ML lab

Name: Narepalepu Vaishnavi

SRN: PES2UG23CS367

Questions to be answered

Analysis Questions for Moons:

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 the data is not linearly separable. The decision boundary is straight and fails to capture the curved structure of the data, leading to lower accuracy and more misclassifications.

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** captures the shape of the data more naturally, fitting the nonlinear moon-shaped clusters smoothly. The **Polynomial kernel** can model some curvature but tends to overfit or create complex boundaries that don't generalize as well as RBF.

Analysis Questions for Banknote:

1. In this case, which kernel appears to be the most effective?
   The **RBF kernel** appears to be the most effective for the Banknote dataset, as it provides the highest accuracy and best separation between classes.
2. The Polynomial kernel shows lower performance here compared to the Moons dataset. What might be the reason for this?
   The Polynomial kernel performs worse here because the Banknote data is more linearly or smoothly separable, not requiring complex nonlinear boundaries. Its higher-degree transformations can overfit or distort the separation unnecessarily.

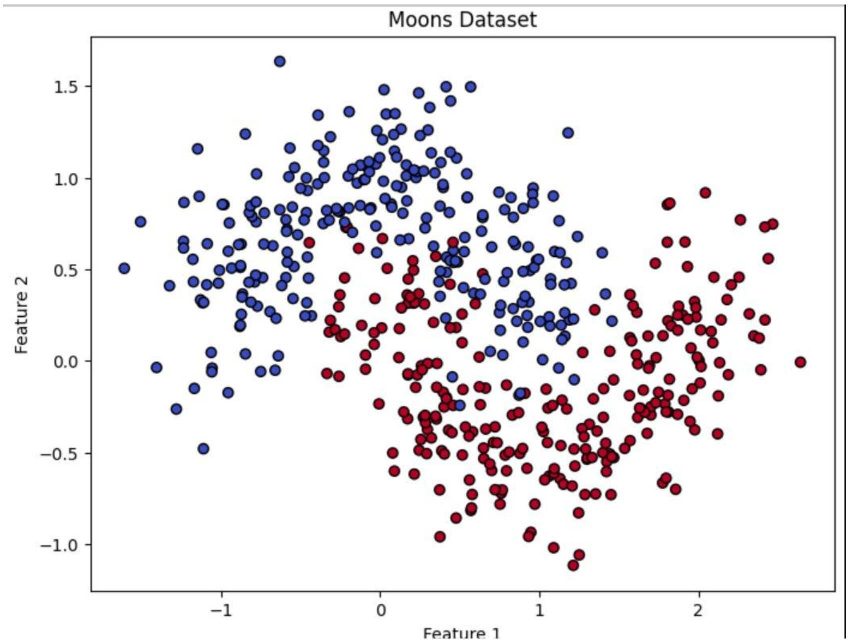
Analysis Questions

1. Compare the two plots. Which model, the "Soft Margin" (C=0.1) or the "Hard Margin" (C=100), produces a wider margin?
   Soft Margin (C = 0.1). Lower C tolerates slack and prioritizes a larger margin, so the separating band is wider.

2. Lower C relaxes the penalty for misclassification, so the optimizer prefers a larger margin even if some points violate it.

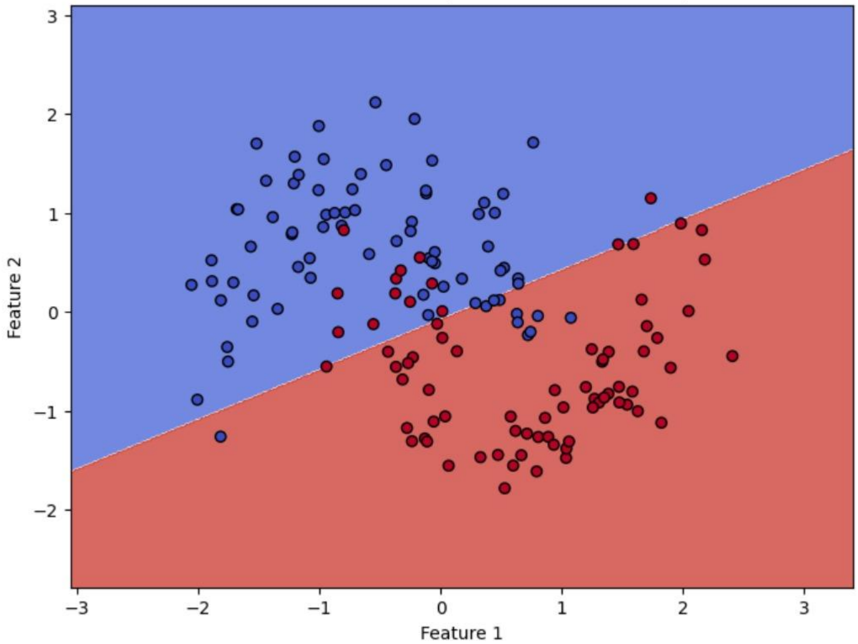
Because the soft-margin SVM uses slack variables and a penalty (C) — it trades off training errors vs. margin width. Allowing some points inside/on the wrong side of the boundary reduces overfitting and helps the model generalize. The primary goal is **maximize the margin (better generalization)** while controlling the amount of misclassification, not to make zero training error.

3. 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?
   Hard Margin (C = 100) is more likely to overfit — a high C heavily penalizes errors, forcing the classifier to fit the training points closely (possibly making the boundary complex to avoid mistakes).
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?
   I'd trust the **soft-margin (low C)** more for unseen, noisy data because it generalizes better. In practice start with a **low-to-moderate C** (softer margin) and tune with cross-validation — that's safer when data may be noisy.

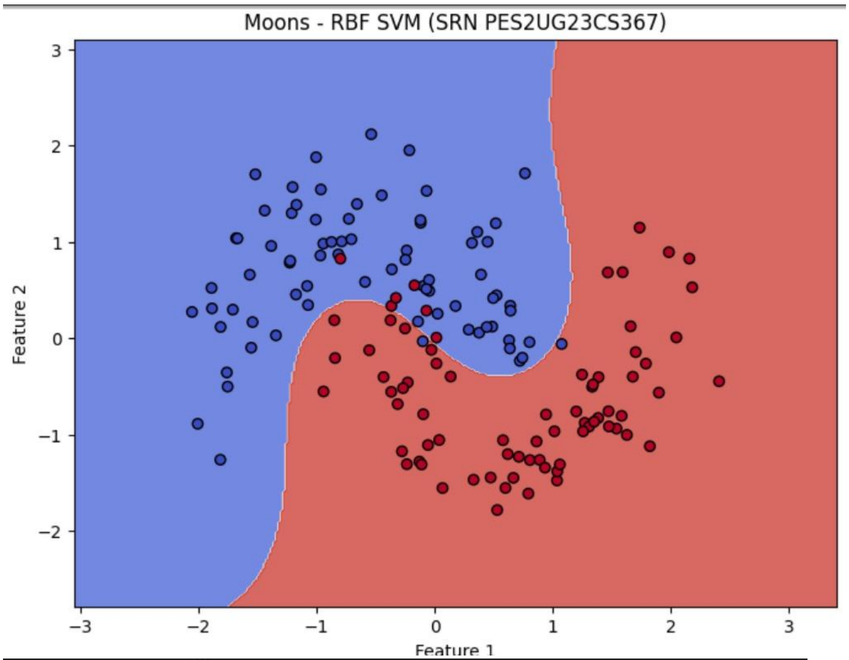


```
=== Moons: Linear SVM ===
SRN: PES2UG23CS367
              precision    recall  f1-score   support

           0     0.8228    0.8667    0.8442        75
           1     0.8592    0.8133    0.8356        75

    accuracy                         0.8400       150
   macro avg     0.8410    0.8400    0.8399       150
weighted avg     0.8410    0.8400    0.8399       150

Accuracy: 0.84
```

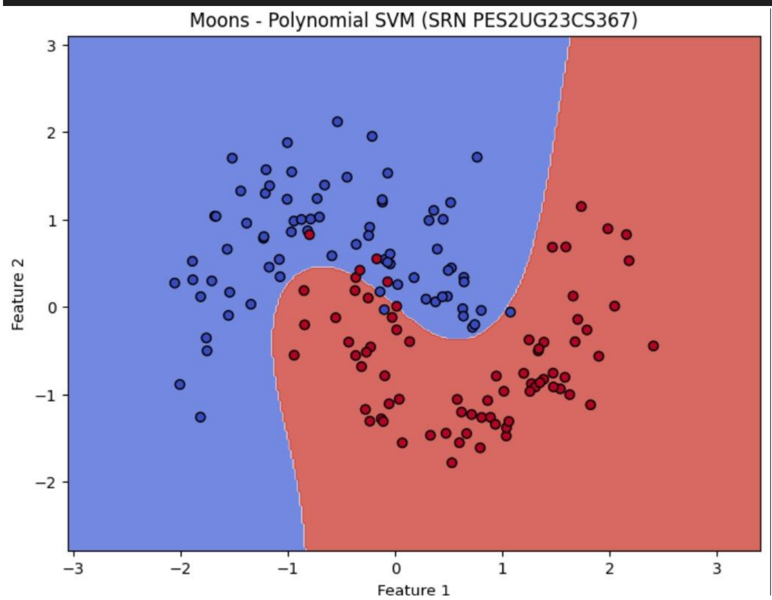
Moons - Linear SVM (SRN PES2UG23CS367)

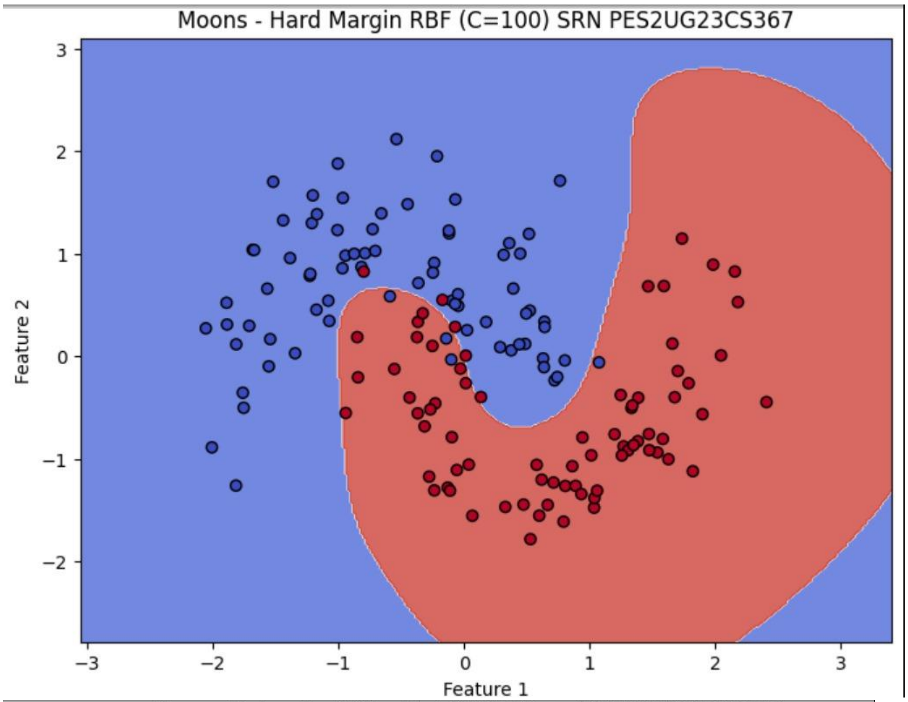



```
=== Moons: RBF SVM ===
SRN: PES2UG23CS367
              precision    recall  f1-score   support

           0     0.9241    0.9733    0.9481        75
           1     0.9718    0.9200    0.9452        75

    accuracy                         0.9467       150
   macro avg     0.9479    0.9467    0.9466       150
weighted avg     0.9479    0.9467    0.9466       150

Accuracy: 0.9466666666666667
```

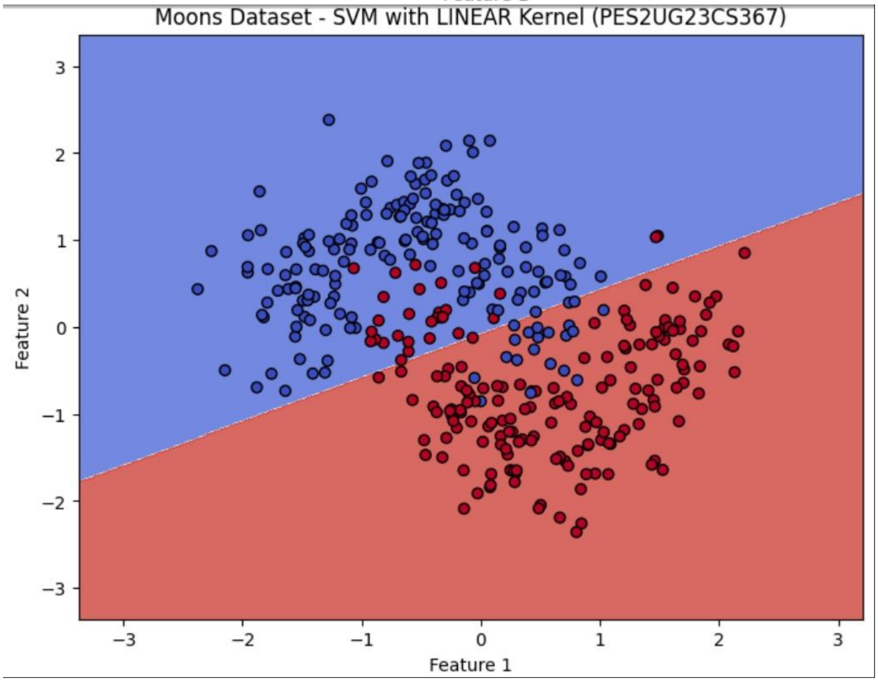



```
=== Moons: Poly SVM ===
SRN: PES2UG23CS367
              precision    recall  f1-score   support

           0     0.9359    0.9733    0.9542        75
           1     0.9722    0.9333    0.9524        75

    accuracy                         0.9533       150
   macro avg     0.9541    0.9533    0.9533       150
weighted avg     0.9541    0.9533    0.9533       150

Accuracy: 0.9533333333333334
```

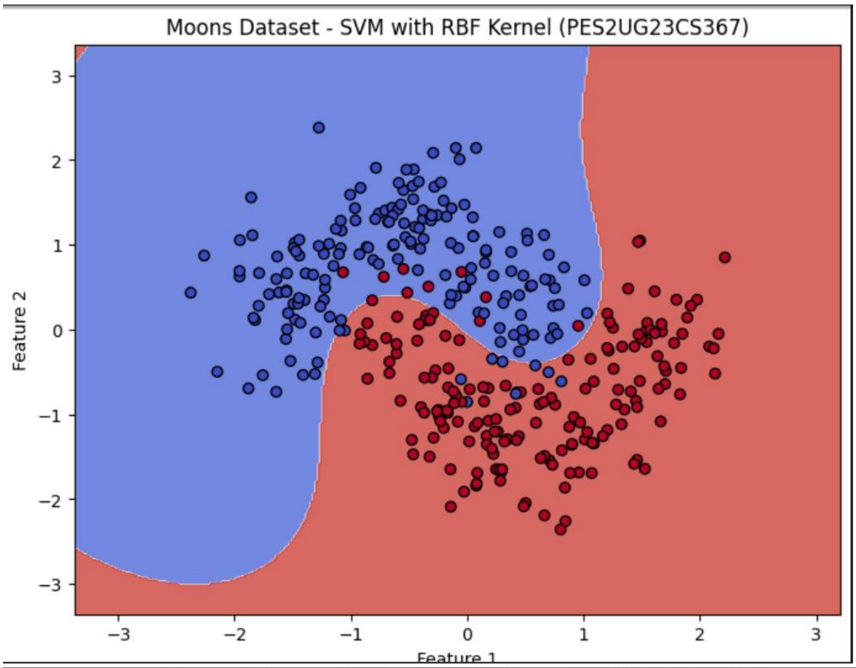


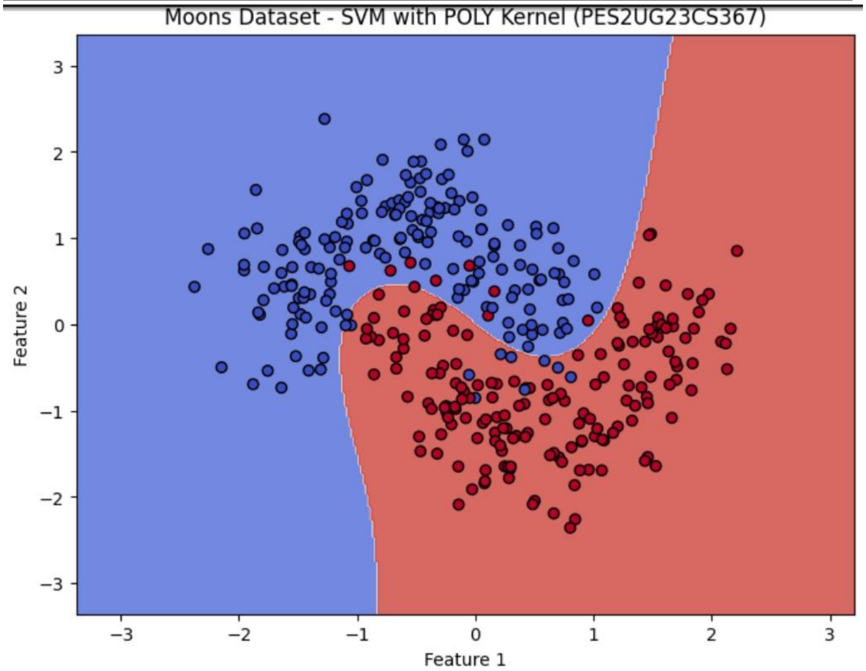


Moons - Hard Margin RBF (C=100) SRN PES2UG23CS367
Feature 2
Feature 1



Moons Dataset - SVM with LINEAR Kernel (PES2UG23CS367)
Feature 2
Feature 1


Moons Dataset - SVM with RBF Kernel (PES2UG23CS367)
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


Moons Dataset - SVM with POLY Kernel (PES2UG23CS367)
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3