UE23CS352A: MACHINE LEARNING

Week 10: SVM

Name	SRN	Section
C Panshul Reddy	PES2UG23CS154	C

Date:11-10-2025

Moons Dataset:

Classification Report for SVM with LINEAR Kernel

```
SVM with LINEAR Kernel <PES2UG23CS154>
              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

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

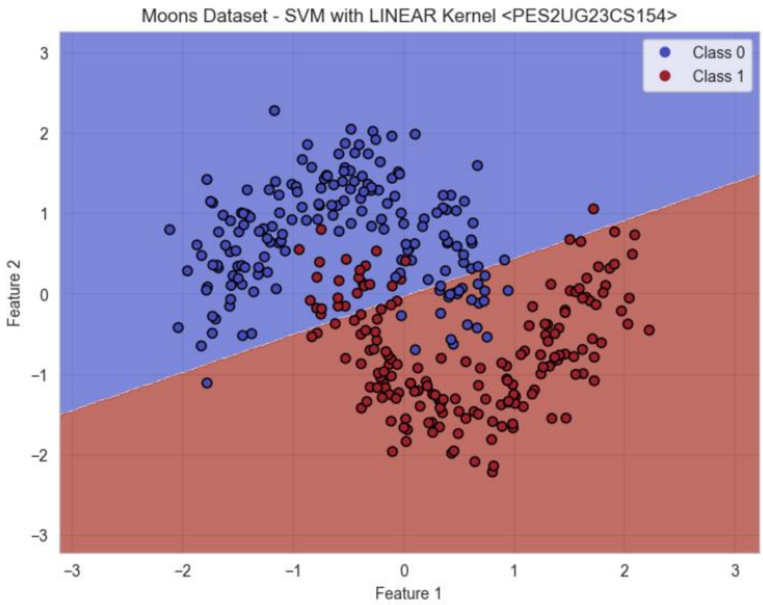
Classification Report for SVM with RBF Kernel

```
SVM with RBF Kernel <PES2UG23CS154>
              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

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

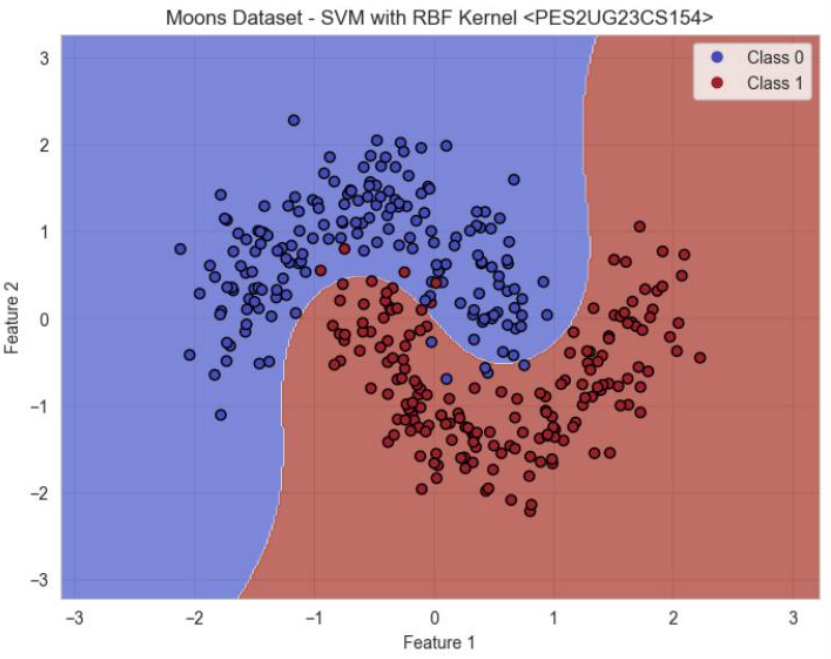
Classification Report for SVM with POLY Kernel

```
SVM with POLY Kernel <PES2UG23CS154>
...
weighted avg       0.89      0.89      0.89       150

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

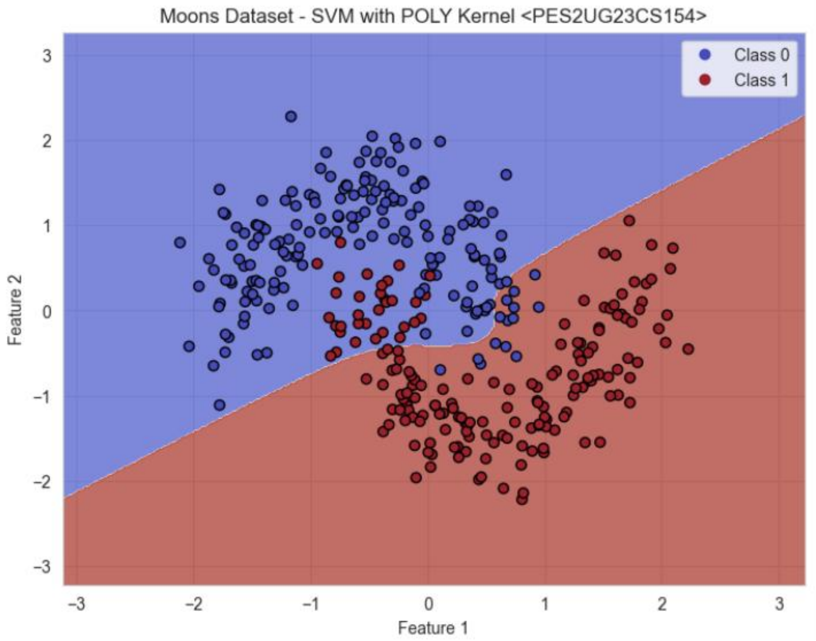
SVM with LINEAR Kernel



SVM with RBF Kernel



SVM with POLY Kernel



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

Ans.

The linear kernel produces a straight line decision boundary, which fails to capture the non-linear pattern of the moons dataset (which is naturally curved).

While it performs decently (87% accuracy), the model misclassifies samples along the curved edges, as seen where points from both classes are intermixed near the boundary.

This indicates that the dataset is not linearly separable, and a linear kernel lacks flexibility to adapt to the data's true structure.

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

Ans.

The RBF kernels decision boundary smoothly wraps around both clusters, effectively capturing the complex, non-linear relationships between the features. (Accuracy=97%)

The Polynomial kernel introduces some curvature but may either *underfit or overfit* depending on its degree — here it's doing slightly better than Linear but worse than RBF. (Accuracy=89%)

Banknote Dataset:

Classification Report for SVM with LINEAR Kernel

```
SVM with LINEAR Kernel <PES2UG23CS154>
              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

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

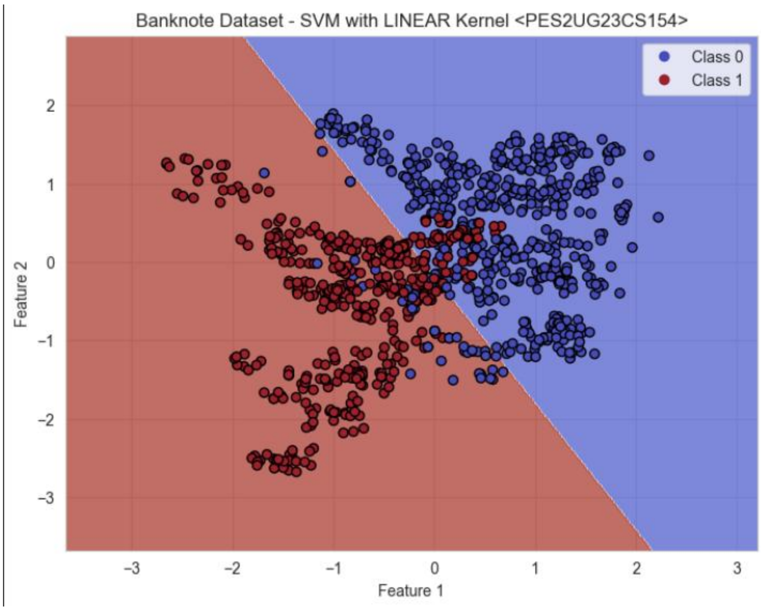
Classification Report for SVM with RBF Kernel

```
SVM with RBF Kernel <PES2UG23CS154>
              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

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

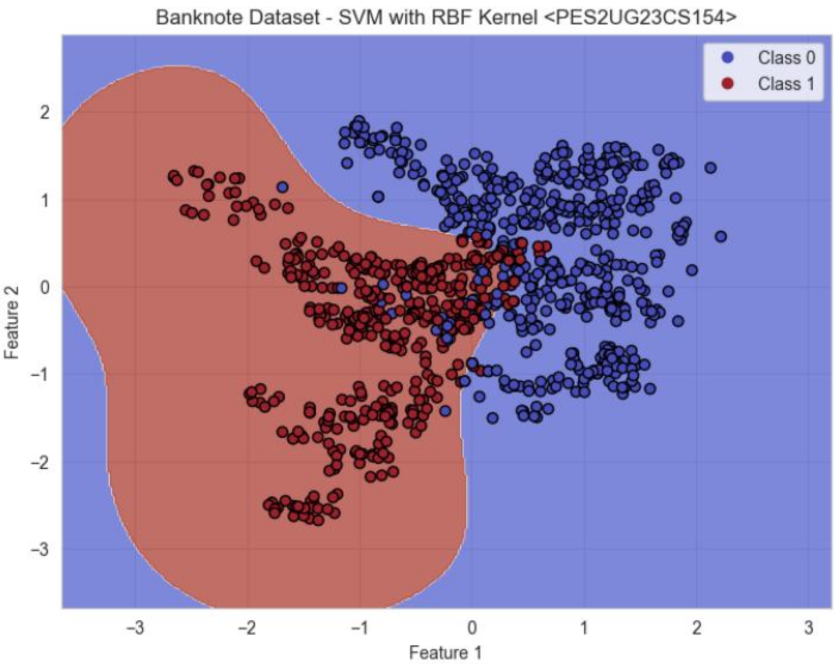
Classification Report for SVM with POLY Kernel

```
SVM with POLY Kernel <PES2UG23CS154>
...
weighted avg       0.85      0.84      0.84       412

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

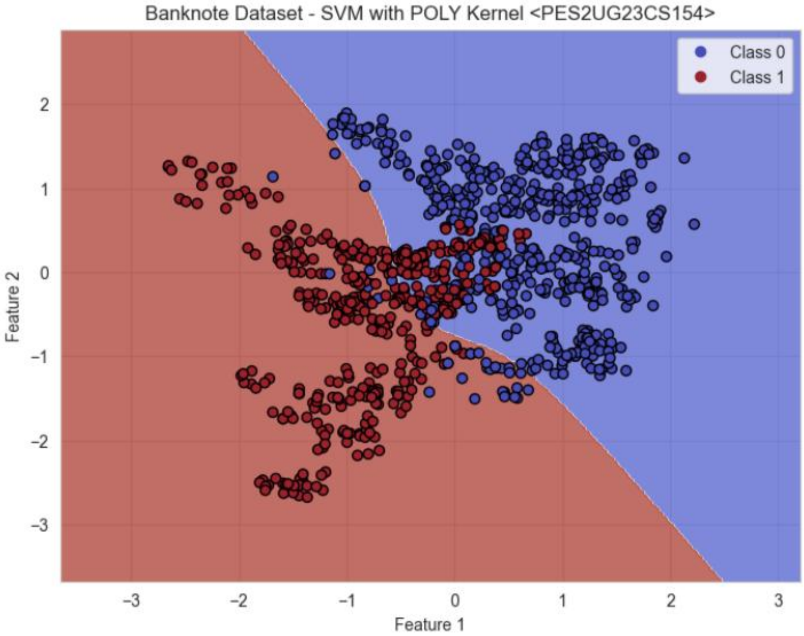
SVM with LINEAR Kernel



SVM with RBF Kernel



SVM with POLY Kernel



1) In this case, which kernel appears to be the most effective?

   Ans.

   The RBF Kernel is the most effective for this dataset — it achieves the highest accuracy (93%) and best-balanced F1-scores (0.94 and 0.93).
   Its non-linear, smooth boundary adapts well to the slightly complex structure of the data, leading to fewer misclassifications compared to Linear and Polynomial kernels.

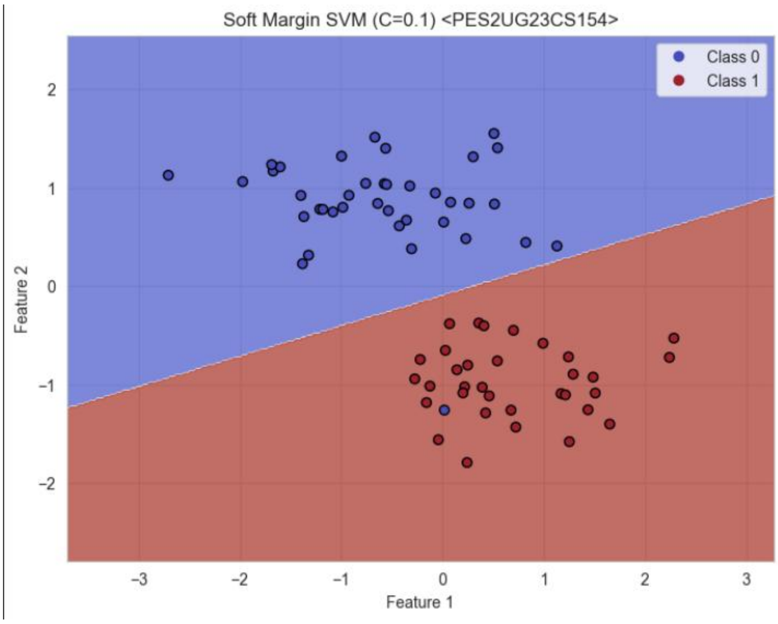
2) The Polynomial kernel shows lower performance here compared to the Moons dataset. What might be the reason for this?

   Ans.

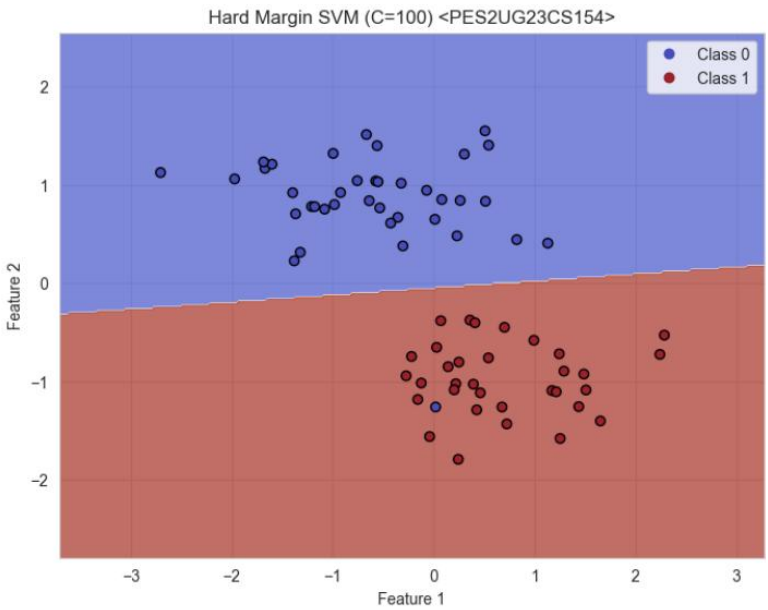
   The relationship between features is more intricate and irregular, not easily expressed as a simple polynomial function.
   Polynomial kernels may overfit or misalign when data doesn't follow a smooth, regular curve.
   The RBF kernel, being based on localized similarity , can adapt better to arbitrary and complex decision regions, while the polynomial kernel struggles to generalize.

Hard vs. Soft Margin Dataset:

Soft Margin SVM (C=0.1)



Hard Margin SVM (C=100)



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

Ans.

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

We can see that the decision boundary is farther away from the nearest points, leaving a broader gap between the classes.

In contrast, the Hard Margin SVM (C = 100) has a narrower margin, tightly fitted around the nearest data points.

2. 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?

Ans.

In the Soft Margin SVM (C = 0.1):

- Some points lie inside the margin or even on the wrong side of the decision boundary.
- The model allows these mistakes because it's trying to balance two goals:
    1. Maximize the margin (make it as wide as possible).
    2. Minimize misclassification errors, but not necessarily eliminate them.

This trade-off helps the model to generalize better to new, unseen data by not overfitting to small irregularities or noise.

3. Which of these two models do you think is more likely to be overfitting to the training data? Explain your reasoning.

Ans.

The Hard Margin SVM (C = 100) is more likely to overfit.

A large C value means the model heavily penalizes misclassifications, forcing it to correctly classify every training point — even if that means creating a very tight boundary around the data.

This can lead to poor generalization on new data, especially if the dataset contains noise or outliers.

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?

Ans.

For new, unseen data, the Soft Margin SVM ($C = 0.1$) is more trustworthy.

It's less sensitive to individual training points and noise, providing smoother, more stable decision boundaries.

In real-world datasets — which almost always contain some noise and overlapping classes — a lower C value (Soft Margin) is preferred as a starting point.