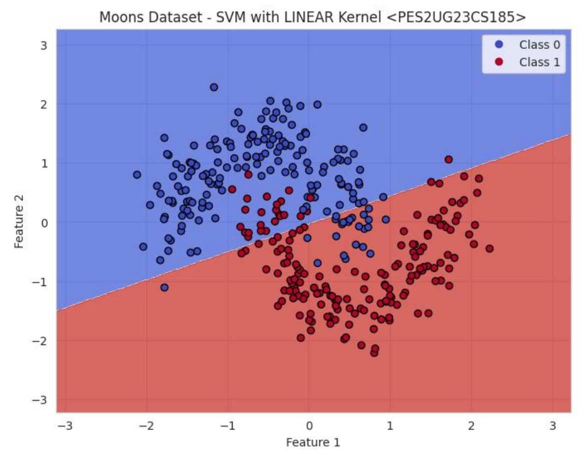
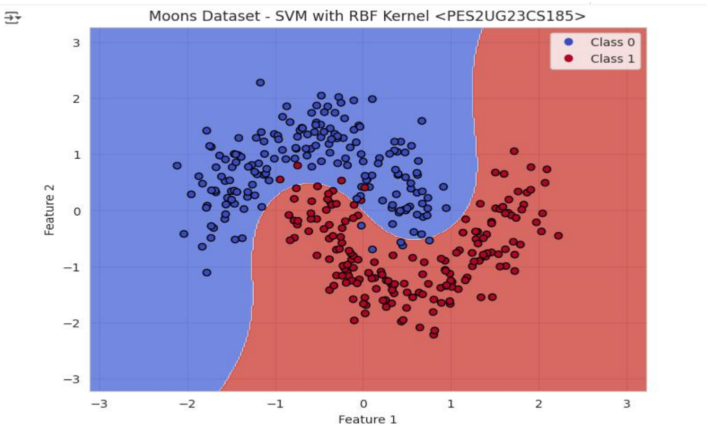
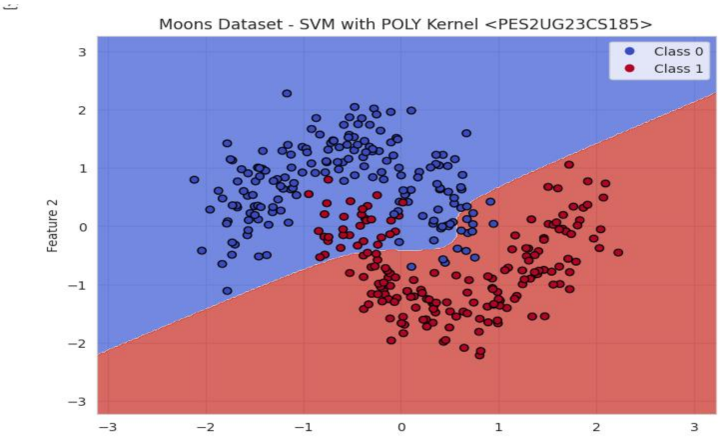
NAME: DRISHTI GOLCHHA
SRN: PES2UG23CS185
SECTION: 5C

Moons Dataset

```
SVM with LINEAR Kernel <PES2UG23CS185>
              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 <PES2UG23CS185>
              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 <PES2UG23CS185>
              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

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







Q1) Based on the metrics and the visualizations, what inferences about the performance of the Linear Kernel can you draw?

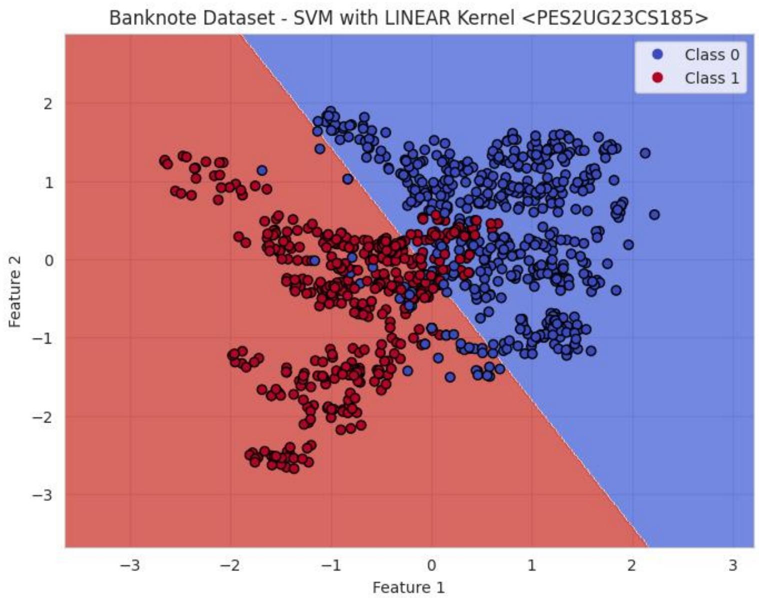
ANSWER) The Linear Kernel is not well-suited for datasets with non-linear patterns like Moons. It provides a rough, oversimplified boundary that cannot capture complex structures.

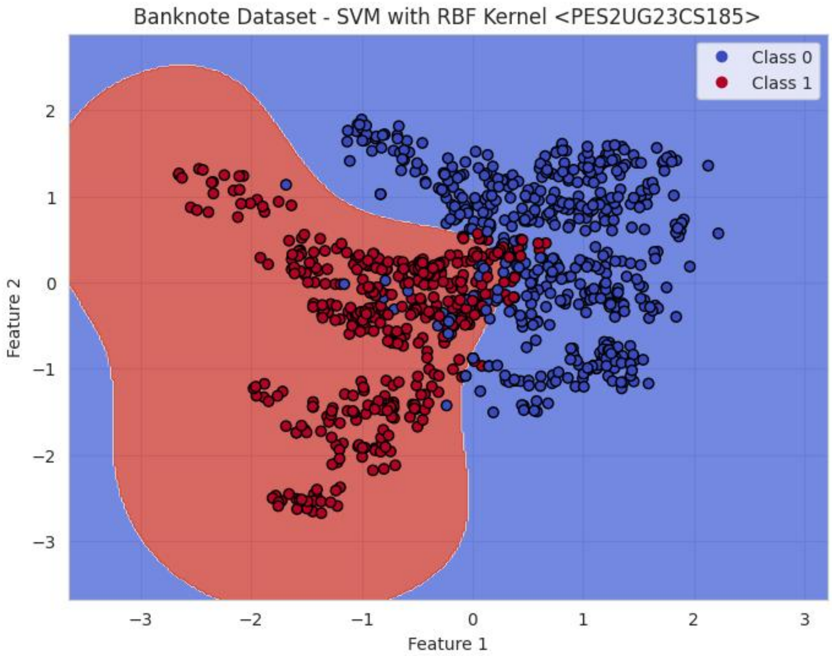
Q2) Compare the decision boundaries of the RBF and Polynomial kernels. Which one seems to capture the shape of the data more naturally?

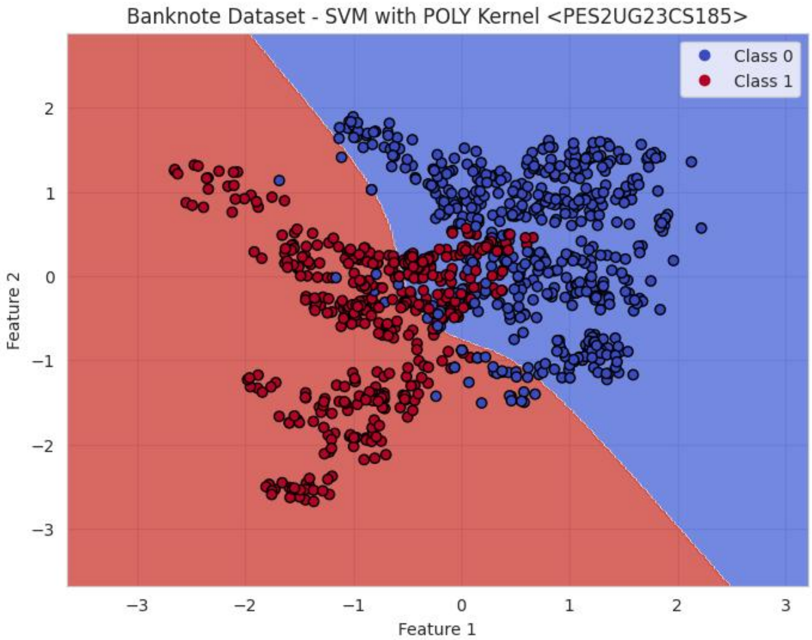
ANSWER) **RBF Kernel** captures the natural, smooth shape of the Moons dataset more effectively and cleanly whereas **Polynomial Kernel** works better than Linear, but may be slightly less natural or more irregular than RBF.

Banknote Authentication

```
SVM with LINEAR Kernel <PES2UG23CS185>
              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 <PES2UG23CS185>
              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 <PES2UG23CS185>
              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

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



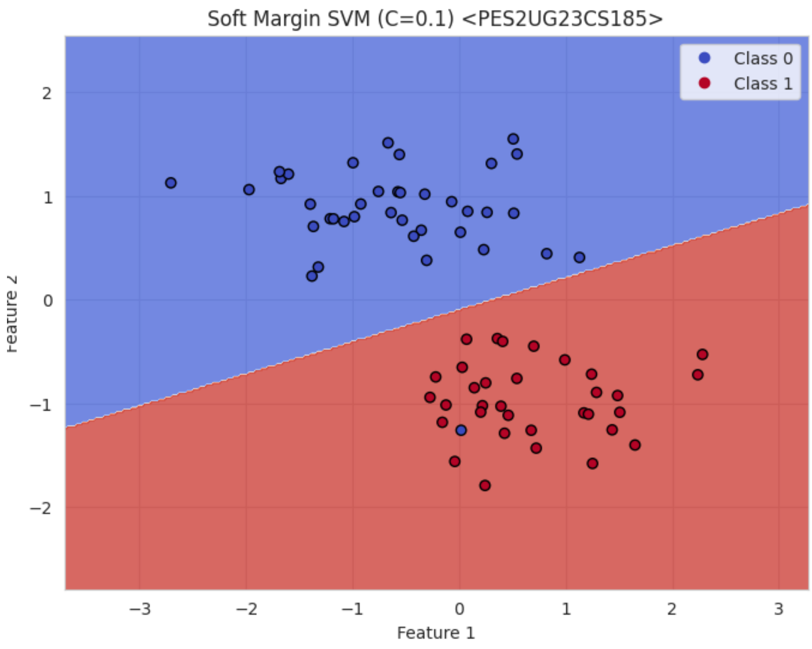


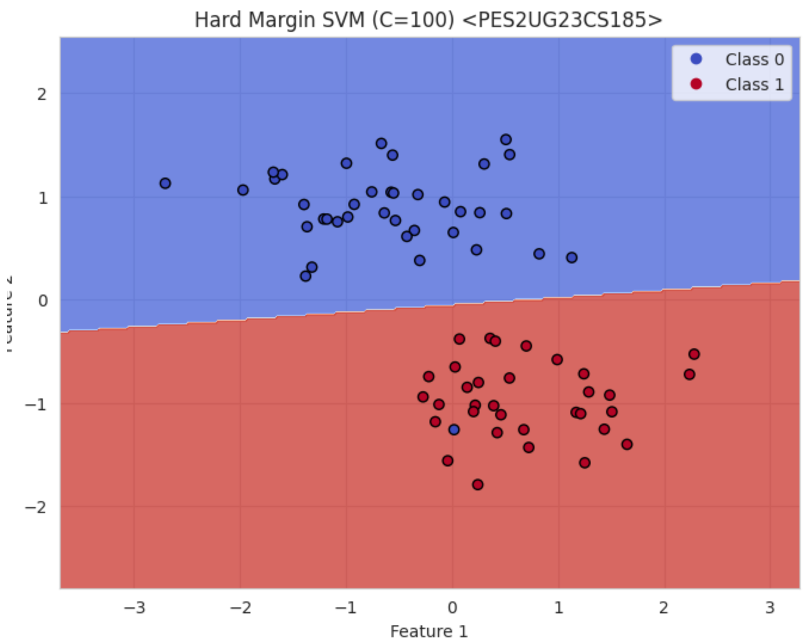


Q1) In this case, which kernel appears to be the most effective?
Answer) The RBF kernel is usually the most effective for this dataset.

Q2) The Polynomial kernel shows lower performance here compared to the Moons dataset. What might be the reason for this?
Answer) Polynomial kernel is less flexible than RBF for irregular, real-world non-linear data**,** unlike synthetic datasets like Moons where polynomial curves can match the shape well.

Understanding the Hard and Soft Margins





Q1) Compare the two plots. Which model, the "Soft Margin" (C=0.1) or the "Hard Margin" (C=100), produces a wider margin?

Answer) The Soft Margin (C=0.1) produces a wider margin**.**

Q2) 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?
Answer) SVM allows some “mistakes” because Soft margin balances margin width and misclassification tolerance to improve generalization. The primary goal is to maximize the margin and generalize well to new data, rather than perfectly classifying all training points.

Q3) Which of these two models do you think is more likely to be overfitting to the training data? Explain your reasoning.
Answer) The Hard Margin (C=100) is more likely to overfit. Reason: It tries to perfectly classify every training point, including outliers.

Q4) 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?
Answer) Soft Margin (C=0.1) is more trustworthy for new, unseen points. **Reason:** It focuses on a smooth, generalizable boundary .It avoids overfitting to noise in the training data. Start with a low C value (soft margin). Most real-world data is noisy, and a low C allows the model to ignore small anomalies and produce a more generalized decision boundary