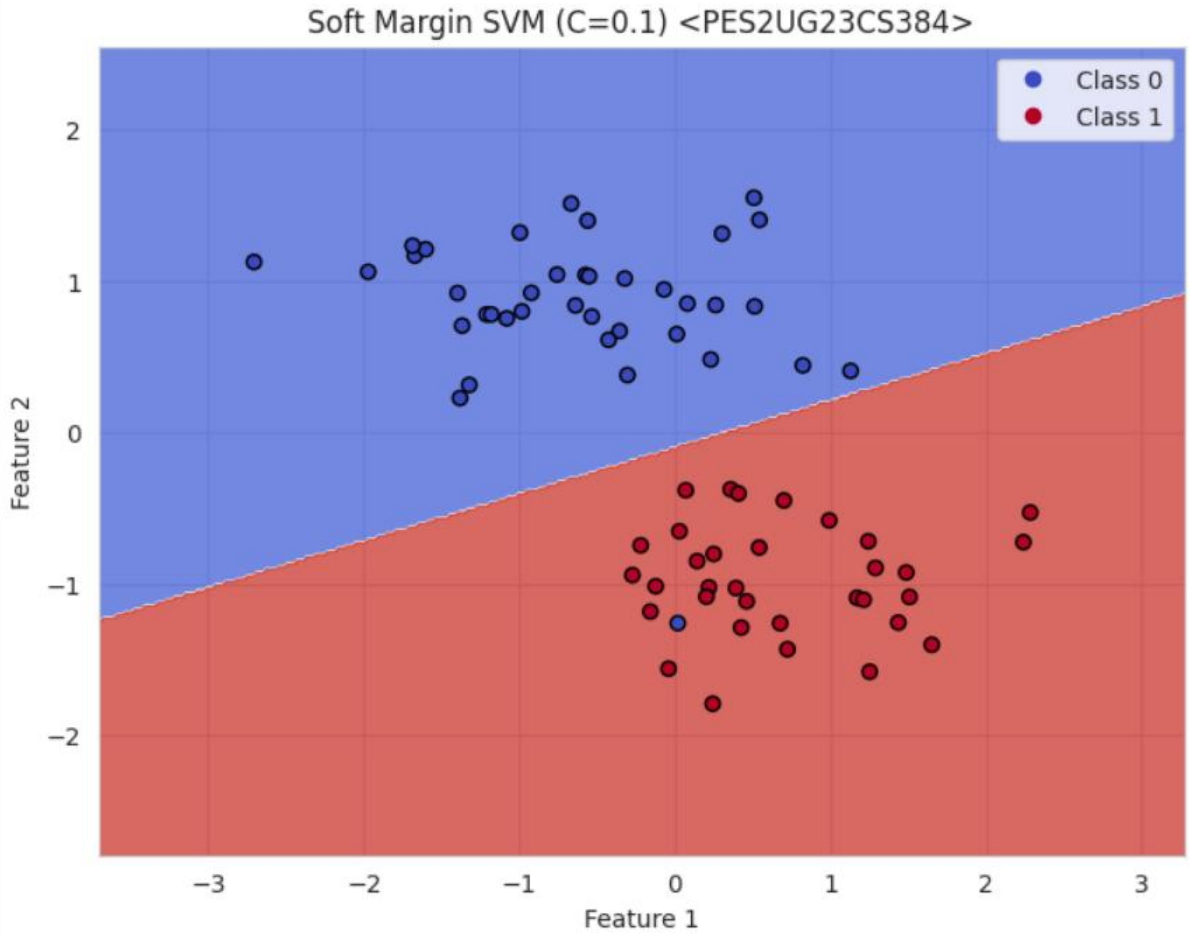


Banknote Dataset - SVM with LINEAR Kernel <PES2UG23CS384>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

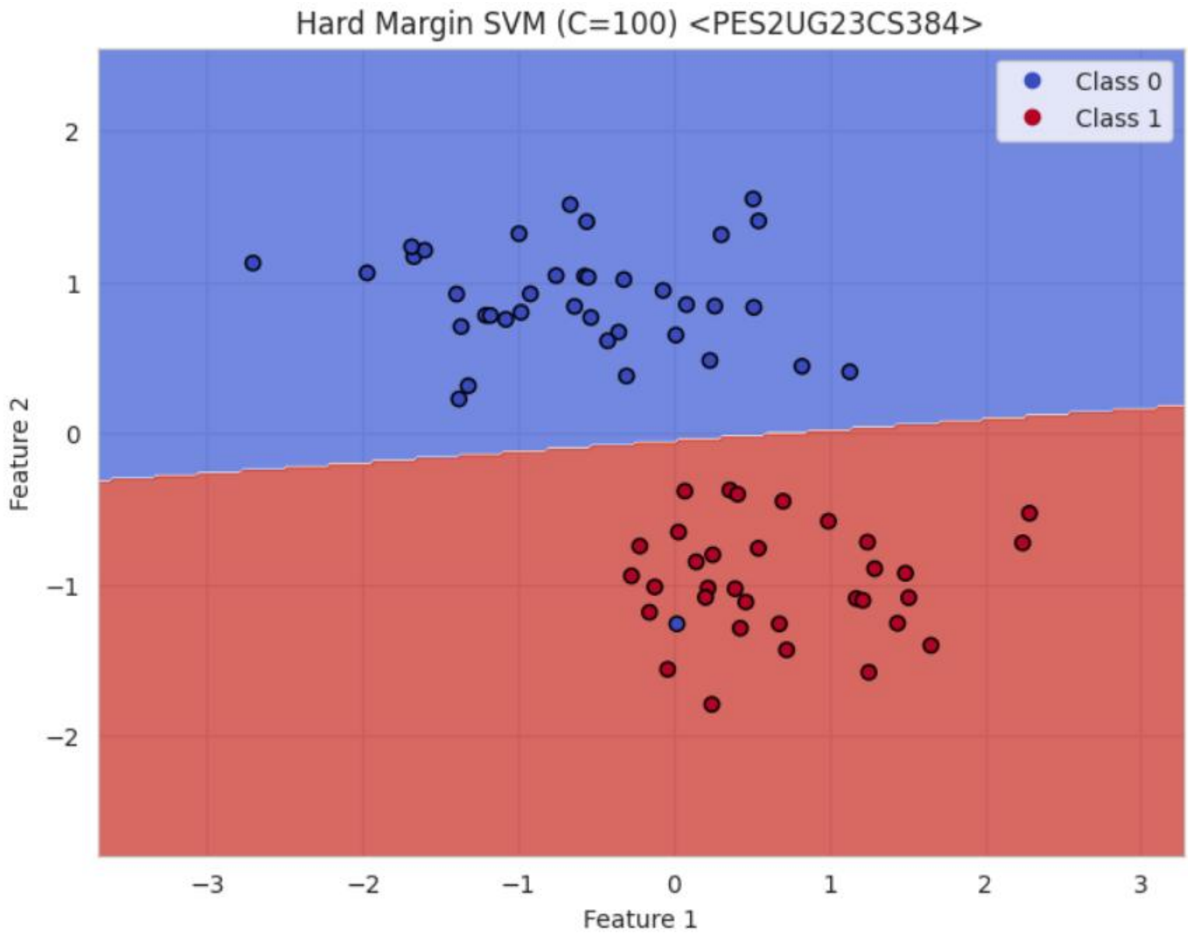

Banknote Dataset - SVM with RBF Kernel <PES2UG23CS384>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3



Banknote Dataset - SVM with POLY Kernel <PES2UG23CS384>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3



Soft Margin SVM (C=0.1) <PES2UG23CS384>
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) <PES2UG23CS384>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−2
−1
0
1
2
3