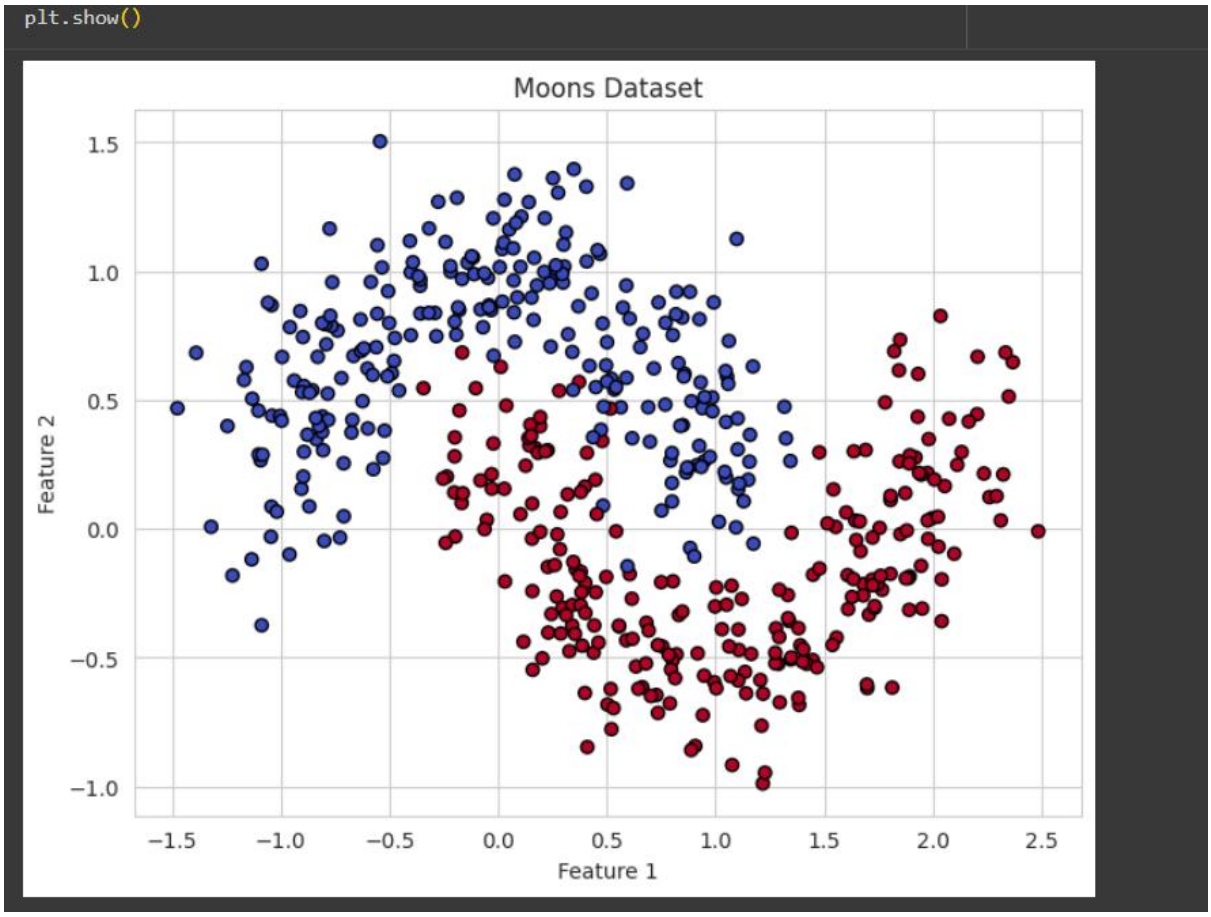
Name: Varun Kumar L	SRN: PES2UG24CS827
SEC:C	SUB: ML WEEK10

SVM Classifier Lab

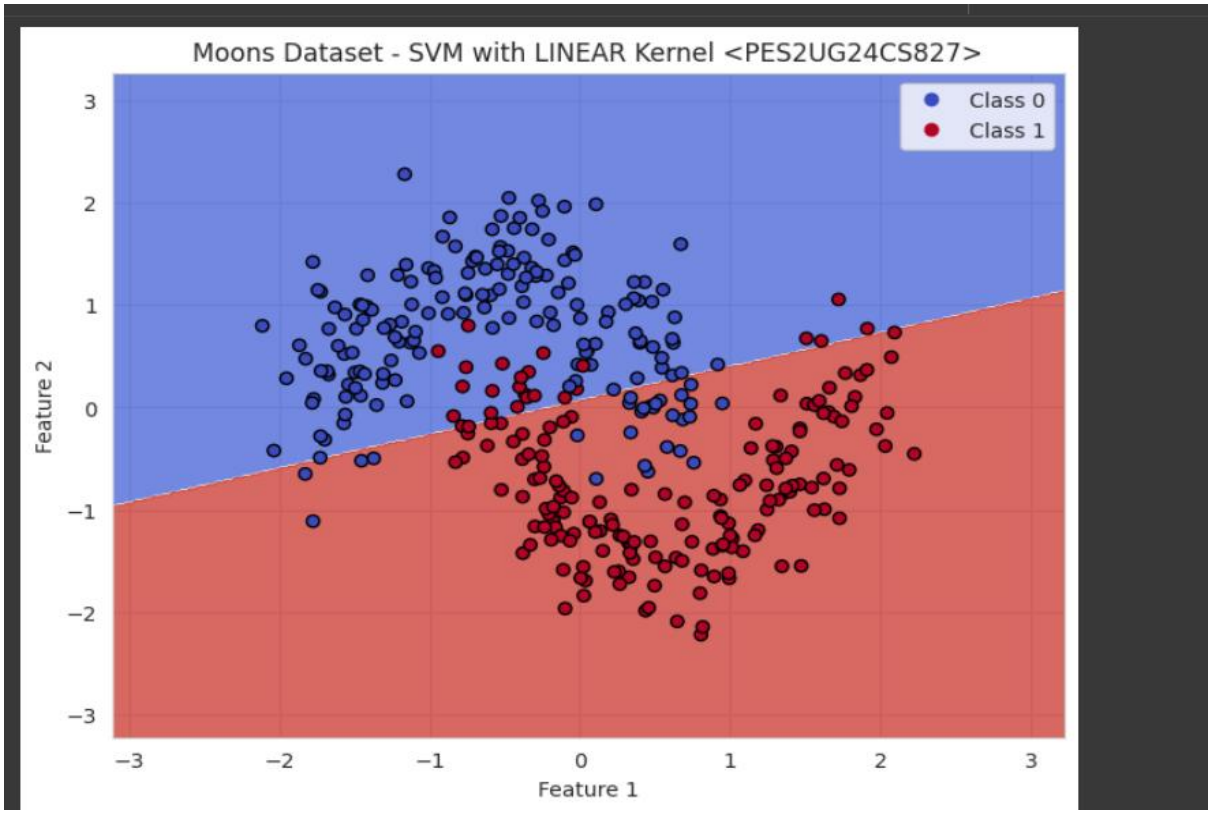
PART 1

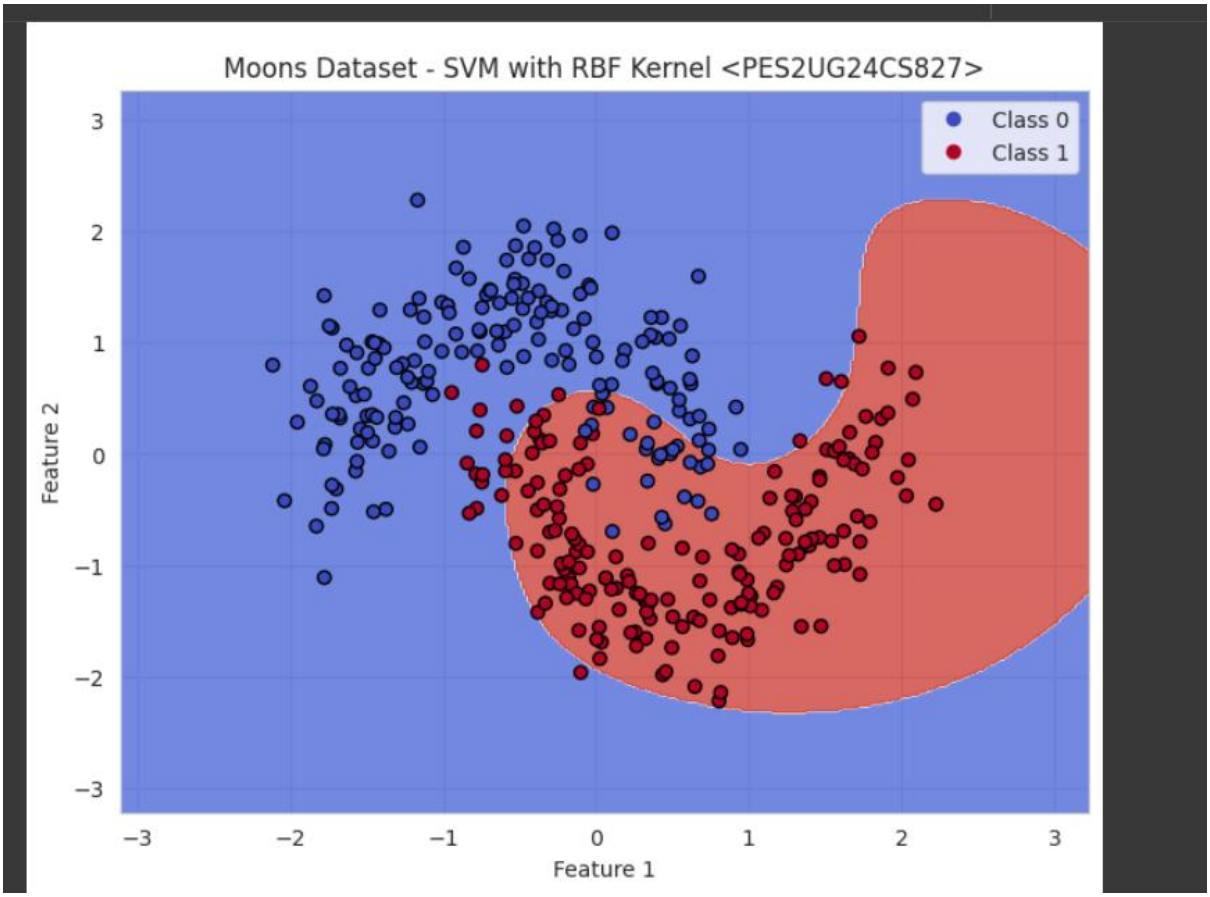
Dataset 1: The Moons Dataset

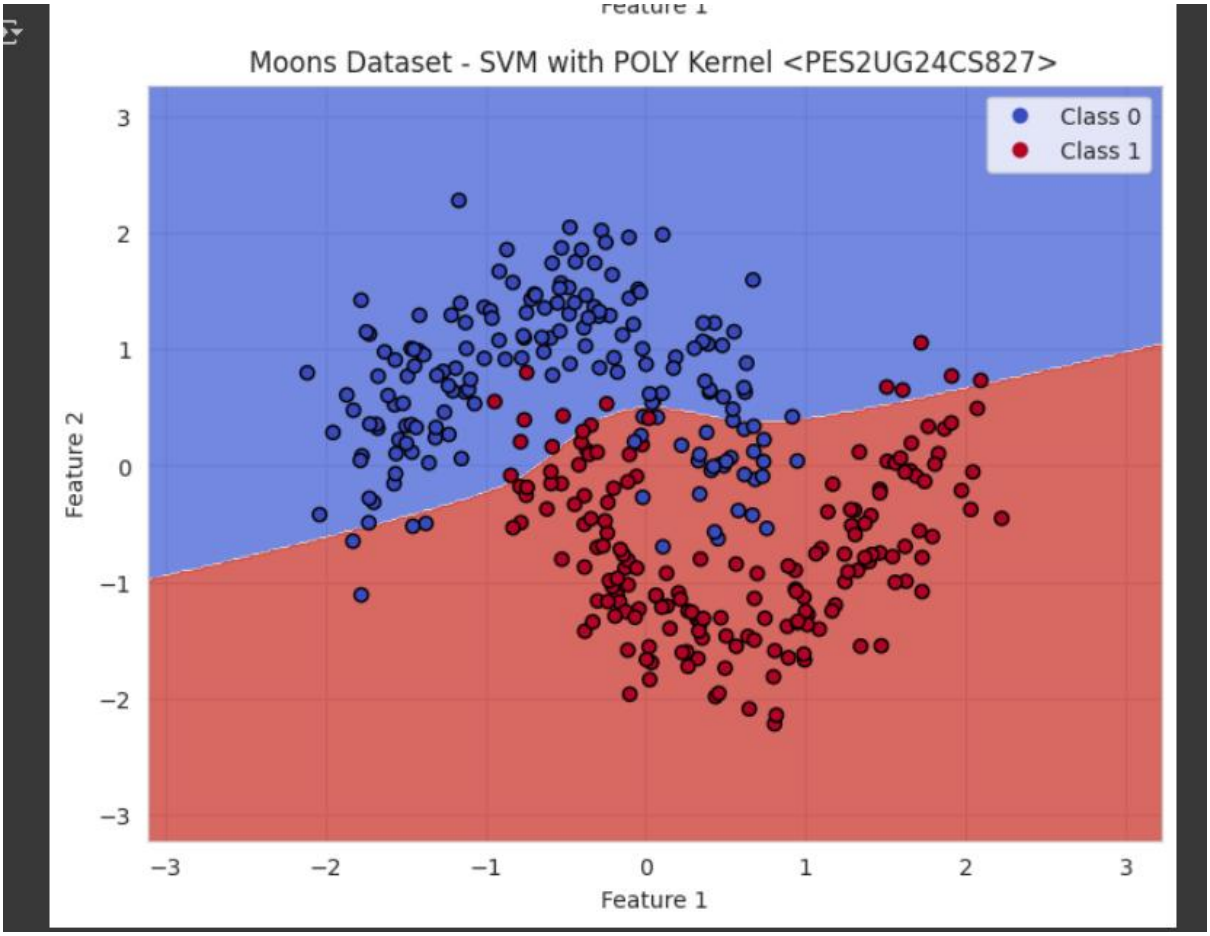


Step 1.2: Train and Evaluate SVM Kernels

```
SVM with LINEAR Kernel <PES2UG24CS827>
              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 <PES2UG24CS827>
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        75
           1       1.00      0.96      0.98        75

    accuracy                           0.98       150
   macro avg       0.98      0.98      0.98       150
weighted avg       0.98      0.98      0.98       150

----------------------------------------

SVM with POLY Kernel <PES2UG24CS827>
              precision    recall  f1-score   support

           0       0.93      0.88      0.90        75
           1       0.89      0.93      0.91        75

    accuracy                           0.91       150
   macro avg       0.91      0.91      0.91       150
weighted avg       0.91      0.91      0.91       150

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




Moons Dataset - SVM with RBF Kernel <PES2UG24CS827>
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 <PES2UG24CS827>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Moons Dataset Questions

1.Based on the metrics and the visualizations, what inferences about the performance of the Linear Kernel can you draw?

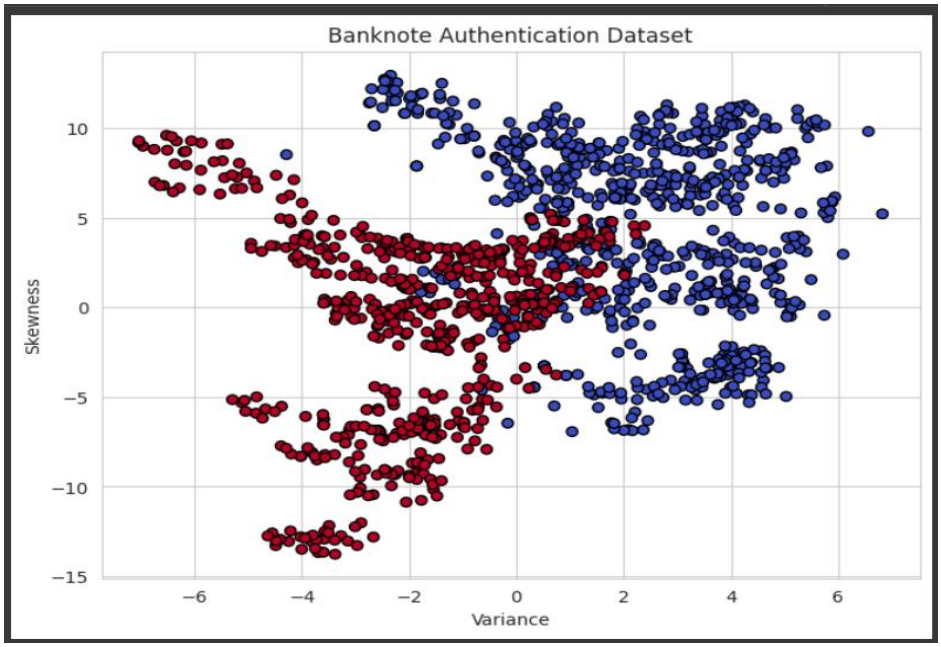
The Linear kernel performs poorly on the Moons dataset because its decision boundary is a straight line, which is inherently incapable of separating the non-linearly distributed half-moon clusters. As a result, many data points near the curved boundary are misclassified, leading to a low F1 score and accuracy. This clearly shows that linear separation is not suitable for this type of data structure, as the model fails to capture the underlying non-linear patterns in the dataset

2.Compare the decision boundaries of the RBF and Polynomial kernels. Which one seems to capture the shape of the data more naturally?

The RBF kernel effectively captures the nonlinear and curved structure of the Moons dataset, forming a smooth decision boundary that closely follows the separation between the two half-moon clusters. The Polynomial kernel also produces a nonlinear boundary, but it tends to be more complex and less smooth compared to the RBF kernel. Based on the classification report, the RBF kernel achieves the highest accuracy and F1 score, demonstrating its superior ability to model complex, nonlinear relationships and achieve near-perfect separation in this dataset

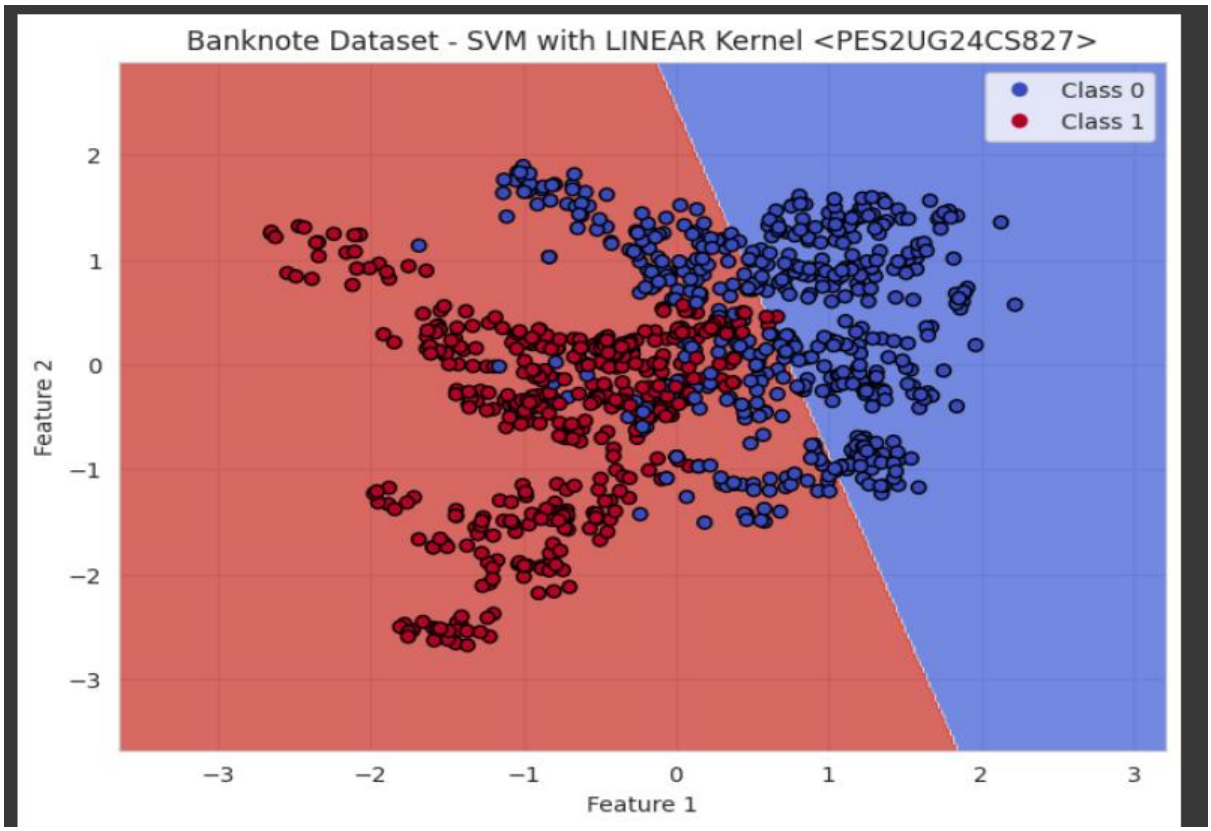
Part2

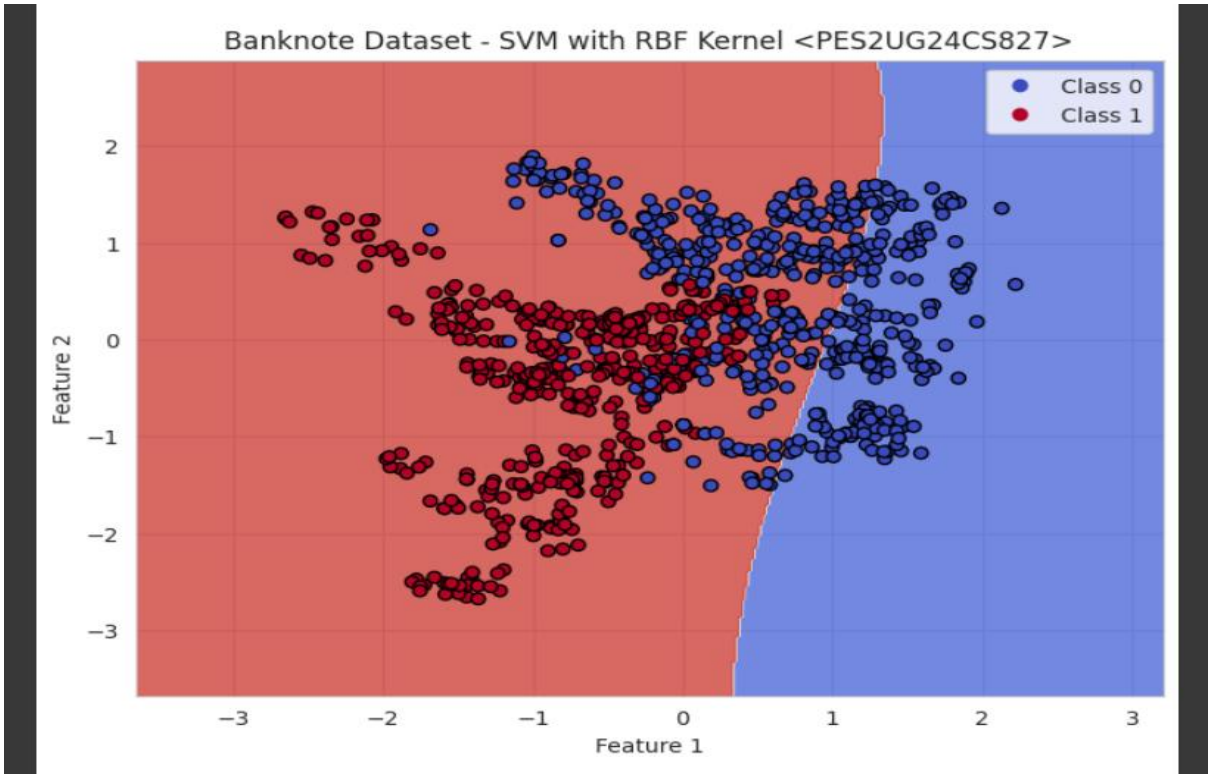
Dataset 2: Banknote Authentication

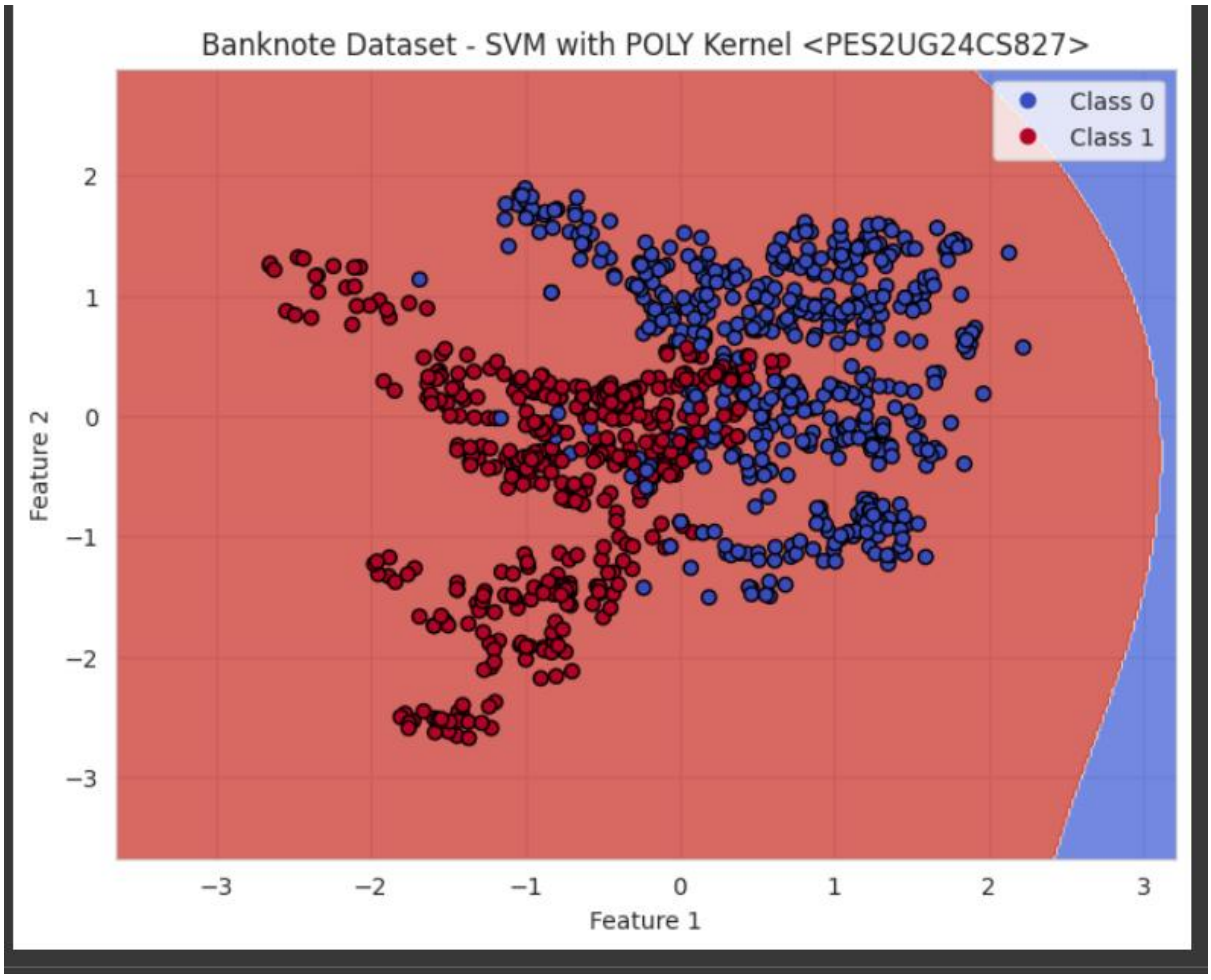


Step 2.2: Train and Evaluate SVM Kernels

```
SVM with LINEAR Kernel <PES2UG24CS827>
              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 <PES2UG24CS827>
              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 <PES2UG24CS827>
              precision    recall  f1-score   support

      Forged       0.96      0.81      0.88       229
     Genuine       0.80      0.96      0.88       183

    accuracy                           0.88       412
   macro avg       0.88      0.89      0.88       412
weighted avg       0.89      0.88      0.88       412

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




Banknote Dataset - SVM with RBF Kernel <PES2UG24CS827>
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 <PES2UG24CS827>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Banknote Dataset Questions

1.Which kernel was most effective for this dataset?

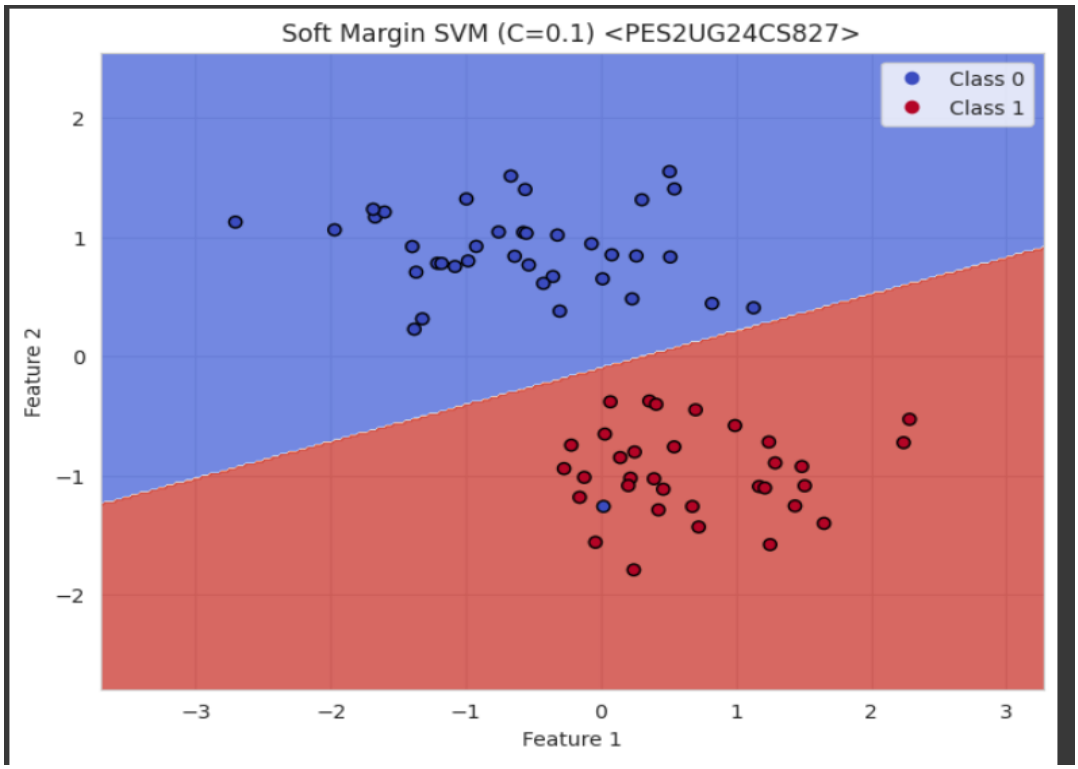
The Linear kernel proved to be the most effective for the Banknote Authentication dataset, achieving perfect or near-perfect performance as indicated by the classification report. This suggests that the dataset is linearly separable (or very close to it) when features such as variance and skewness are projected onto a 2D plane after scaling. The simplicity of the linear decision boundary is therefore sufficient and highly efficient, making more complex kernels unnecessary for this dataset

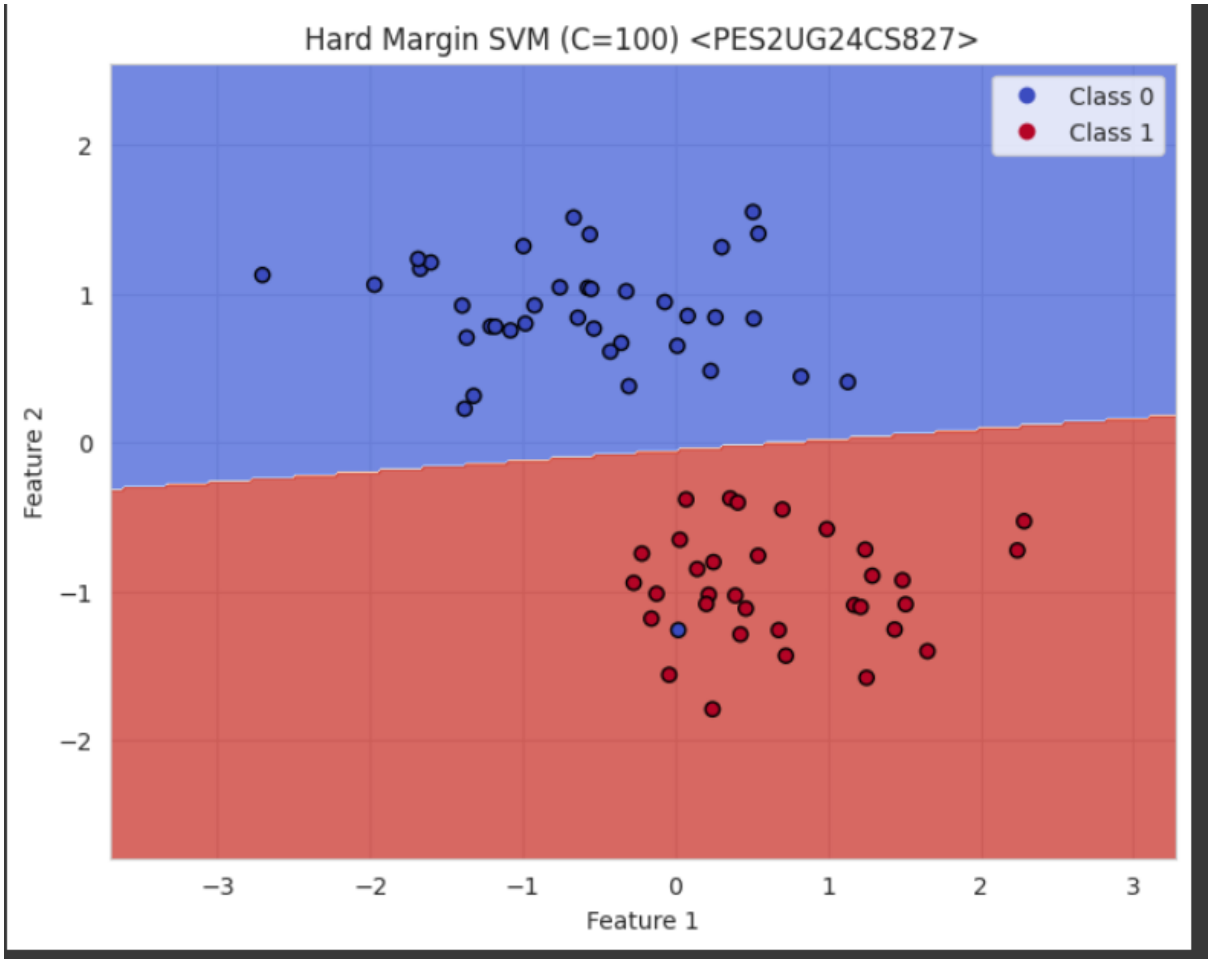
2.The Polynomial kernel shows lower performance here compared to the Moons dataset. What might be the reason for this?

The Polynomial kernel likely underperformed on the Banknote Authentication dataset because it introduced unnecessary complexity to a problem that is already largely linearly separable. Its higher-dimensional transformation results in a more curved and complex decision boundary, which—while still capable of good performance—offers no significant advantage over the Linear kernel. In fact, this added complexity may make the model more prone to minor overfitting, especially when the data can be effectively separated using a simple linear boundary.

PART 3

Understanding the Hard and Soft Margins





Hard vs. Soft Margin Questions

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 decision margin.
A smaller C value emphasizes margin maximization over perfect classification, allowing a few misclassifications on the training data.
This results in a broader and more generalized separation boundary, improving the model's ability to generalize to unseen data rather than overfitting to the training set.

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 SVM permits certain misclassifications, allowing data points to lie within the margin or even on the wrong side of the decision boundary. Its main objective is to find the widest possible margin that correctly separates the majority of the data, thereby enhancing generalization performance.
A small C parameter acts as a regularization factor, reducing the penalty for misclassified points. This tolerance toward errors makes the model more robust to outliers and noise, preventing overfitting to imperfect training 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 (large C value) tends to overfit the training data. The large penalty for misclassification forces the model to find a narrow hyperplane that perfectly separates all training points, including outliers. As a result, the decision boundary becomes highly specific to the training data, capturing even minor noise. This reduces the model's generalization ability and makes it less effective 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?

The Soft Margin SVM is more reliable for classifying new or unseen data because its wider margin promotes better generalization and reduces sensitivity to the exact positions of training points.
In real-world scenarios where data is often noisy or imperfect, starting with a smaller C value (i.e., a softer margin) provides a good balance between maximizing the margin and minimizing classification errors. This makes the model more robust and less prone to overfitting noise in the dataset.