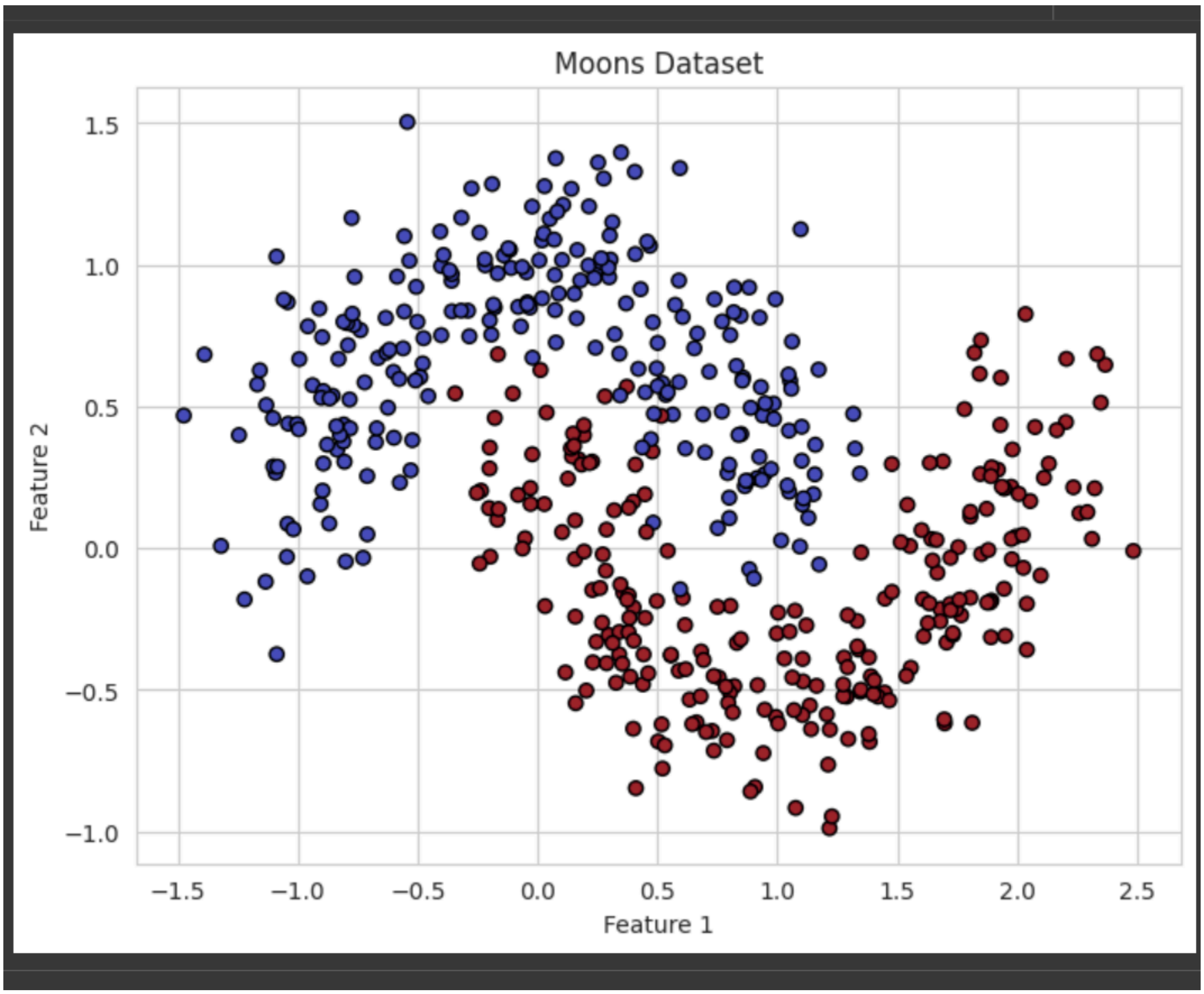
**Machine Learning lab**

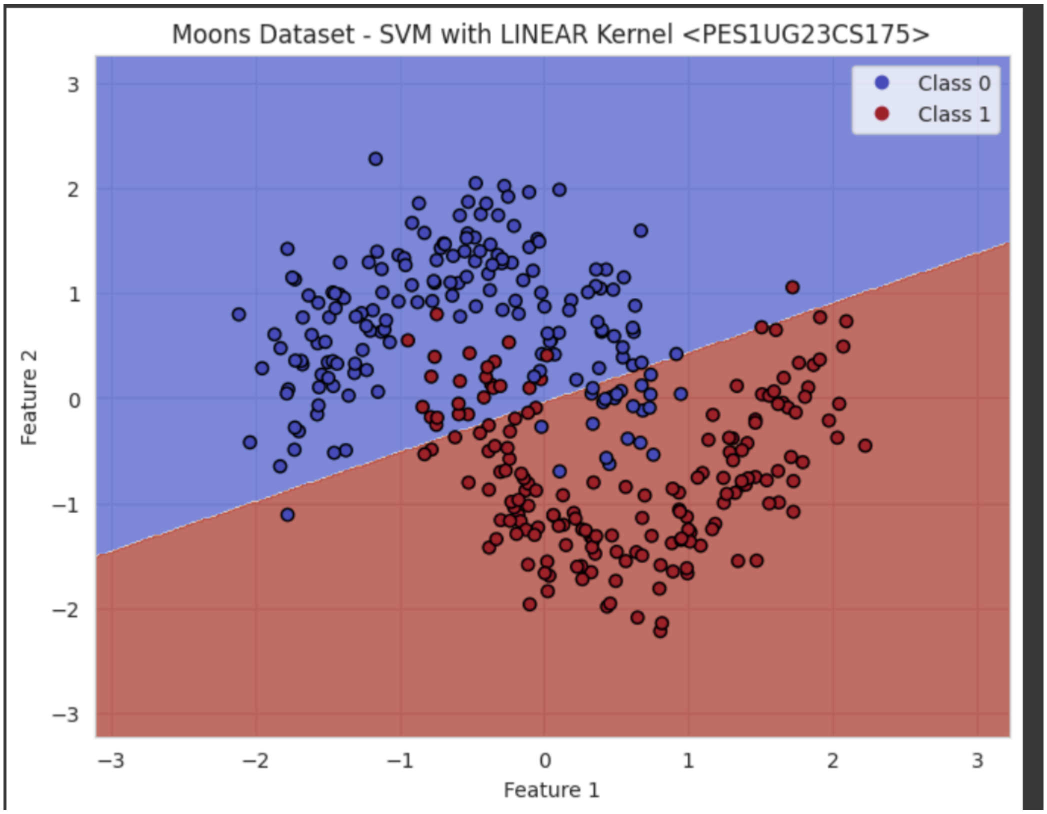
10-10-2025

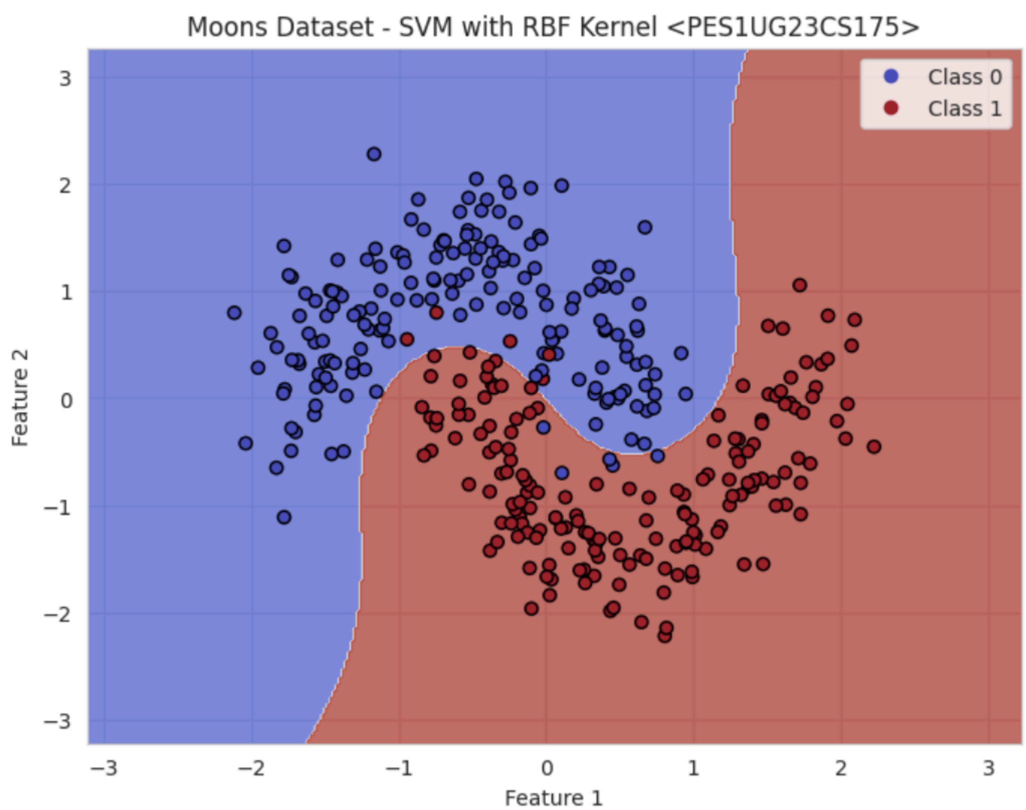
| Name: DHRUV THAKUR | SRN:PES2UG223CS 175 | Section C |
|---|---|---|

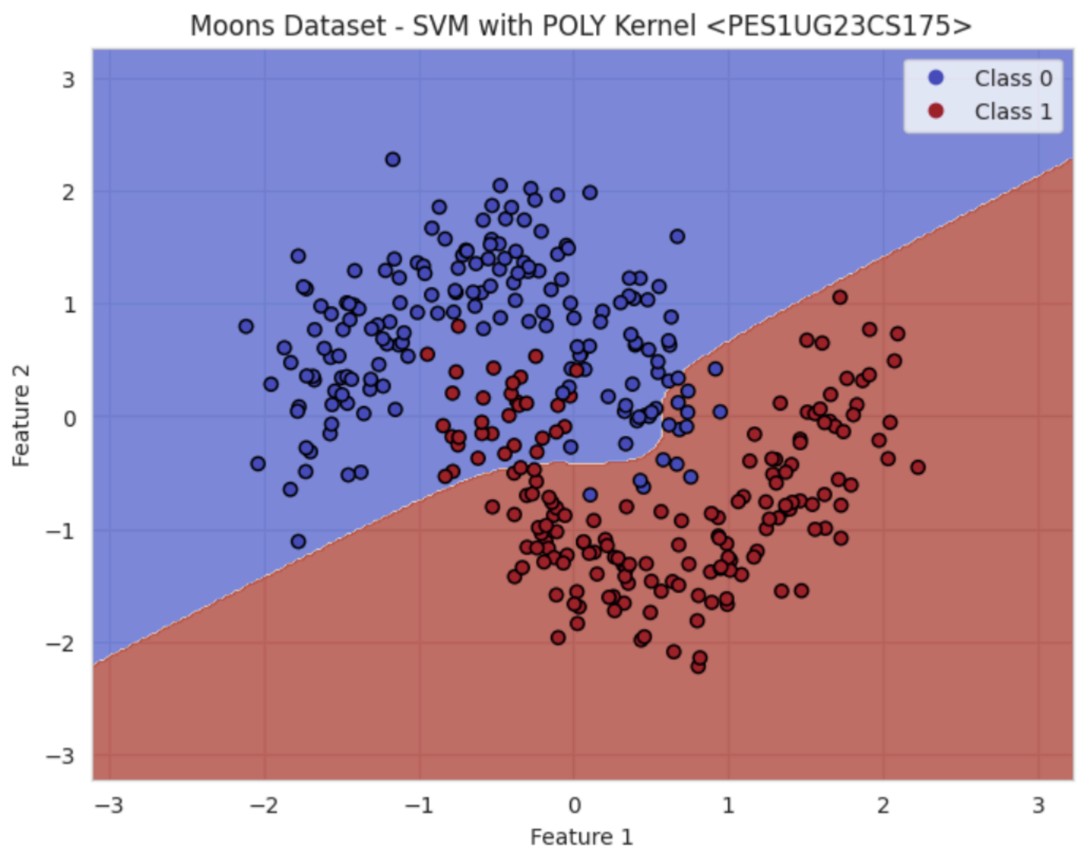
PART 1: The Moons Dataset



```
SVM with LINEAR Kernel <PES2UG23CS175>
              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 <PES2UG23CS175>
              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 <PES2UG23CS175>
              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 LINEAR Kernel <PES1UG23CS175>
Class 0
Class 1
Feature 1
Feature 2


Moons Dataset - SVM with RBF Kernel <PES1UG23CS175>
Class 0
Class 1
Feature 1
Feature 2




q1) The Linear Kernel performs poorly on the Moons dataset.

Metrics: It will likely show a lower accuracy (around 80-85%) and worse F1-scores compared to the non-linear kernels.

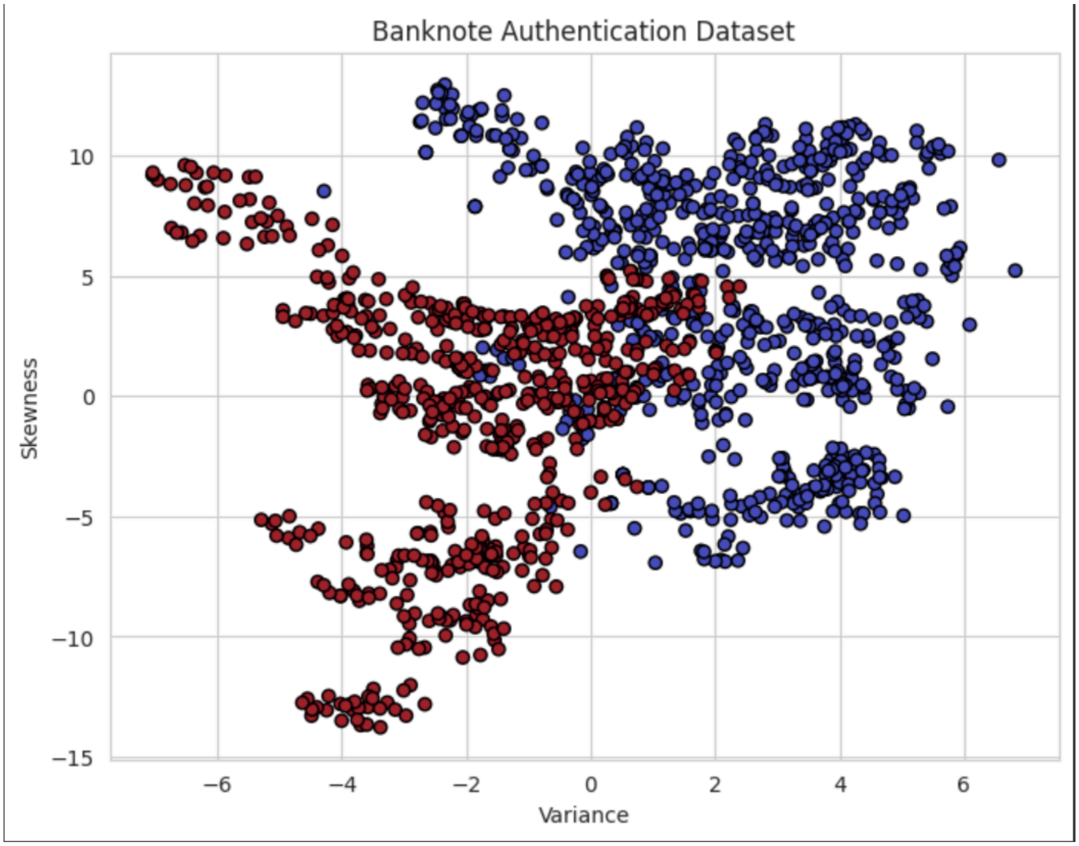
Visualization: The Linear Kernel creates a straight-line decision boundary that attempts to separate the two half-moons. Since the data is fundamentally non-linear, a single straight line cannot capture the true data structure, leading to many misclassifications near the center of the figure. This demonstrates that a linear model is not suitable for non-linearly separable data.

q2) The RBF (Radial Basis Function) Kernel generally captures the shape of the Moons dataset more naturally.

RBF Boundary: It produces a smooth, wave-like, or curved boundary that follows the natural separation between the two crescent shapes. It's highly effective at creating complex, non-linear boundaries.

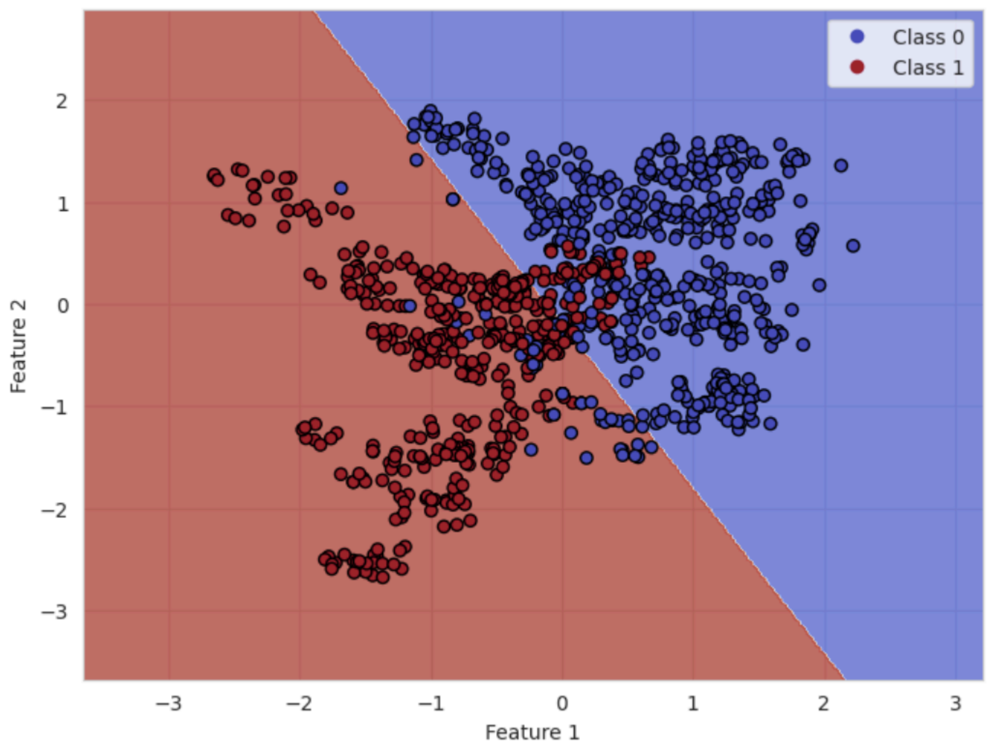
Polynomial Boundary (Degree 3 default): It also produces a non-linear, curved boundary, but it might be less smooth or more constrained than the RBF kernel. Depending on the polynomial degree, it might create overly complex or sometimes less intuitive shapes, whereas the RBF kernel excels at defining circular or spherical separation boundaries.

## Dataset 2: Banknote Authentication

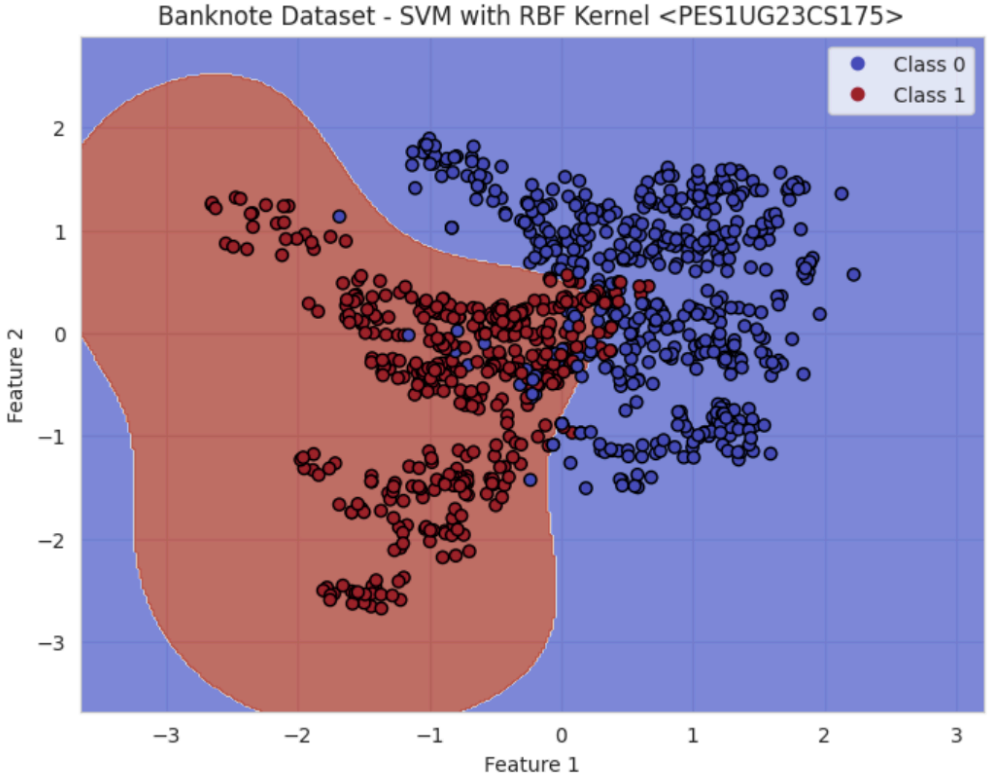


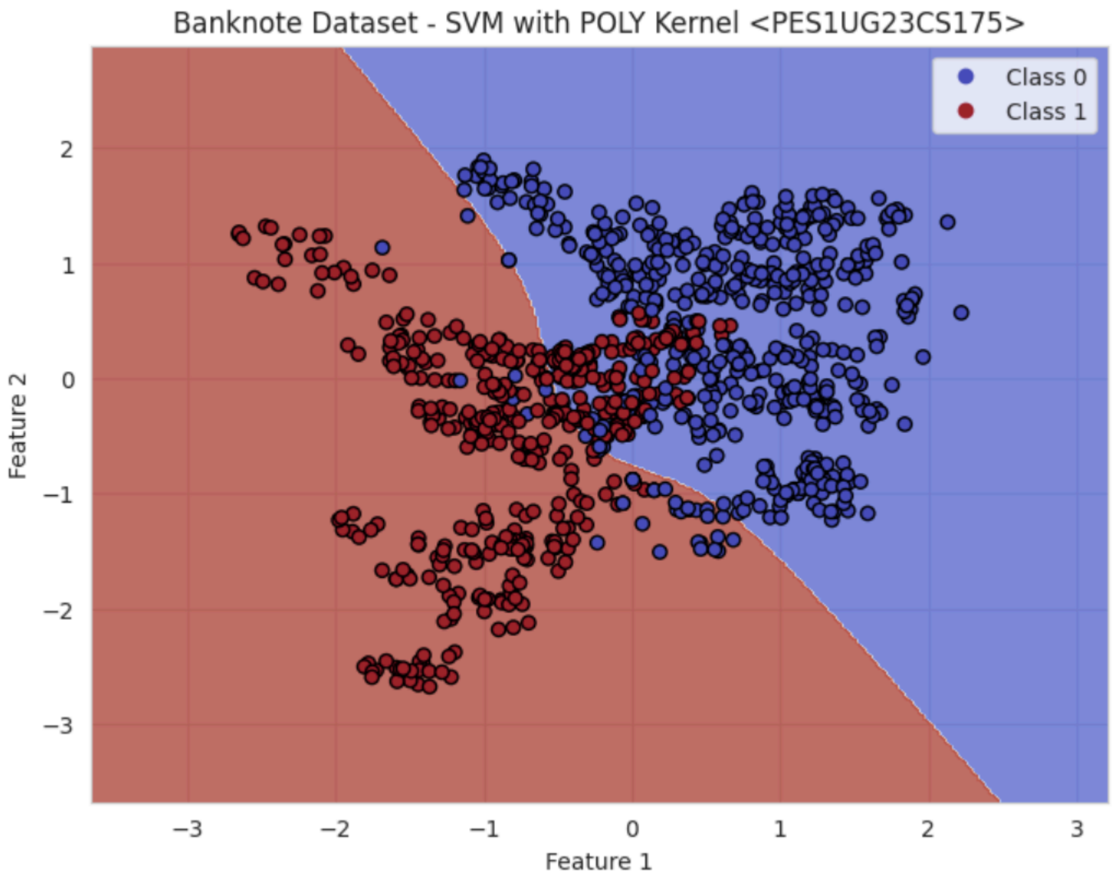
```
SVM with LINEAR Kernel <PES2UG23CS175>
              precision    recall  f1-score   support

      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 <PES2UG23CS175>
              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 <PES2UG23CS175>
              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 LINEAR Kernel <PES1UG23CS175>




Banknote Dataset - SVM with RBF Kernel <PES1UG23CS175>






q1) Based on the data visualization, the Banknote dataset is largely, but not perfectly, linearly separable. The two classes form distinct clusters that can be separated by a curve, or even a straight line with only a few errors.

The Linear Kernel or the RBF Kernel are likely the most effective.

The Linear Kernel often performs very well and offers the best balance of performance and simplicity when data is close to linearly separable.

The RBF Kernel might achieve slightly higher accuracy by creating a slightly curved boundary to better separate the few mixed points, potentially outperforming the linear kernel marginally, but it is computationally more complex.

q2) The main reason is the nature of the decision boundary required:

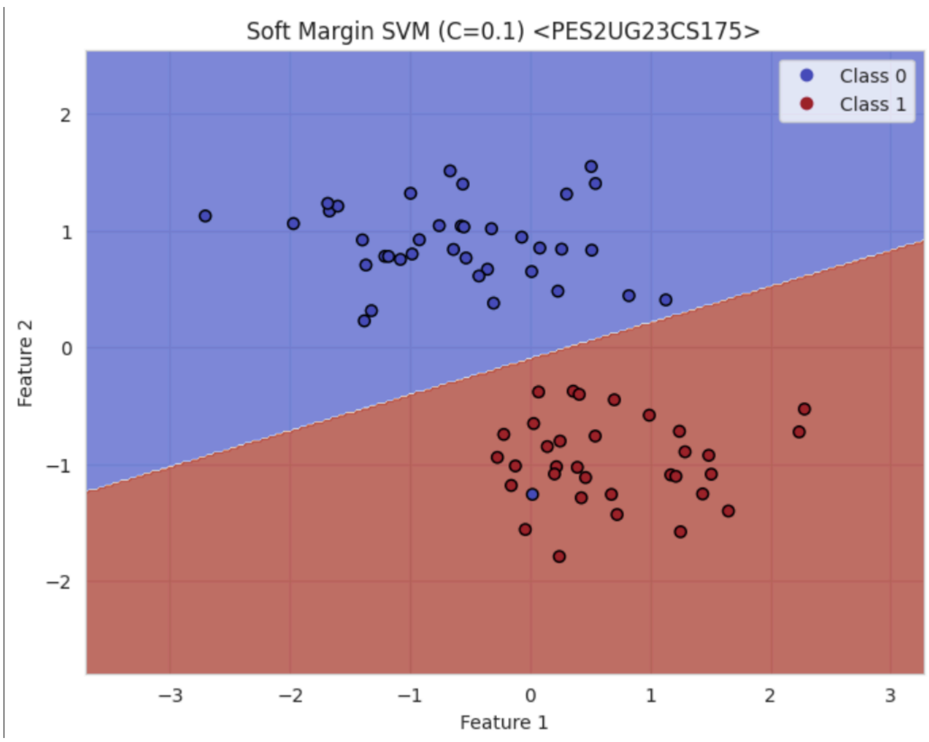
Moons Dataset: Requires a complex, S-shaped or highly curved boundary, which a (default degree 3) Polynomial kernel is well-suited to model, leading to high performance.

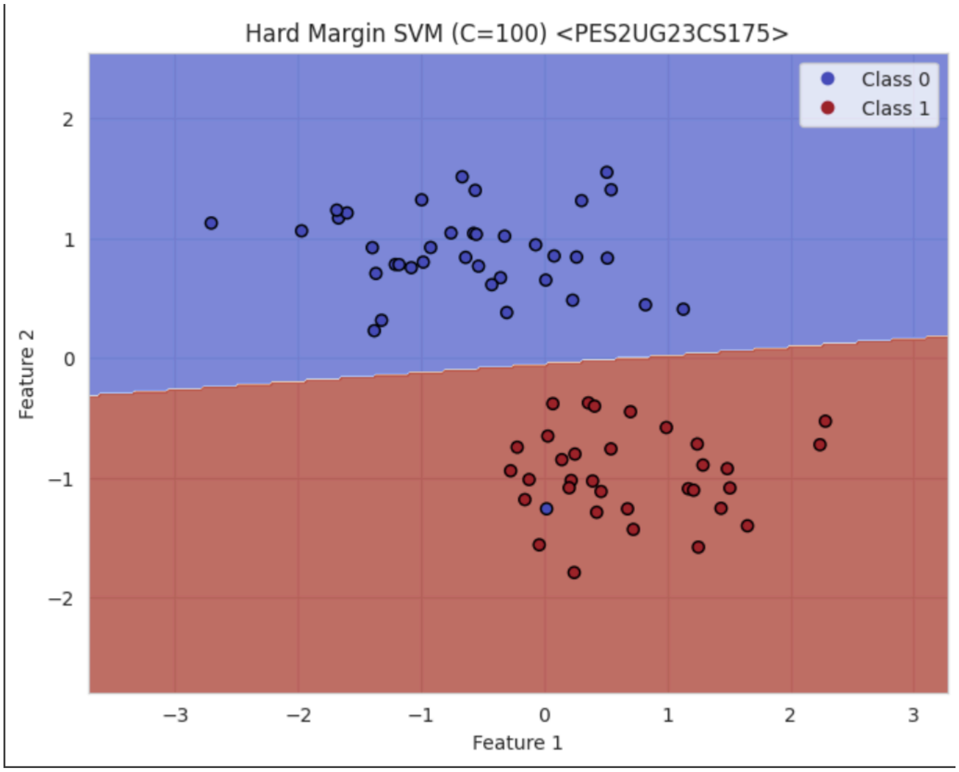
Banknote Dataset: Requires a simpler, potentially linear or mildly curved boundary to separate two distinct, compact clusters. The Polynomial kernel, by default, tries to fit a higher-degree curve (e.g., degree 3). This can lead to a decision boundary that is too complex or wiggly for the simple, structured data. This complexity can result in:

Overfitting: Creating a boundary that is sensitive to minor noise or outliers in the training data.

Suboptimal Generalization: A simple linear separation is usually the "cleanest" solution here, and the polynomial kernel's complexity can hurt its ability to generalize to new data.

## Part 3: Understanding the Hard and Soft Margins






q1)

the Soft Margin (C=0.1) model produces a wider margin. A small value of C means a low penalty for misclassified points, allowing the model to ignore outliers and focus on finding a simpler, wider hyperplane that generalizes better to the main clusters.

q2)

The Soft Margin SVM allows these mistakes because of the regularization parameter C.

The SVM's objective is to maximize the margin while simultaneously minimizing the training errors.

With a small C, the penalty for violating the margin or misclassifying points is low. This allows the model to prioritize a wider, simpler margin over perfectly classifying every training point, especially outliers.

The primary goal is to find the hyperplane with the best generalizability, even if it means tolerating a few training errors.

q3)

The Hard Margin (C=100) model is more likely to be overfitting.

A large C imposes a high penalty on misclassification, forcing the model to find a complex boundary (in this linear case, a very specific and often narrower one) that correctly separates all training points, including the added outliers.

By being overly sensitive to every single training point, the model captures the noise and specifics of the training set (the outliers), which compromises its ability to generalize well to new, unseen data.

q4) I would trust the Soft Margin (C=0.1) model more to classify a new, unseen data point correctly.

Reasoning: The Soft Margin model focuses on the general structure of the data by ignoring minor perturbations (like outliers). Its wider margin provides a larger safety area, which makes it more robust to the natural variation and noise found in real-world test data.