MACHINE LEARNING LAB
SUPPORT VECTOR MACHINE
WEEK-10

NAME: BOBBA KOUSHIK

SRN: PES2UG23CS133

SECTION: 5C

Moons Dataset Questions:

1. Inferences about the Linear Kernel's performance.

- The **Linear Kernel** achieved an accuracy of **0.87** and an F1-score of **0.87**, which is **lower** than RBF (0.97) and Polynomial (0.89).
- It produces a **straight-line decision boundary**, which cannot capture the **curved, non-linear structure** of the Moons dataset.
- Several points are **misclassified along the crescent edges**, showing **underfitting**.
- The model has **high bias and low variance**, meaning it's too simple for this dataset.
- Overall, the **Linear Kernel performs poorly** for non-linear data like Moons because it cannot adapt to curved boundaries.

2. Comparison between RBF and Polynomial kernel decision boundaries.

- The **RBF kernel** achieved **highest accuracy (0.97)** with a **smooth, flexible boundary** that perfectly follows the curved shapes of the two moons.
- The **Polynomial kernel** had **moderate accuracy (0.89)** and produced a **globally curved** but **less adaptive** boundary.
- The **RBF kernel** adapts **locally**, capturing small variations and fitting the true data shape more naturally.
- The **Polynomial kernel** can sometimes **overfit or underfit**, depending on degree and data distribution.
- **Conclusion:** RBF fits the Moons dataset **more naturally and accurately**, while Polynomial fits only approximately.

Banknote Dataset Questions:

1.Which kernel was most effective for this dataset?

- The **Linear Kernel** performed the best, showing the **highest accuracy (≈1.00)** and **F1-score** among all kernels.
- The dataset is **almost linearly separable**, so a straight-line decision boundary works effectively.
- The **RBF Kernel** also performed well but added unnecessary complexity for this simple structure.
- The Linear Kernel thus provides the **most efficient and interpretable** model for this dataset.

2. Why might the Polynomial kernel have underperformed here?

- The **Polynomial Kernel** introduces **non-linear and complex boundaries**, which are not needed for this mostly linear dataset.
- This added complexity can lead to **overfitting**, reducing accuracy on unseen test data.
- The **Banknote features** (like variance and skewness) already provide a clear linear separation between classes.
- Hence, the Polynomial Kernel performs worse because it **overcomplicates a problem that a simple linear model can solve better**.

Hard vs. Soft Margin Questions:

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

- The **Soft Margin (C = 0.1)** produces a **wider margin**.
- It allows more flexibility by not forcing all points to be classified perfectly.
- The boundary lies farther from the closest data points.
- This helps the model generalize better to unseen data.

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

- The **Soft Margin SVM** allows some points to be **inside the margin** or **misclassified**.
- This is done intentionally to **avoid overfitting** noisy data.
- The model prioritizes a **wider margin** over perfect accuracy on the training set.
- Its main goal is to **maximize generalization performance**, not to classify every point perfectly.

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

- The **Hard Margin (C = 100)** model is more likely to overfit.

- It tries to classify **every training point correctly**, even noisy or outlier points.
- This leads to a **narrow margin** and poor adaptability to new data.
- Overfitting occurs because the model becomes **too rigid and sensitive** to training noise.

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

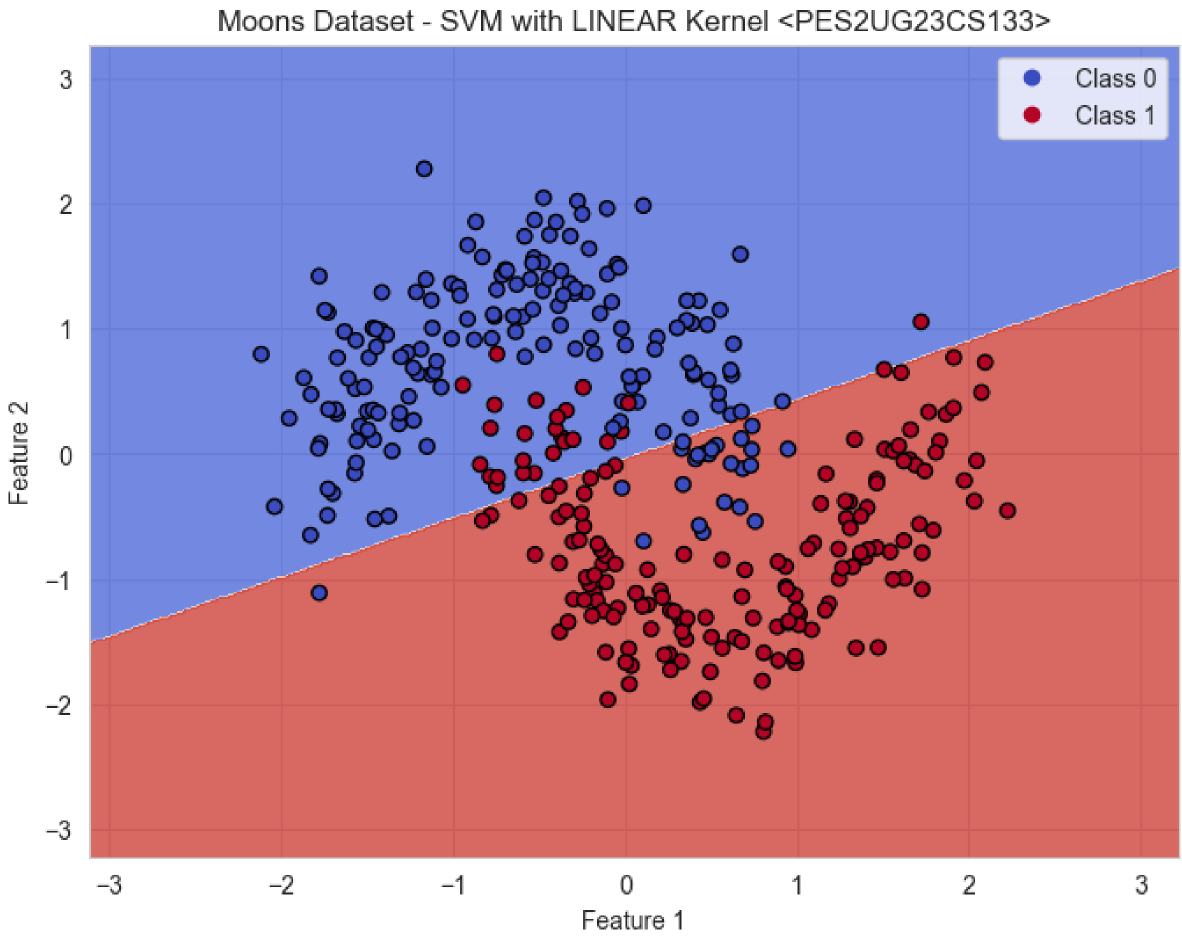
- The **Soft Margin (C = 0.1)** model is more reliable for **new, unseen data**.
- It generalizes better by allowing small errors during training.
- The wider margin helps handle **noise and variability** in real-world data.
- In practice, starting with a **lower C value** is preferred for noisy datasets.

SCREENSHOTS:

MOONS DATASET:

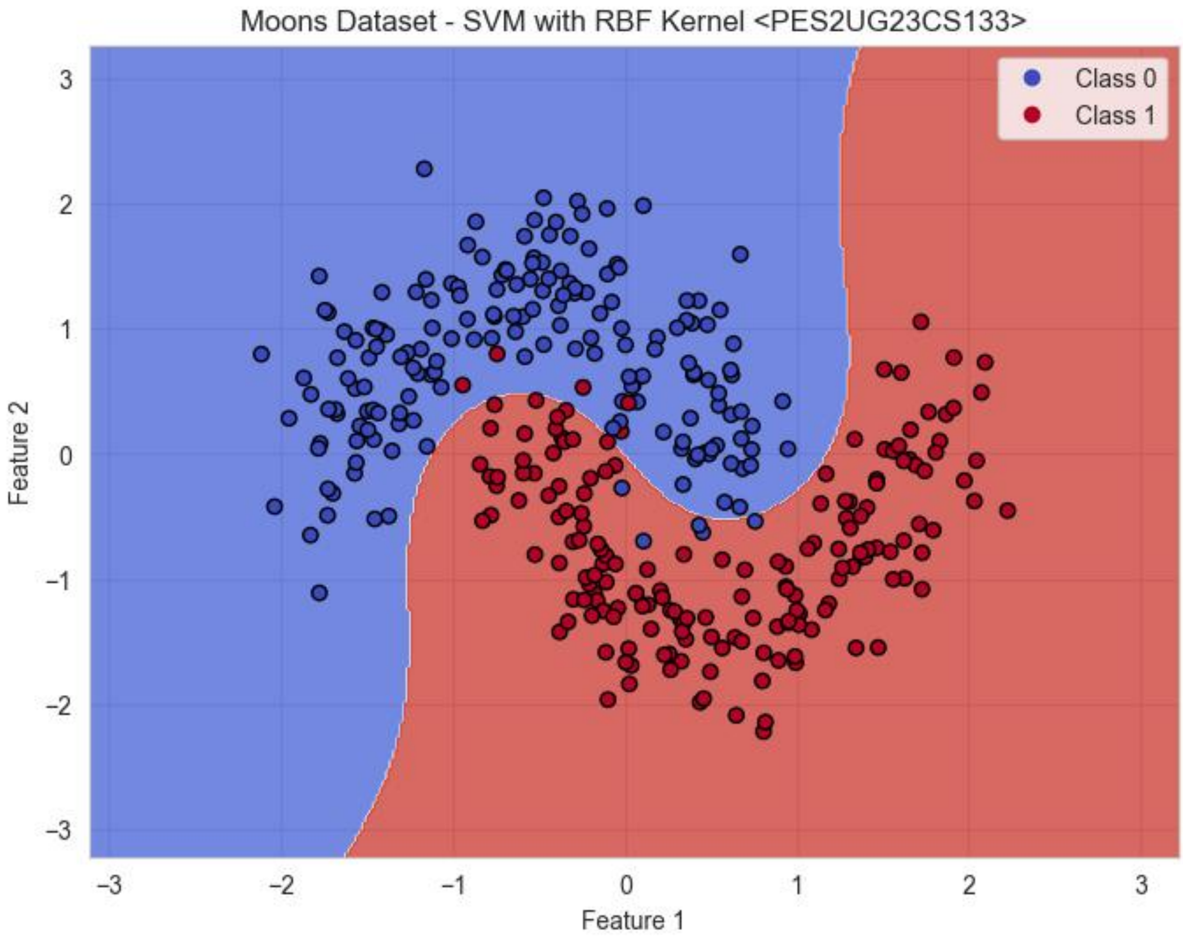
LINEAR KERNEL:

```
SVM with LINEAR Kernel <PES2UG23CS133>
              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

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



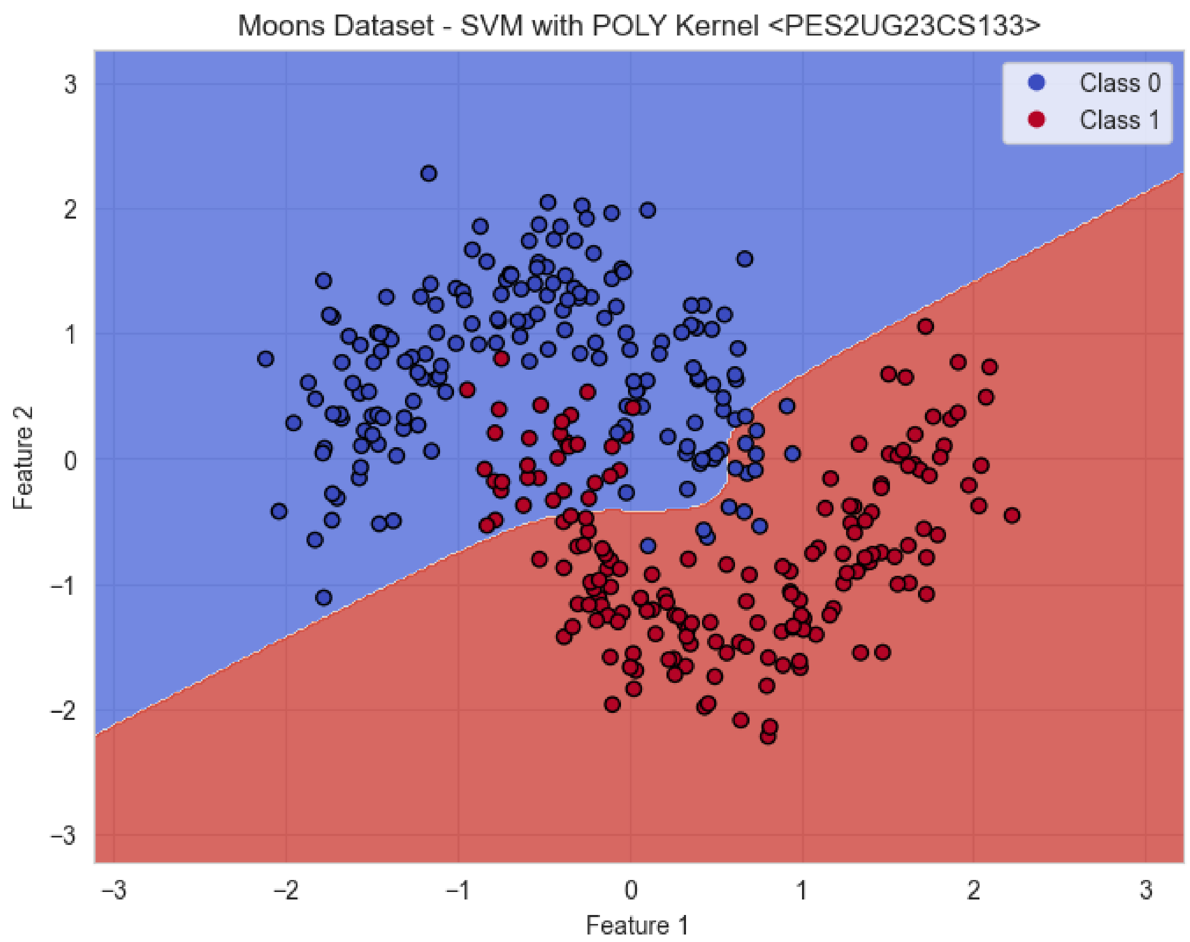
RBF KERNEL:

```
SVM with RBF Kernel <PES2UG23CS133>
              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

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



POLYNOMIAL KERNEL:

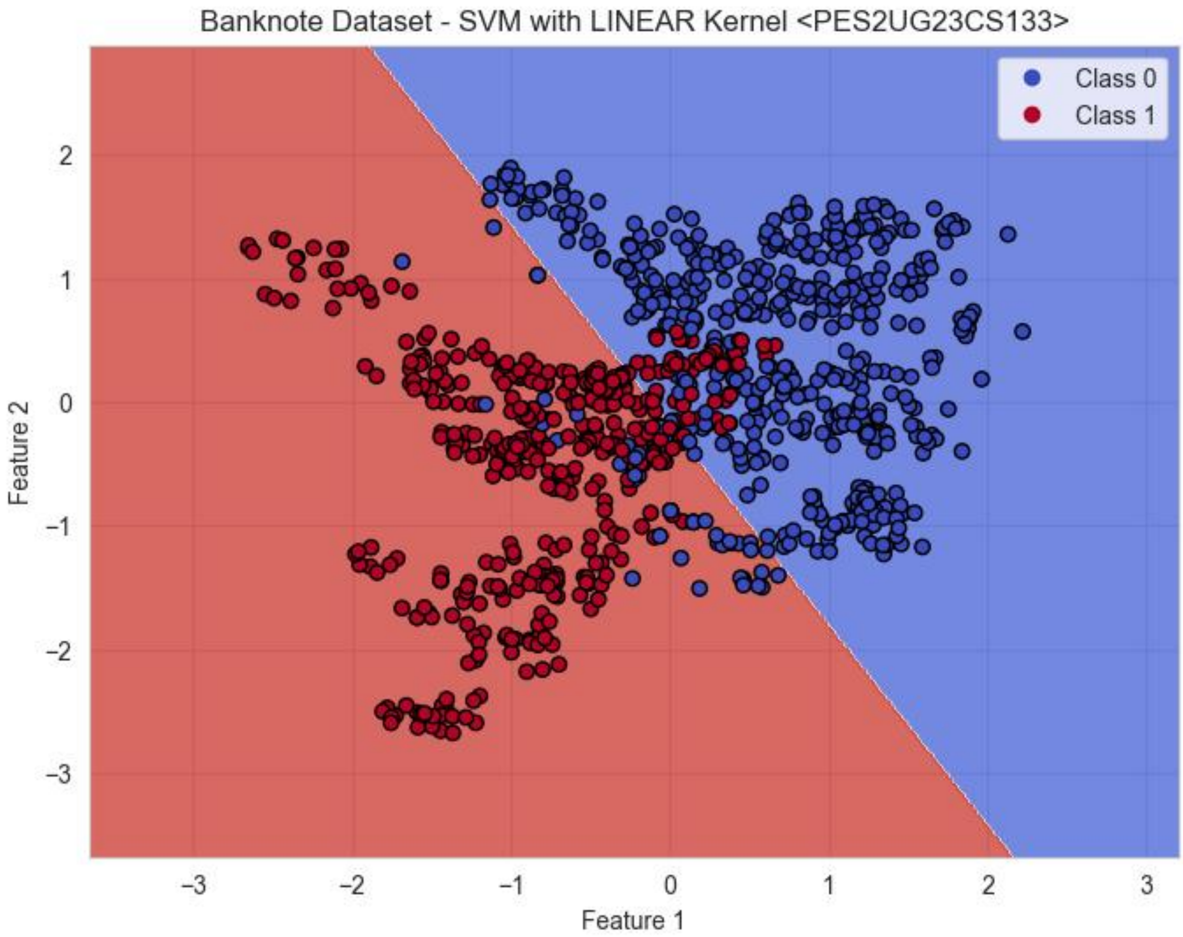
```
SVM with POLY Kernel <PES2UG23CS133>
              precision    recall  f1-score   support

           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:

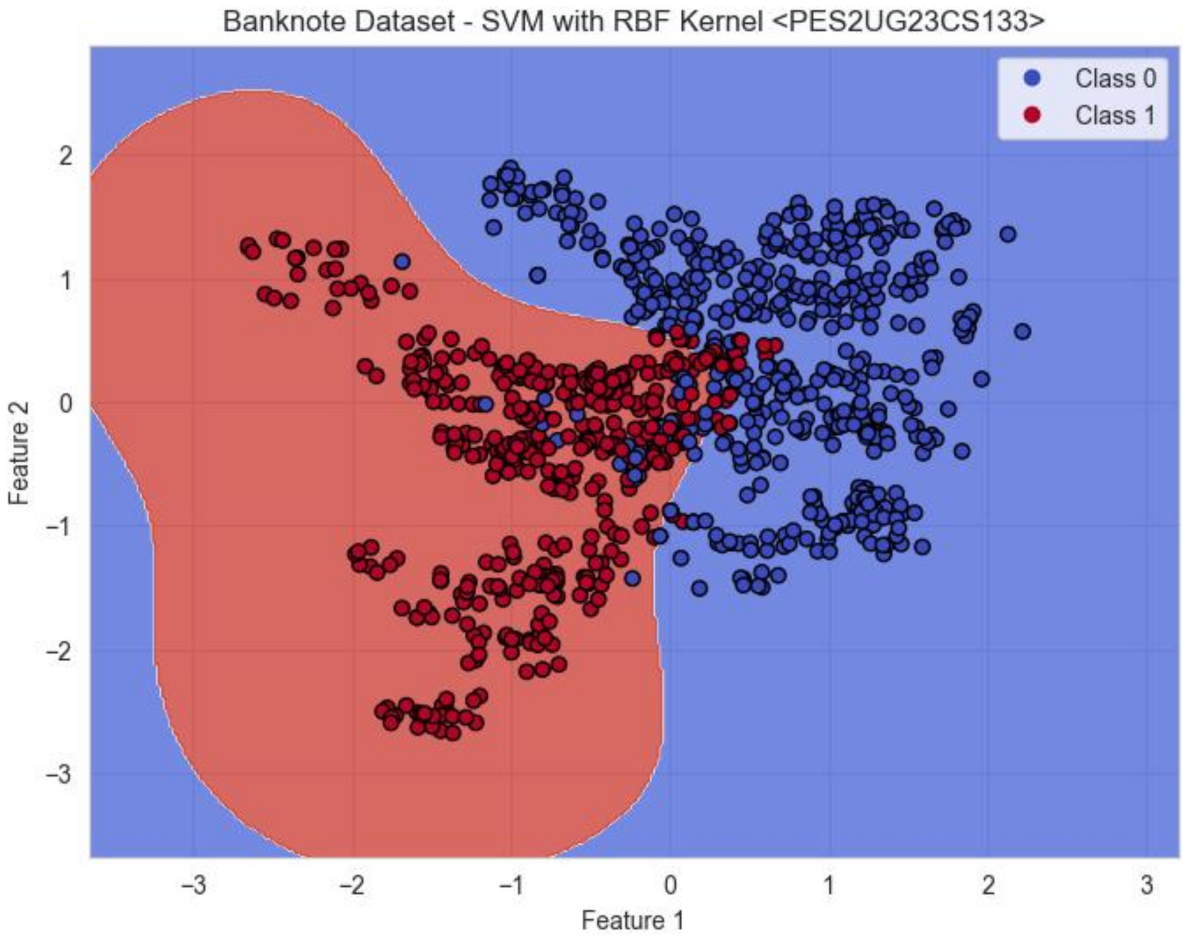
Linear Kernel:

```
SVM with LINEAR Kernel <PES2UG23CS133>
              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

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



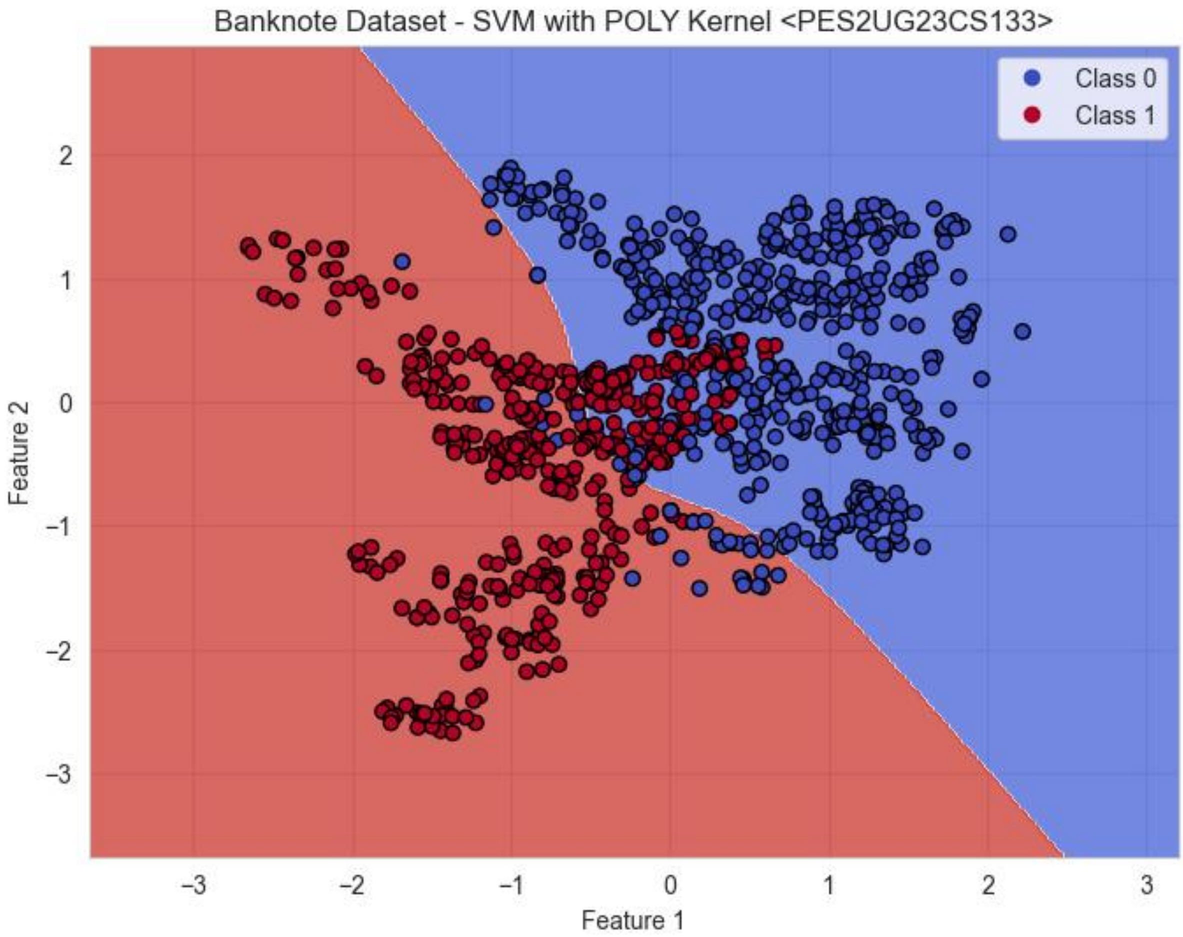
RBF kernel:

```
SVM with RBF Kernel <PES2UG23CS133>
              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

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

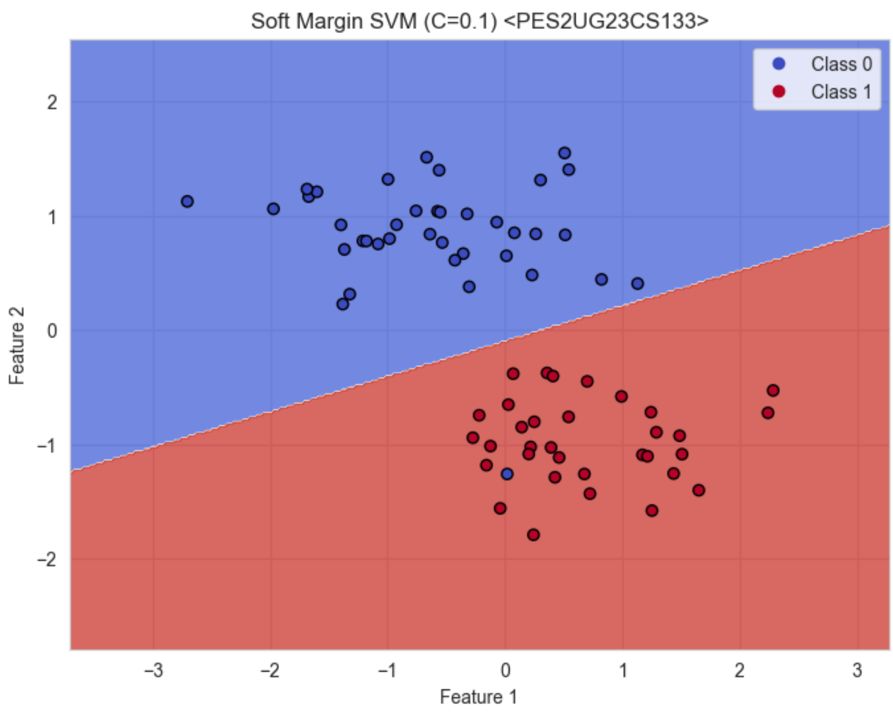


POLYNOMIAL Kernel:

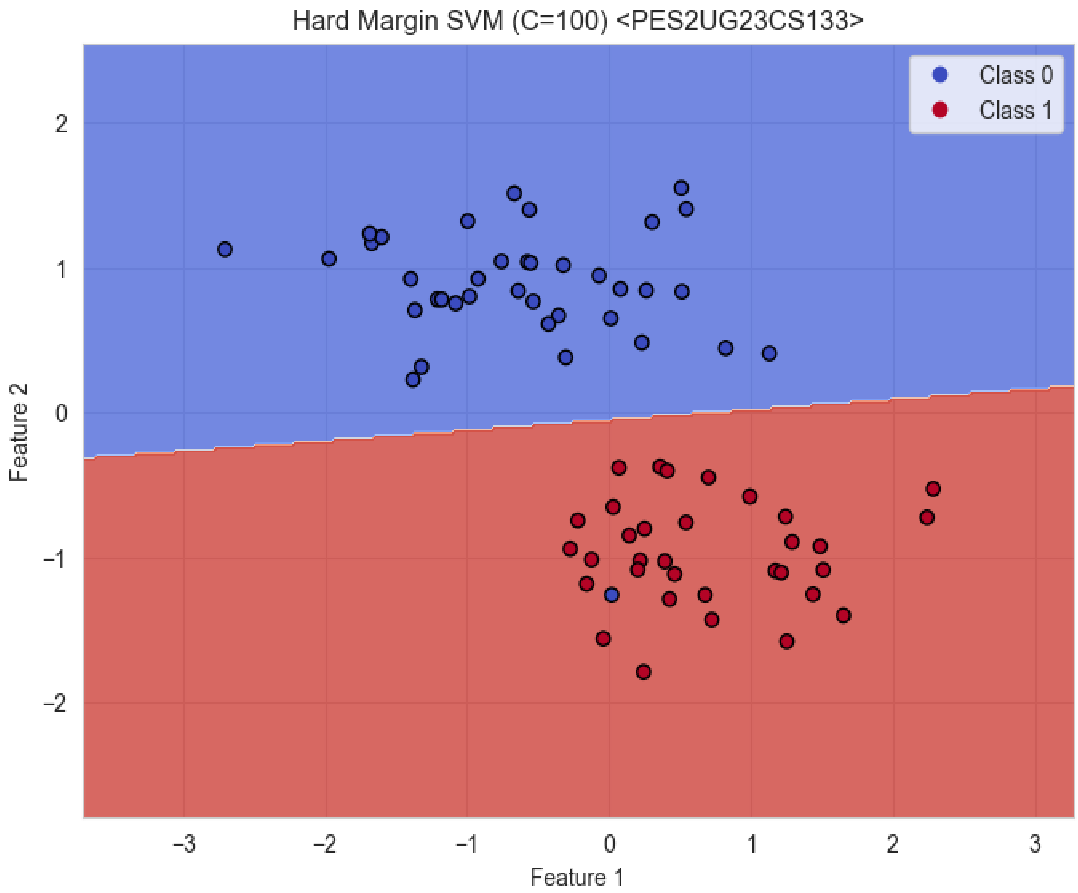
```
SVM with POLY Kernel <PES2UG23CS133>
              precision    recall  f1-score   support

      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 POLY Kernel <PES2UG23CS133>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

SOFT-MARGIN:


Soft Margin SVM (C=0.1) <PES2UG23CS133>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−2
−1
0
1
2
3

HARD-MARGIN:


Hard Margin SVM (C=100) <PES2UG23CS133>
Class 0
Class 1
Feature 1
Feature 2
2
1
0
−1
−2
−3
−2
−1
0
1
2
3