MLlab

Name: Nikitha Devaraj

SRN: PES2UG23CS388

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?
   1. Linear Kernel Performance:
   The Linear Kernel performs well only if the data is **linearly separable**.
   Metrics (like accuracy and F1-score) are usually **lower** than RBF or Polynomial kernels when the data is non-linear.
   Visualizations show **straight decision boundaries**, often **misclassifying curved regions** of data.
2. Compare the decision boundaries of the RBF and Polynomial kernels. Which one seems to capture the shape of the data more naturally?
   RBF vs Polynomial Kernels:
   RBF Kernel captures the **shape of the data more naturally**, forming smooth, flexible boundaries that fit complex, non-linear patterns.
   Polynomial Kernel can model curves but may **overfit** or produce **rigid, wavy boundaries**.
   Inference: RBF kernel provides a better, more natural fit for most non-linear datasets.

   ### Analysis Questions for Banknote:
   1. In this case, which kernel appears to be the most effective?

   For the **Banknote dataset**, the **Linear Kernel** usually performs best because the data is **nearly linearly separable**.
   It gives **high accuracy** with simpler, faster computation compared to RBF or Polynomial kernels.
   2. The Polynomial kernel shows lower performance here compared to the Moons dataset. What might be the reason for this?
      The **Banknote dataset** has **features that relate linearly**, so adding polynomial terms doesn't help.
      The **Polynomial kernel overfits** or adds unnecessary complexity, reducing generalization and performance compared to the simpler linear boundary.

   ### 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?
      Wider margin:

      Soft Margin (C = 0.1) produces the wider margin.

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

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?
   SVM uses **slack variables** to allow violations; with a low C those violations cost little.
   The **primary goal** is **maximize the margin** (reduce model complexity) to improve generalization, not to force zero training errors.

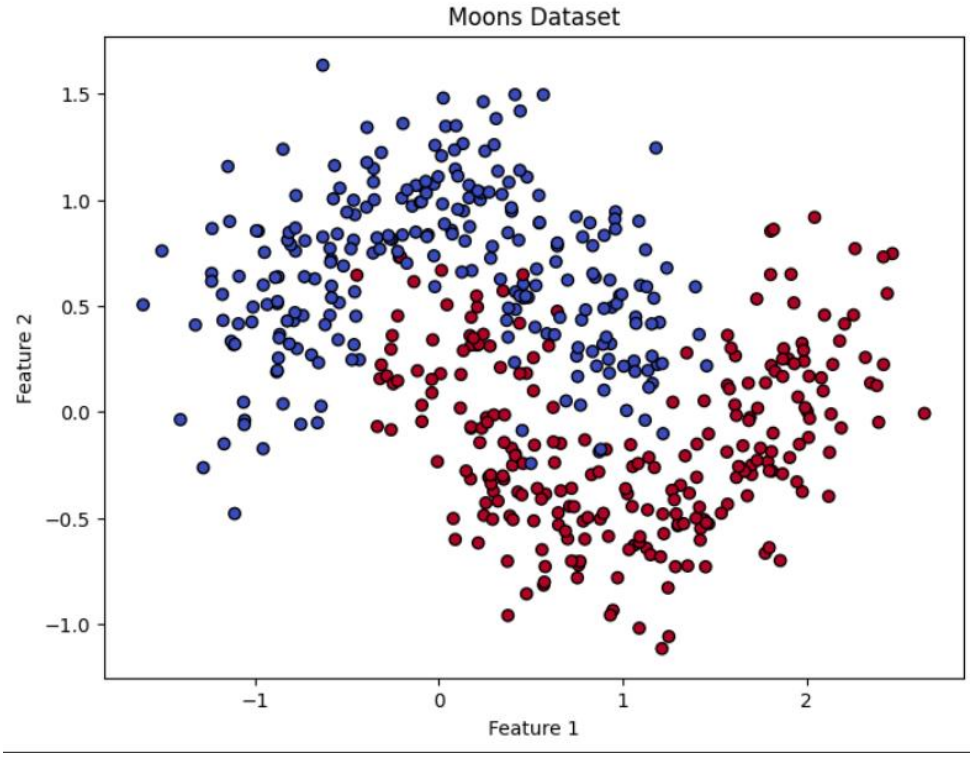
4. Imagine you receive a new, unseen data point. Which model do you trust more to classify it correctly? Why? In a real-world scenario where data is often noisy, which value of C (low or high) would you generally prefer to start with?

I'd trust the **Soft Margin (low C)** more for a new unseen point — it generalizes better.

In noisy, real-world settings, **start with a low C** (more regularization / wider margin) and then tune upward if you underfit.

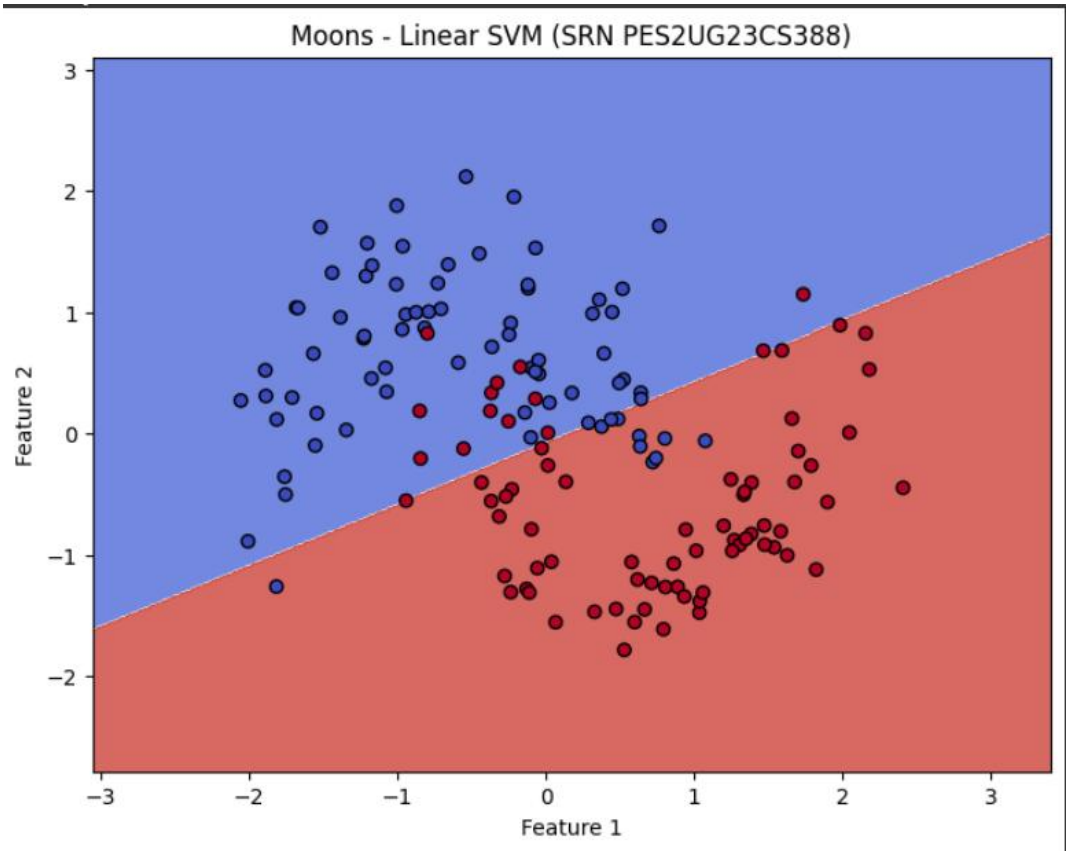
SCREENSHOTS :

1. Moons Dataset :



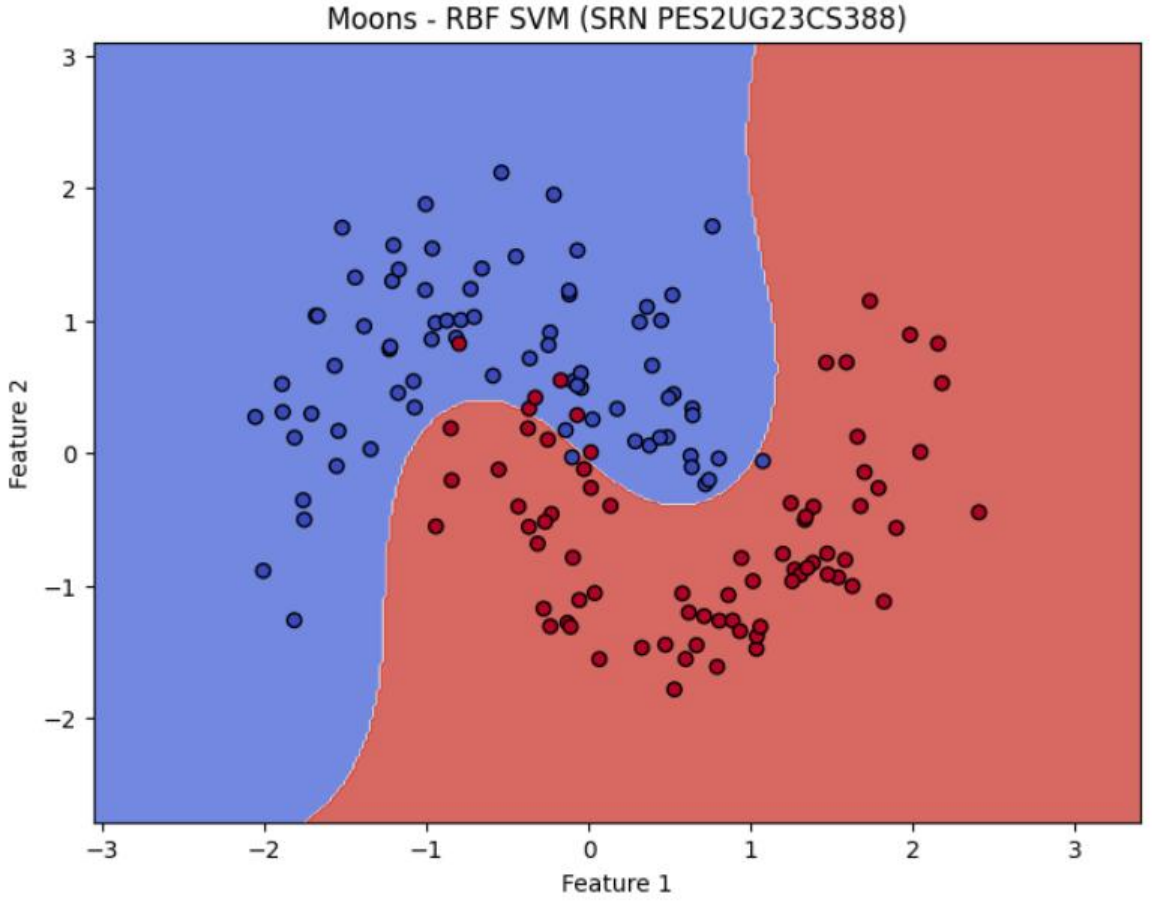
(a) SVM with Linear Kernel

```
=== Moons: Linear SVM ===
SRN: PES2UG23CS388
              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
```



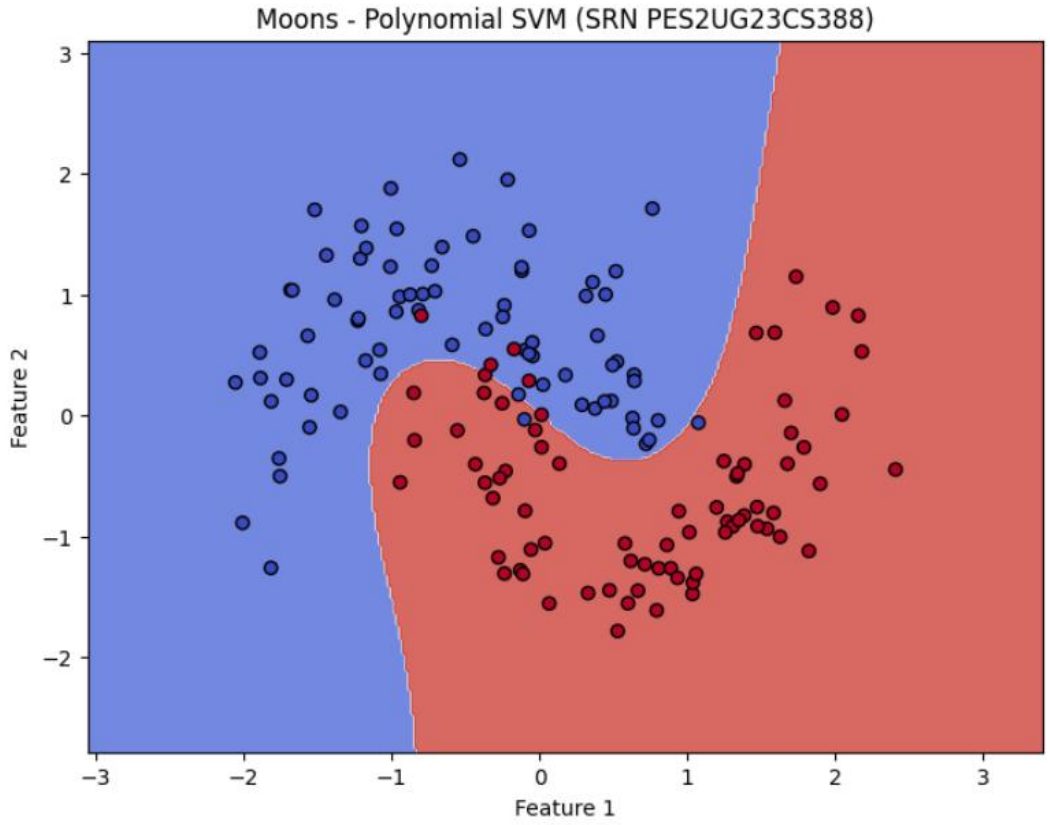
(b) SVM with RBF Kernel :

```
=== Moons: RBF SVM ===
SRN: PES2UG23CS388
              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
```



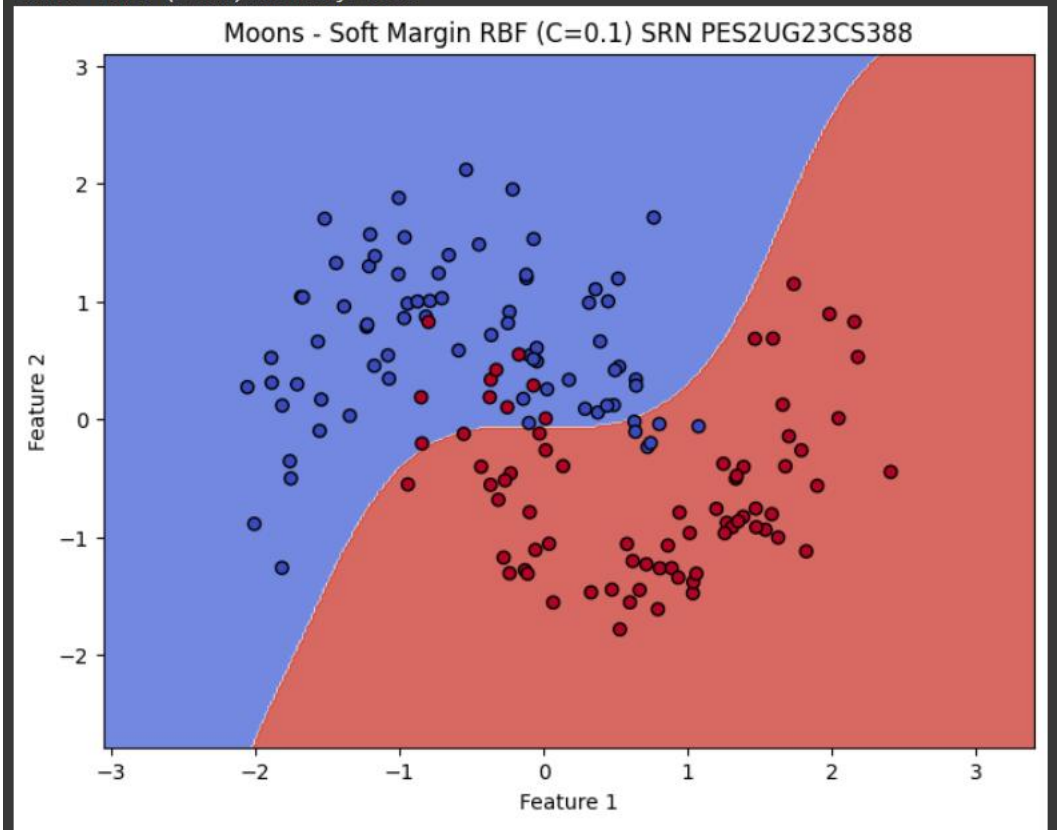
(c) SVM with POLY kernel :

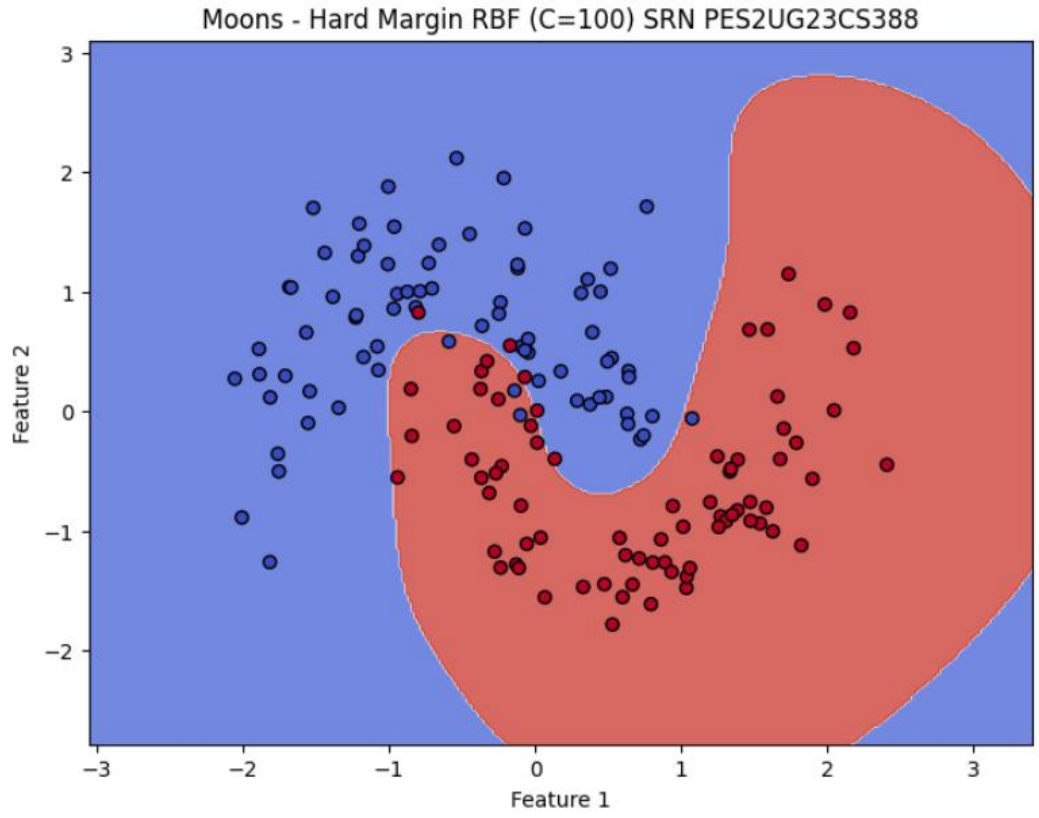
```
=== Moons: Poly SVM ===
SRN: PES2UG23CS388
              precision    recall  f1-score   support

           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
```

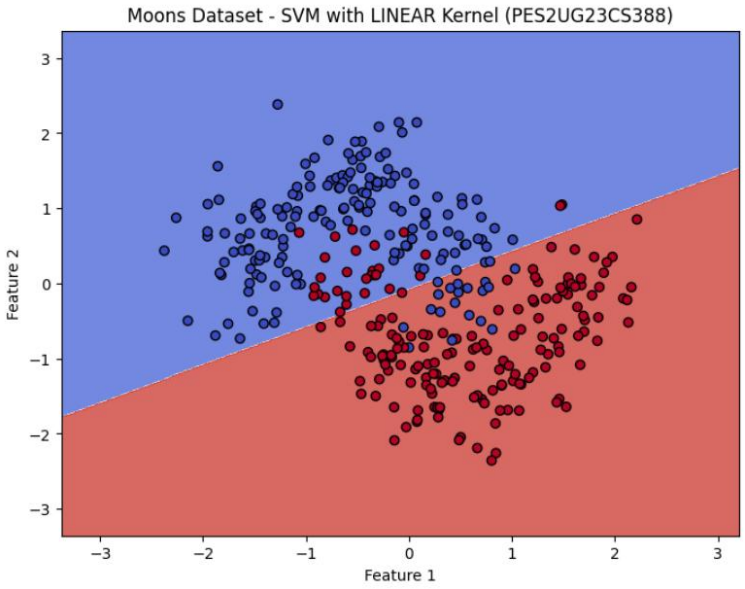


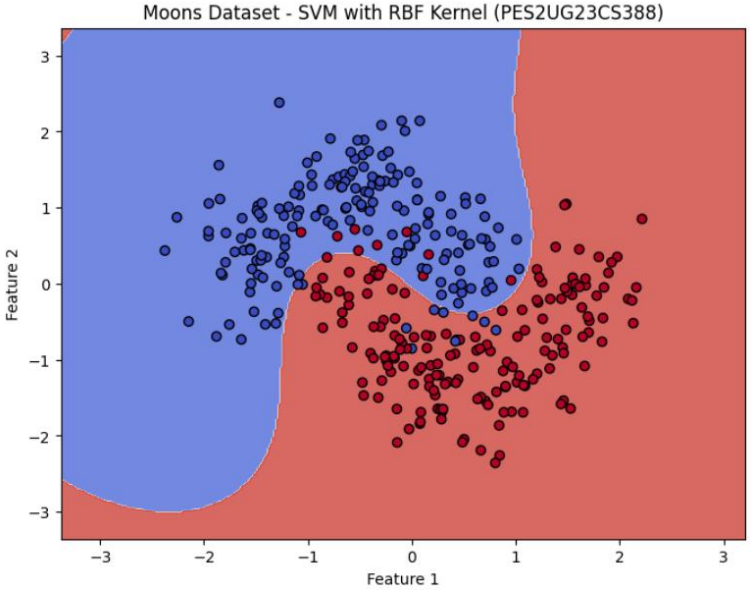
2. Margin Analysis :

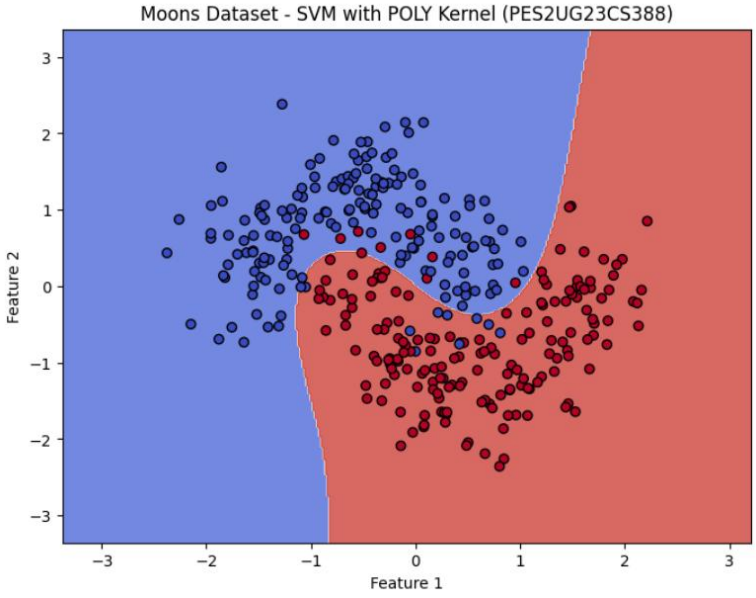
```
Moons - Soft (C=0.1) Accuracy: 0.8933333333333333
Moons - Hard (C=100) Accuracy: 0.94
```






Moons Dataset - SVM with LINEAR Kernel (PES2UG23CS388)
Feature 1
Feature 2


Moons Dataset - SVM with RBF Kernel (PES2UG23CS388)
Feature 1
Feature 2


Moons Dataset - SVM with POLY Kernel (PES2UG23CS388)
Feature 1
Feature 2