ML Lab: Week 10

Name:

Maya Nithyanand Bhagath

SRN:

PES2UG23CS331

Section:

F

Analysis Questions

Moons Dataset

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

The Linear Kernel is not suitable for the Moons dataset. A linear function should be able to divide the data points using a straight line. However, the Moons dataset is non-linearly separable, so it cannot be divided by a straight line. This leads to misclassification and poor performance metrics.

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 creates a complex, non-linear boundary that is highly flexible. This results in a smooth curve that leads to an almost-perfect separation of data points. The Polynomial kernel also creates a curved boundary, but it is more rigid. So it isn't as precise as the RBF kernel. Therefore, the RBF provides more accurate fit for the Moons dataset.

Banknote Dataset

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

The RBF kernel is the most effective for the Banknote dataset. It gives an almost perfect boundary.

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

The reason for this might be hyperparameter sensitivity and data complexity. The Banknote dataset has a more complex and high-dimensional structure as compared to the Moons dataset. Therefore, the default degree selected by the Polynomial kernel might not have been right for the values in the Banknote dataset.

Hard vs. Soft Margin

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

The Soft Margin (C=0.1) model produces a visibly wider margin. This is because a smaller C value prioritizes a larger gap between the decision boundary and the support vectors.

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?

The SVM allows these mistakes as a trade-off. The primary goal of the SVM is not to perfectly classify all points, but to create a decision boundary that generalizes well to new, unseen data. It helps the model not be overly affected by potential noise and minimizes classification errors.

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

The Hard Margin (C=100) model is more likely to be overfitting. A large C value places a high penalty on misclassification, forcing the model to create a narrow, complex boundary. This makes the model learn the exact training data, including the noise and the outliers, rather than learning its underlying pattern. This would lead to overfitting.

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?

The Soft Model (C=0.1) model would be more trustworthy in classifying a new, unseen data point. It has a wider margin, so it is less sensitive to specifics like the noise in the data, telling us that it has captured the general trend in the data.

In a real-world scenario, data almost always has noise, so it is better to start with a low value of C. This prevents overfitting.

Screenshots

Training Results

Moons Dataset

```
SVM with LINEAR Kernel <PES2UG23CS331>
              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 <PES2UG23CS331>
              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 <PES2UG23CS331>
              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

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

Banknote Dataset

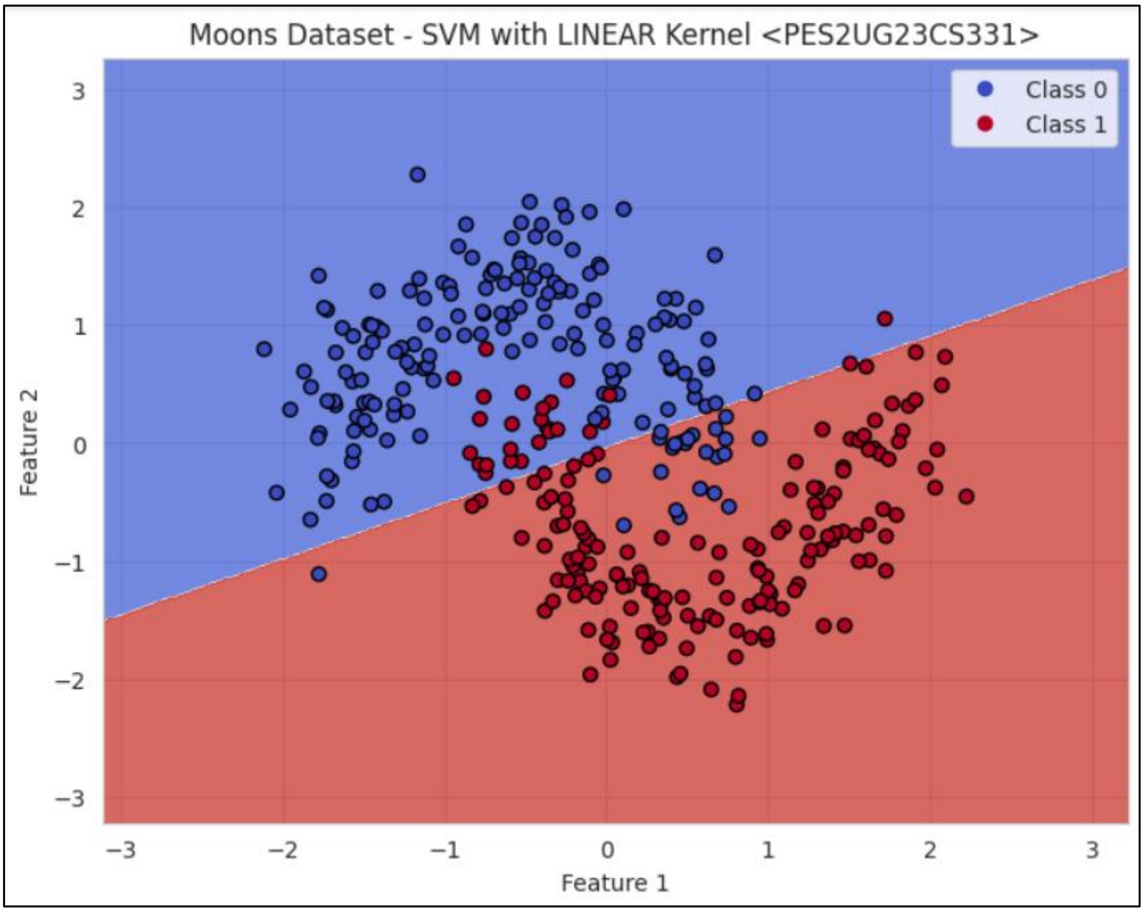
```
SVM with LINEAR Kernel <PES2UG23CS331>
              precision    recall  f1-score   support

      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

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

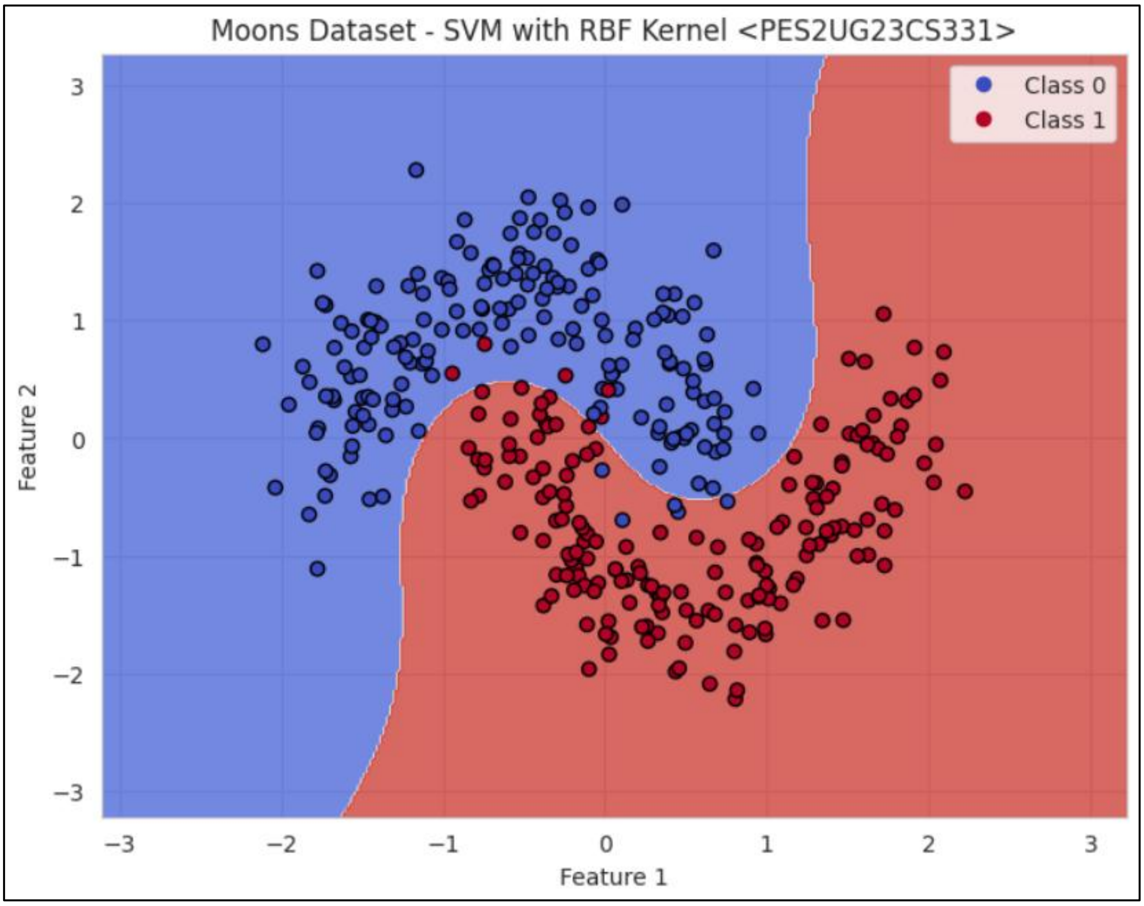
```
SVM with RBF Kernel <PES2UG23CS331>
              precision    recall  f1-score   support

      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

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

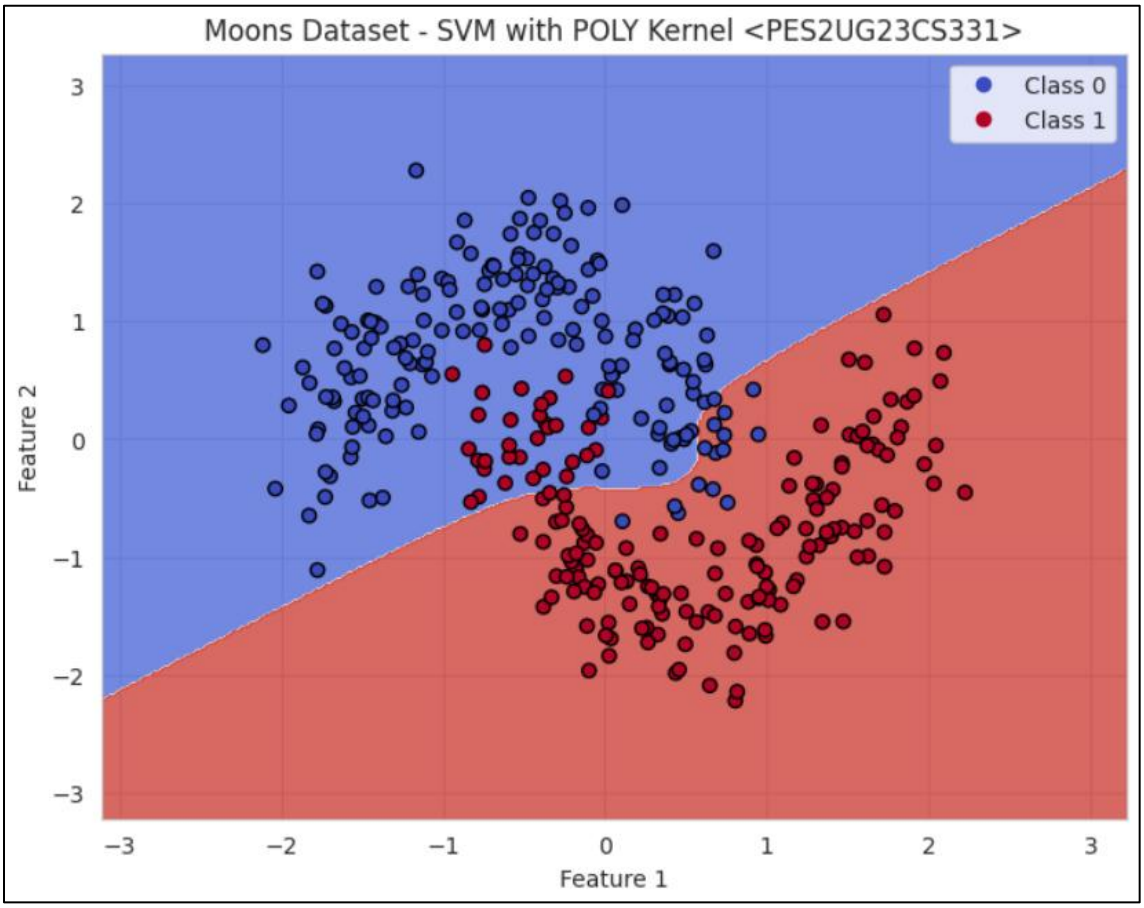
```
SVM with POLY Kernel <PES2UG23CS331>
              precision    recall  f1-score   support

      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

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

Decision Boundary Visualizations

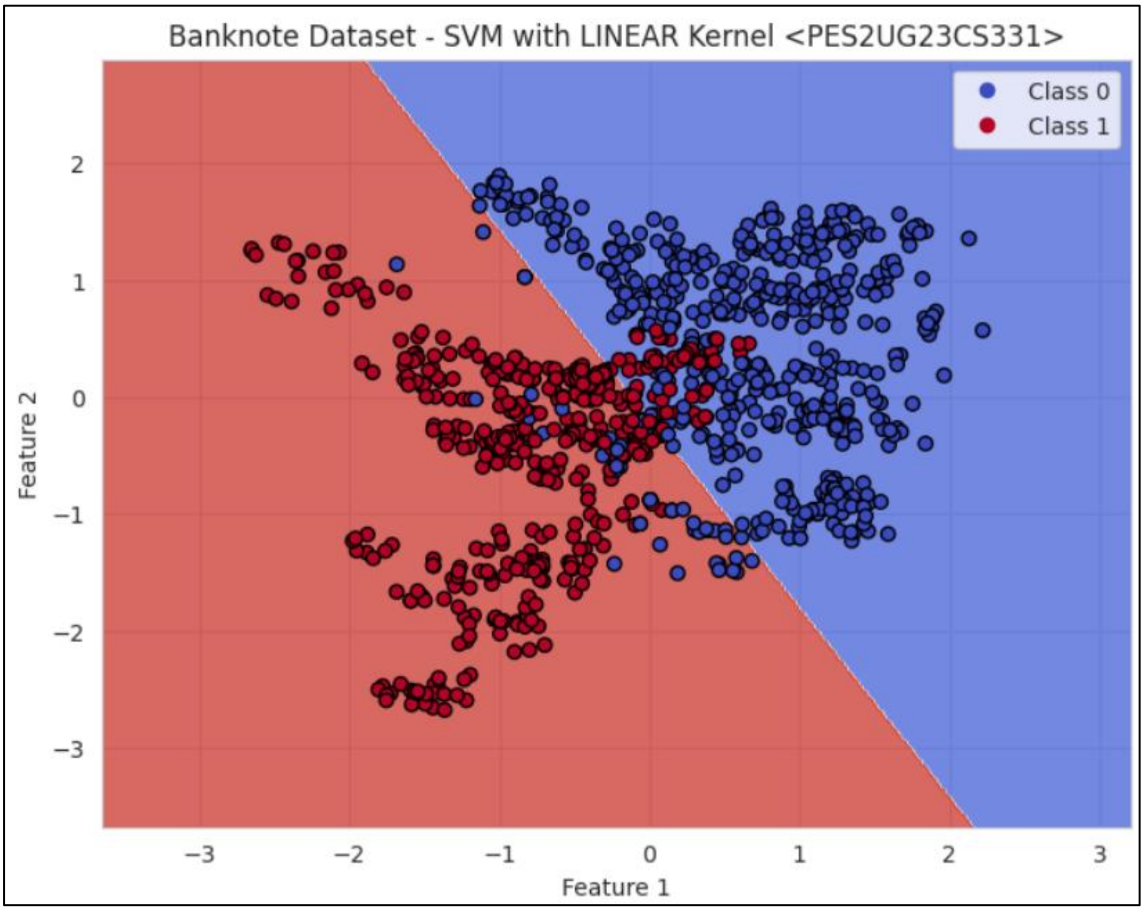
Moons Dataset

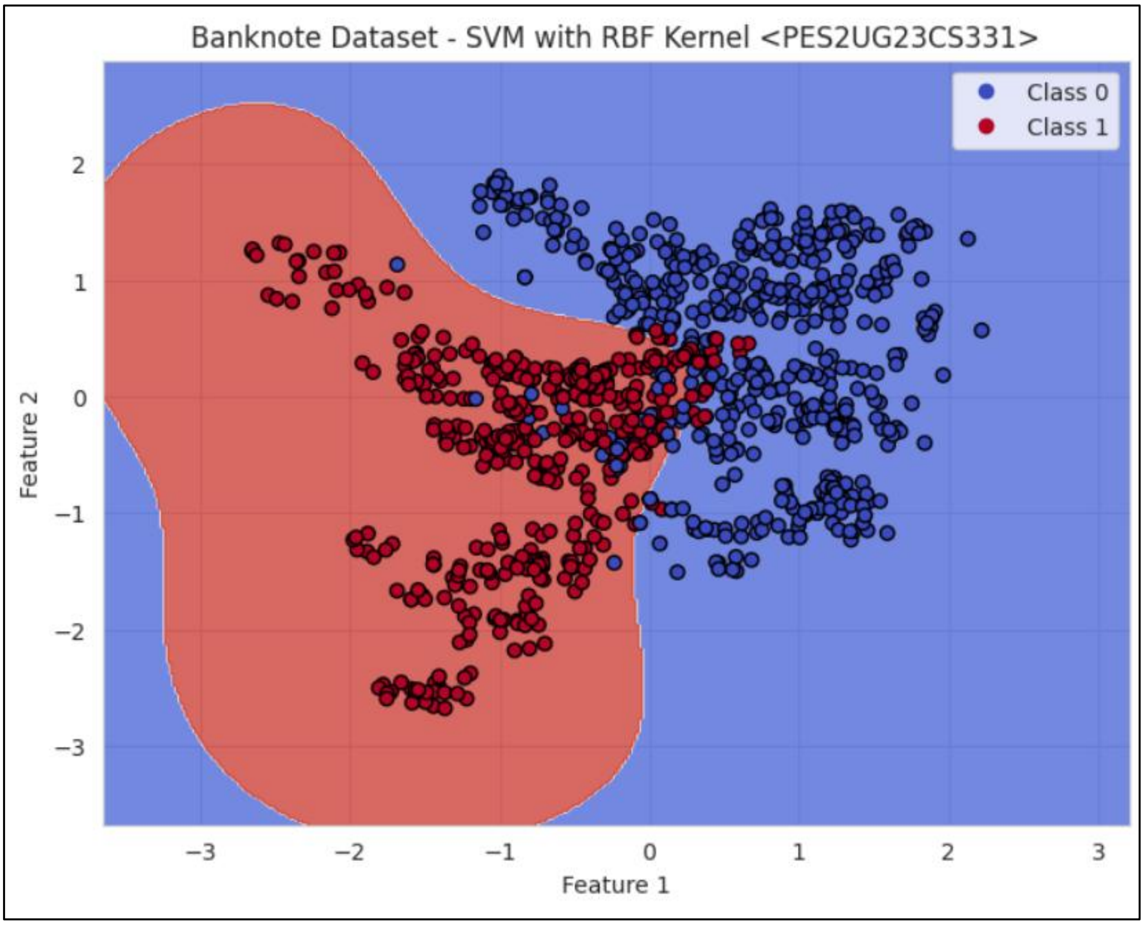


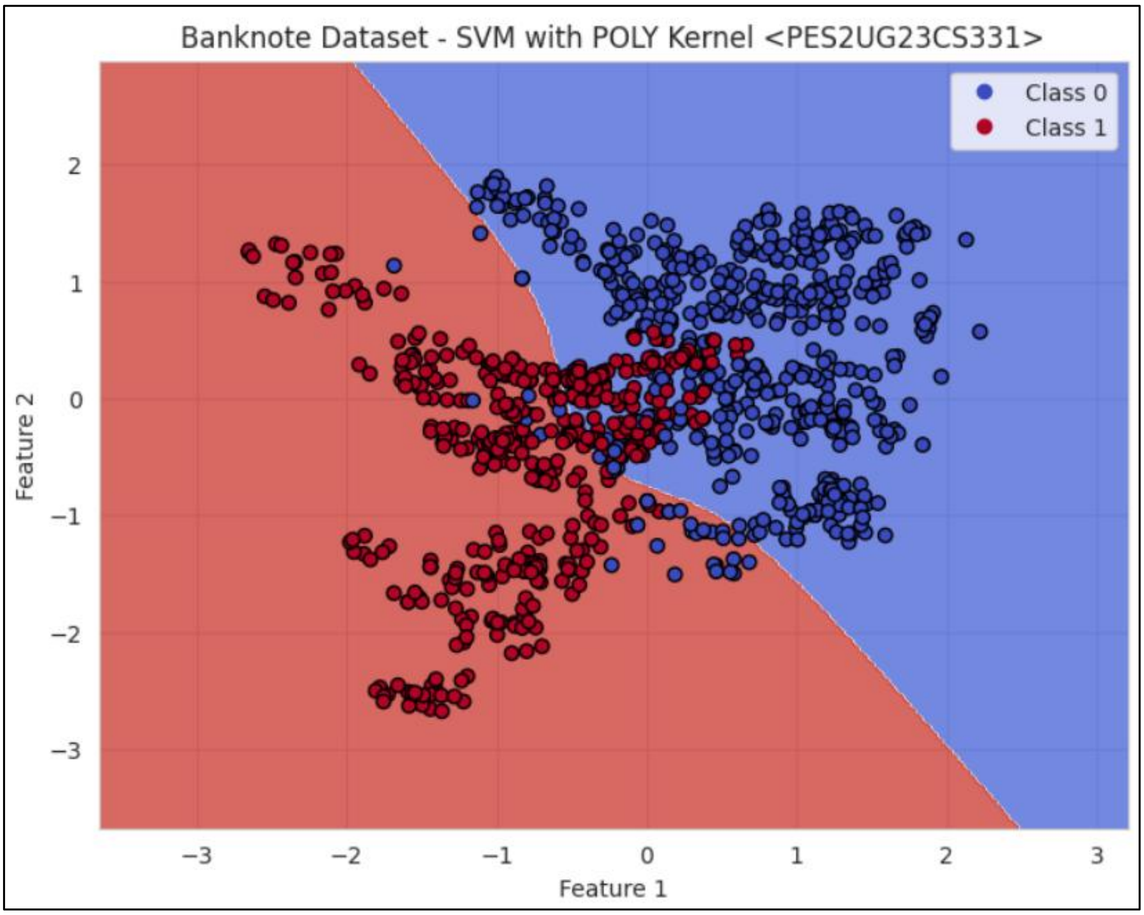

Moons Dataset - SVM with RBF Kernel <PES2UG23CS331>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


Moons Dataset - SVM with POLY Kernel <PES2UG23CS331>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Banknote Dataset




Banknote Dataset - SVM with RBF Kernel <PES2UG23CS331>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


Banknote Dataset - SVM with POLY Kernel <PES2UG23CS331>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Margin Analysis