ML Lab

Name: Nikitha P

SRN: PES2UG23CS389

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 **adequately but is not ideal** for the Moons dataset.

- **Metrics:** The accuracy is (84.00%), with precision and recall values around 82-86%. This shows it has some predictive power, but there's significant room for improvement.
- **Visualization:** The decision boundary for the Linear SVM is a **straight line** . Since the Moons dataset is **not linearly separable** (the two classes are intertwined in a crescent shape), the linear boundary cannot effectively separate the classes, leading to misclassifications for points near the "curves" of the moons. This demonstrates that a linear model is insufficient for this non-linear problem.

2. Compare the decision boundaries of the RBF and Polynomial kernels. Which one seems to capture the shape of the data more naturally?
   Both the RBF and Polynomial kernels perform significantly better than the Linear kernel, with similar, high accuracy:

- **RBF SVM Accuracy (C=1.0):**
- **Polynomial SVM (degree=3, C=1.0) Accuracy:**
  The **Polynomial kernel** (degree=3) seems to **capture the shape of the data slightly more naturally**.
- **Polynomial Kernel Visualization:** Its decision boundary has a distinct **curved shape** that closely follows the contours of the two crescent-shaped clusters. This shape looks slightly more tailored to the non-linear structure of the dataset than the RBF boundary.
- **RBF Kernel Visualization:** The RBF kernel's boundary is also non-linear and separates the data very well, but it tends to create a **smoother, more organic** boundary that is less directly influenced by the specific polynomial-like curve of the data, achieving a very similar, slightly lower, accuracy.

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:
   The **"Soft Margin" ()** model produces a **wider margin** .

- A **lower value of** corresponds to a softer margin, which tolerates more misclassifications (or points inside the margin) to find a simpler, more generalized decision boundary with a wider margin.

- A **higher value of** (like 100) corresponds to a harder margin, which places a higher penalty on misclassification errors, forcing the model to find a narrower margin that separates the training data more perfectly.

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

The SVM allows these "mistakes" because the parameter (the penalty for misclassification) is set to a **low value ()**. This prioritizes **finding the widest possible margin** over perfectly classifying every single training point.

The primary goal of a soft margin SVM is to find the optimal trade-off between:

Maximizing the width of the margin.

Minimizing the number of misclassification errors (or "slack").

When is low, the optimization problem places more importance on the **margin width** (regularization) than on the error penalty, leading to a more generalized model that is less sensitive to individual noisy points.

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?

The **"Hard Margin" ()** model is more likely to be **overfitting** .

- A high value of (like 100) means the model is **strongly penalized for every misclassification** on the training data.
- To avoid these penalties, the hard margin model finds a complex, potentially highly non-linear boundary that forces nearly all training points to be correctly classified or outside the margin. This focus on fitting the training data exactly, even the noise or outliers, results in a more complex decision boundary that may not generalize well to unseen data. The accuracy for is slightly lower (0.94) than the RBF default (, accuracy 0.9467), suggesting this complexity might not be beneficial.

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?

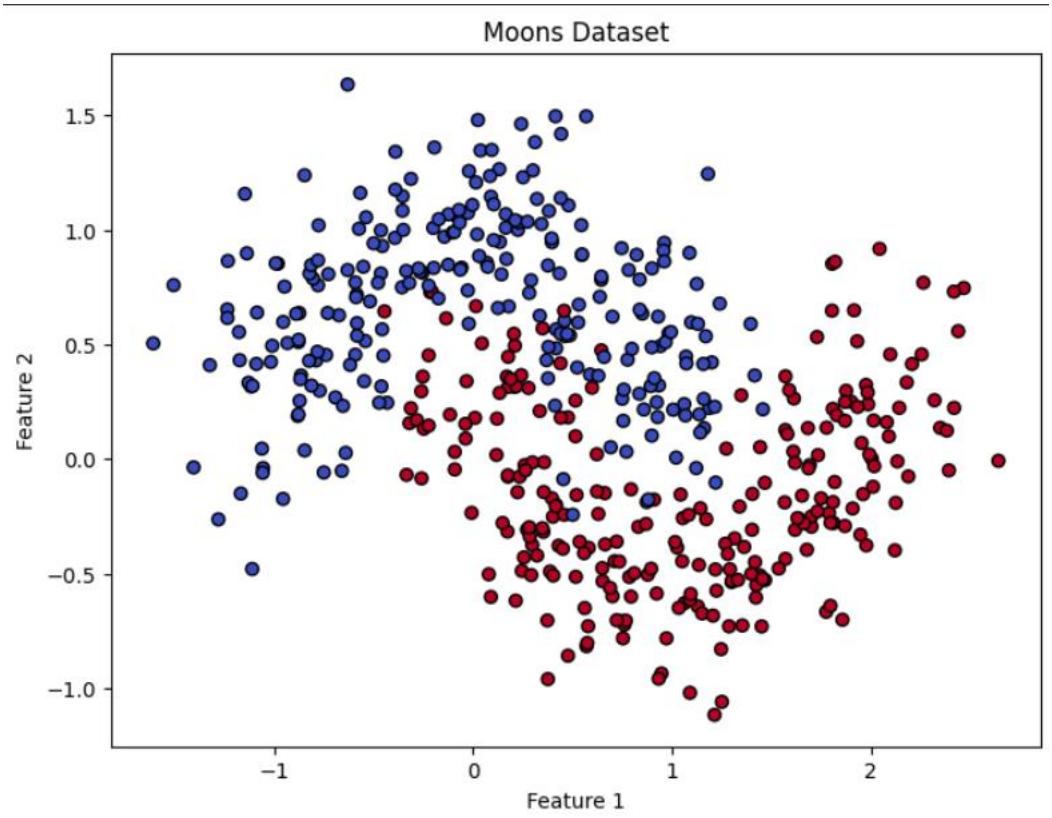
I would trust the **"Soft Margin" ()** model (or, more broadly, an intermediate value like) more to classify a new, unseen data point correctly.

- **Why:** The softer margin produces a **more generalized decision boundary** by tolerating minor errors in the training data. This makes the model more robust to the natural variability and noise that exist in real-world data, whereas the hard margin model is overly sensitive to the training set.

In a real-world scenario where data is often noisy, you would generally prefer to start with a **low (or intermediate) value of** . A low helps prevent immediate overfitting to noise and outliers, yielding a model that is more likely to generalize effectively to new data.

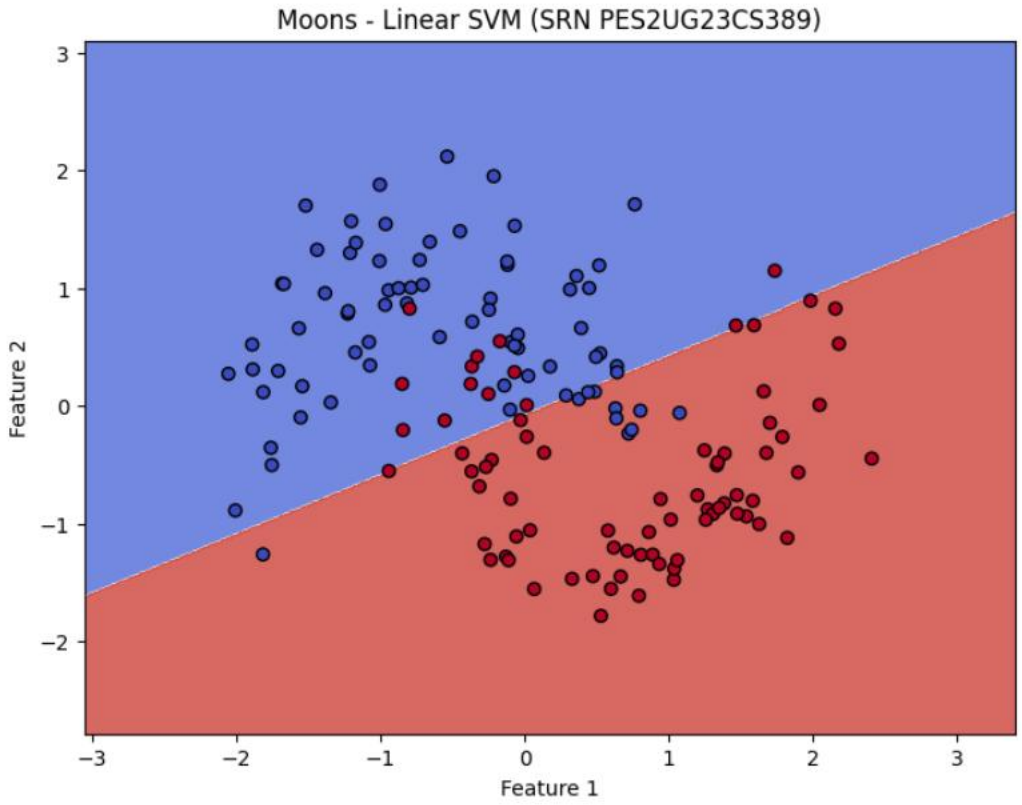
SCREENSHOTS :

1. Moons Dataset :



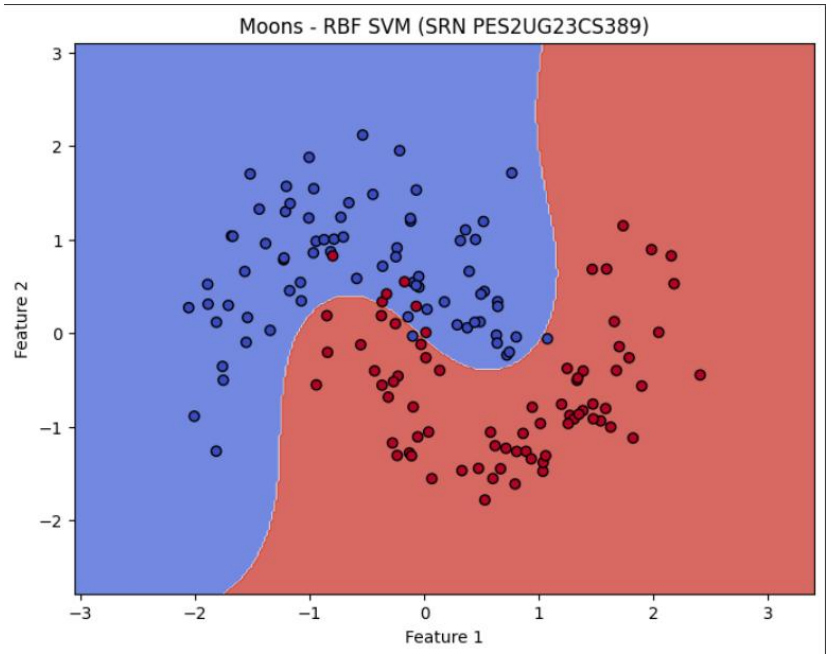
(a) SVM with Linear Kernel

```
=== Moons: Linear SVM ===
SRN: PES2UG23CS389
              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
```



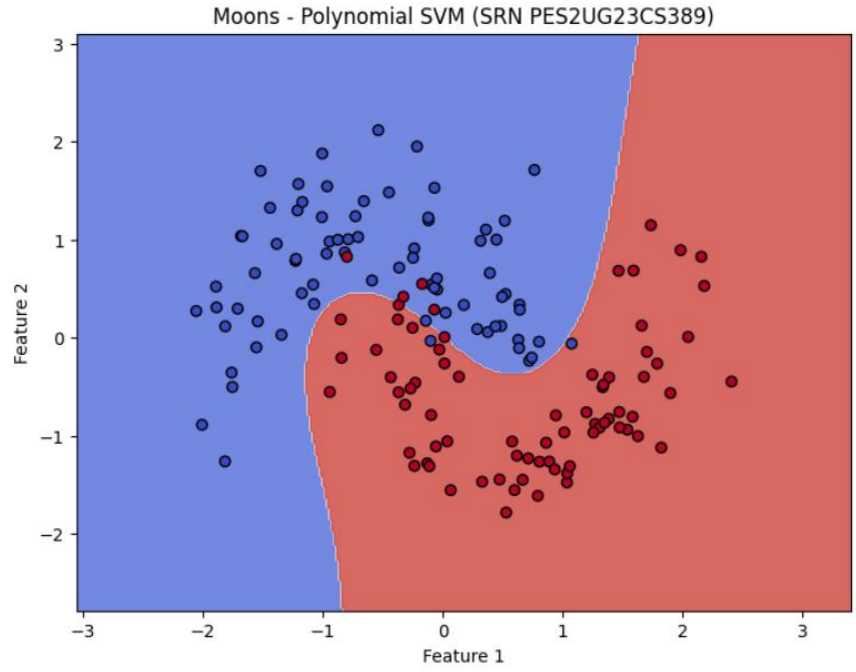
(b) SVM with RBF Kernel :

```
=== Moons: RBF SVM ===
SRN: PES2UG23CS389
              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
```



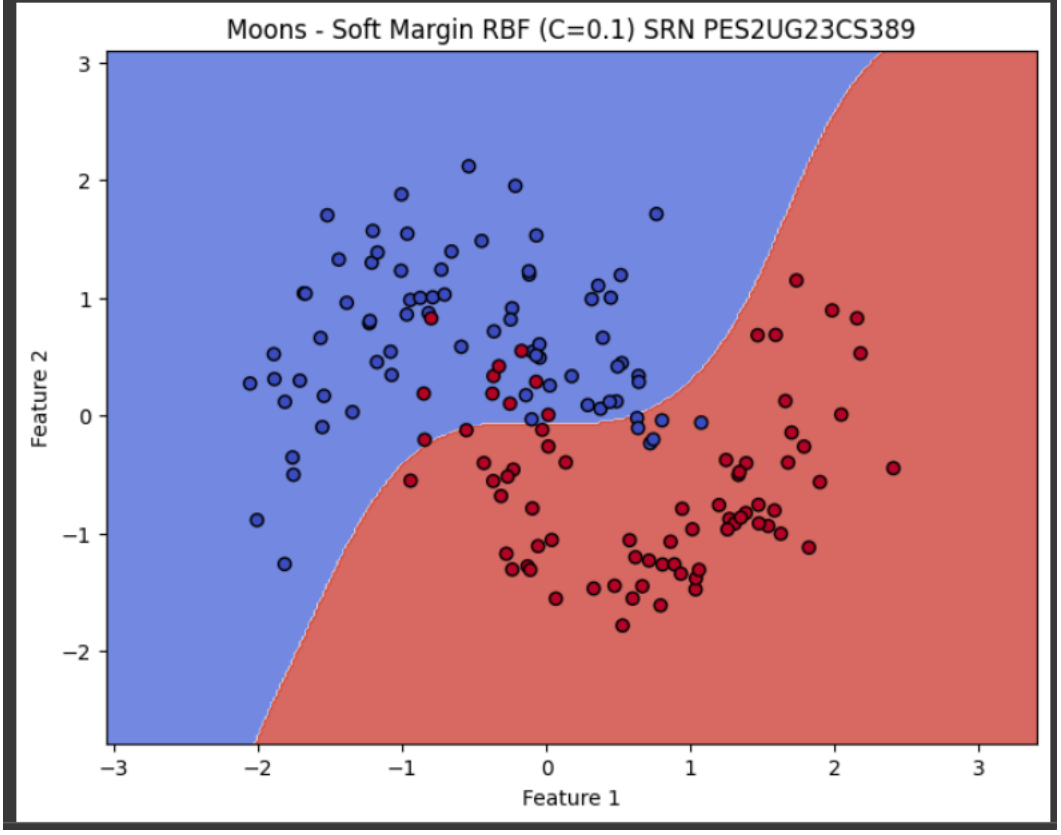
(c) SVM with POLY kernel :

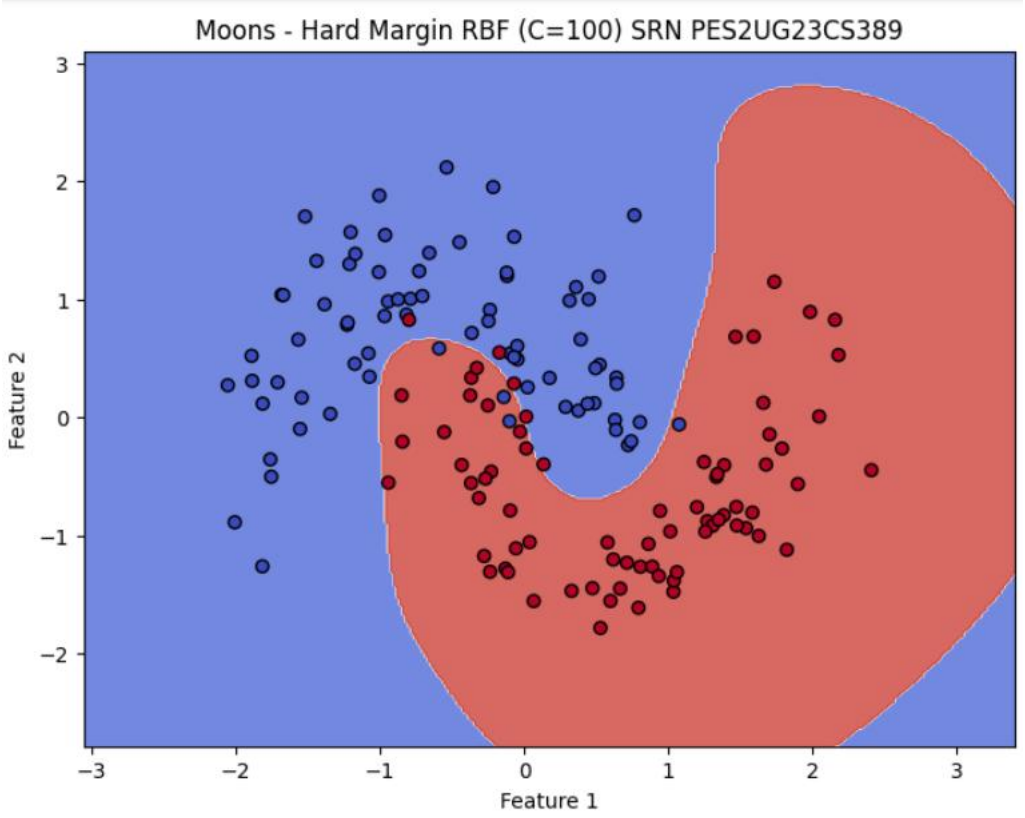
```
=== Moons: Poly SVM ===
SRN: PES2UG23CS389
              precision    recall  f1-score   support

           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
```

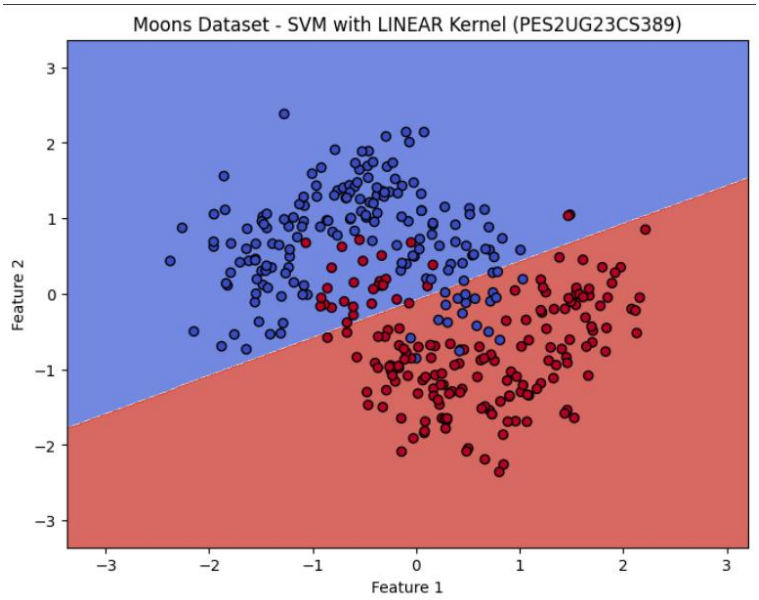


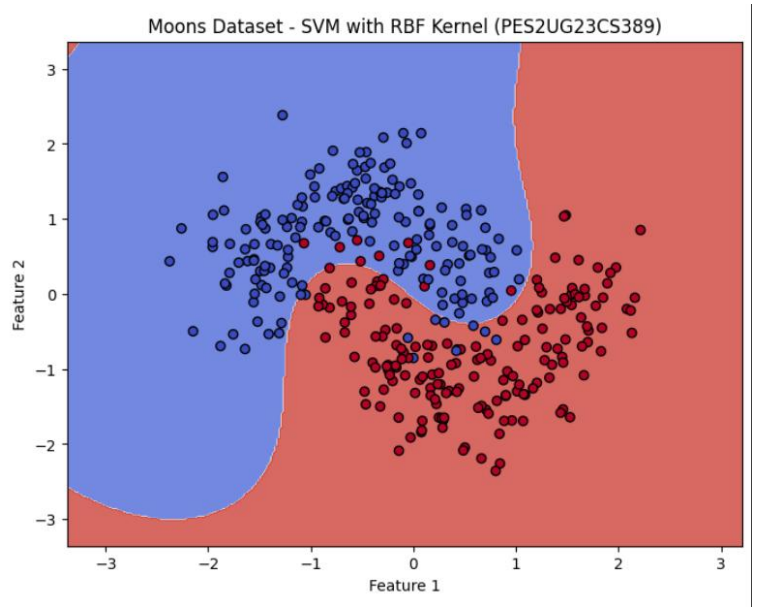
2. Margin Analysis :

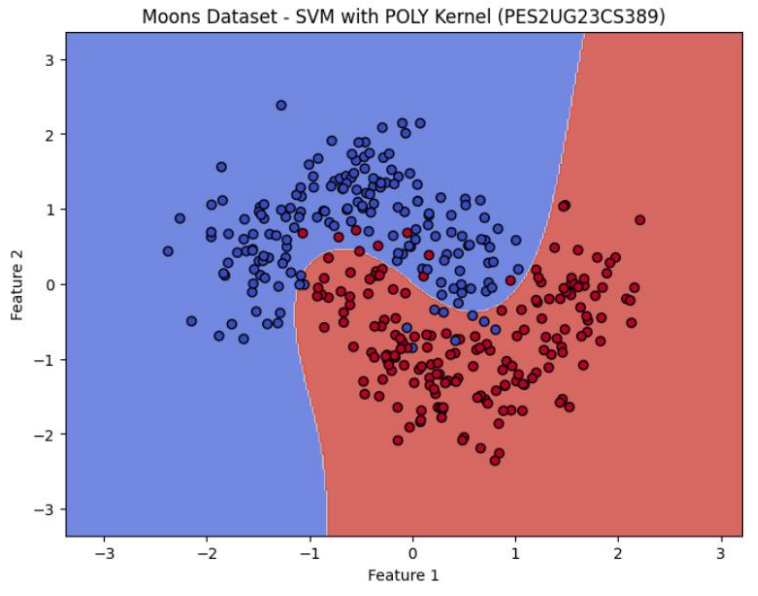
```
Moons - Soft (C=0.1) Accuracy: 0.8933333333333333
Moons - Hard (C=100) Accuracy: 0.94
```






Moons Dataset - SVM with LINEAR Kernel (PES2UG23CS389)
Feature 1
Feature 2


Moons Dataset - SVM with RBF Kernel (PES2UG23CS389)
Feature 1
Feature 2


Moons Dataset - SVM with POLY Kernel (PES2UG23CS389)
Feature 1
Feature 2