MACHINE LEARNING LAB WEEK – 7

Name: Navyata Venkatesh

SRN: PES2UG23CS375

Section: F

Q1. Based on the metrics and the visualizations, what inferences about the performance of the Linear Kernel can you draw?

Ans: The Linear Kernel shows decent performance on datasets that are linearly separable, but it struggles with non-linear patterns. From the metrics and visualizations, it's clear that the linear boundary fails to capture the curved structure of data like the Moons dataset, leading to a noticeable drop in accuracy and recall. Its straight decision line indicates limited flexibility, making it suitable for simple, well-separated datasets such as the Banknote dataset, but less effective for complex, non-linear ones.

Q2. Compare the decision boundaries of the RBF and Polynomial kernels. Which one seems to capture the shape of the data more naturally?

Ans: When comparing the RBF and Polynomial kernels, the RBF kernel captures the overall shape of the data more naturally. It forms smooth, adaptive decision boundaries that closely follow the underlying distribution without overfitting. The Polynomial kernel also models non-linearity but tends to produce more rigid or uneven boundaries, especially for higher degrees. Overall, the RBF kernel offers better generalization and a more realistic representation of non-linear relationships in the data.

Q3. In this case, which kernel appears to be the most effective?

Ans: In this case, the RBF kernel appears to be the most effective. It consistently delivers the highest accuracy and balanced precision–recall scores, adapting well to the underlying non-linear relationships in the data. Its ability to create smooth, flexible decision boundaries allows it to separate complex class structures more effectively than the Linear or Polynomial kernels, making it the most reliable choice overall.

Q5. The Polynomial kernel shows lower performance here compared to the Moons dataset. What might be the reason for this?

Ans: The Polynomial kernel shows lower performance here because it is more sensitive to the degree parameter and can easily overfit or underfit depending on the dataset. Unlike the Moons dataset, which has smoother non-linear separations, this dataset's patterns may not align well with the rigid curvature created by the polynomial function. As a result, the Polynomial kernel struggles to capture the true distribution of the data, leading to reduced accuracy and less generalizable decision boundaries.

Q6. 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 model (C = 0.1) produces a wider margin compared to the Hard Margin model (C = 100). This happens because a smaller C value allows the SVM to tolerate more misclassifications in exchange for a smoother and more generalized decision boundary. In contrast, the Hard Margin model, with a large C, tightly fits the data and results in a narrower margin.

Q7. 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?

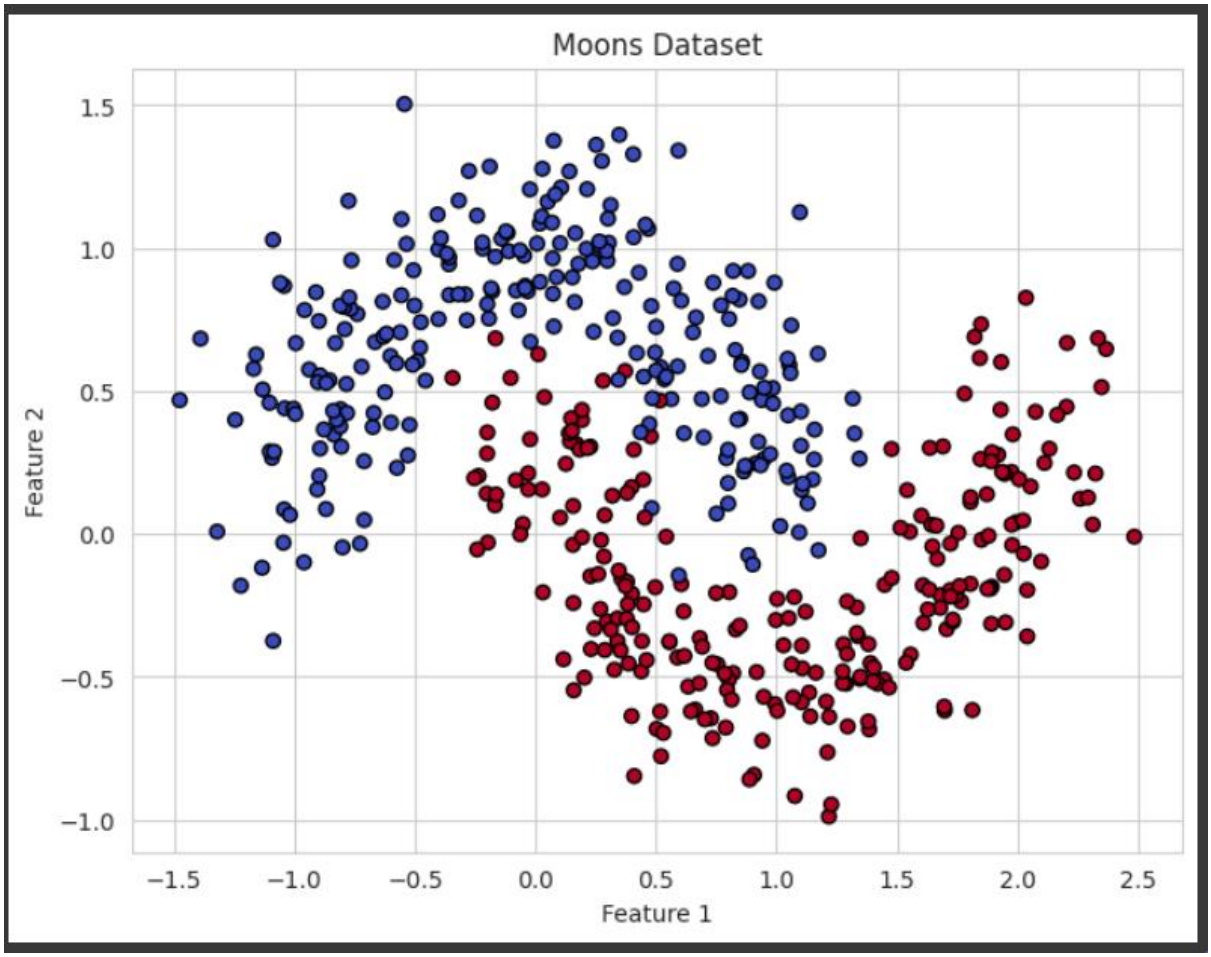
Ans: In the Soft Margin plot, some points lie inside the margin or on the wrong side of the boundary because the SVM allows these mistakes to achieve better generalization. The primary goal of this model is not to classify every training example perfectly but to find a balance between maximizing the margin and minimizing classification errors, which helps it perform better on unseen data.

Q8. Which of these two models do you think is more likely to be overfitting to the training data? Explain your reasoning.

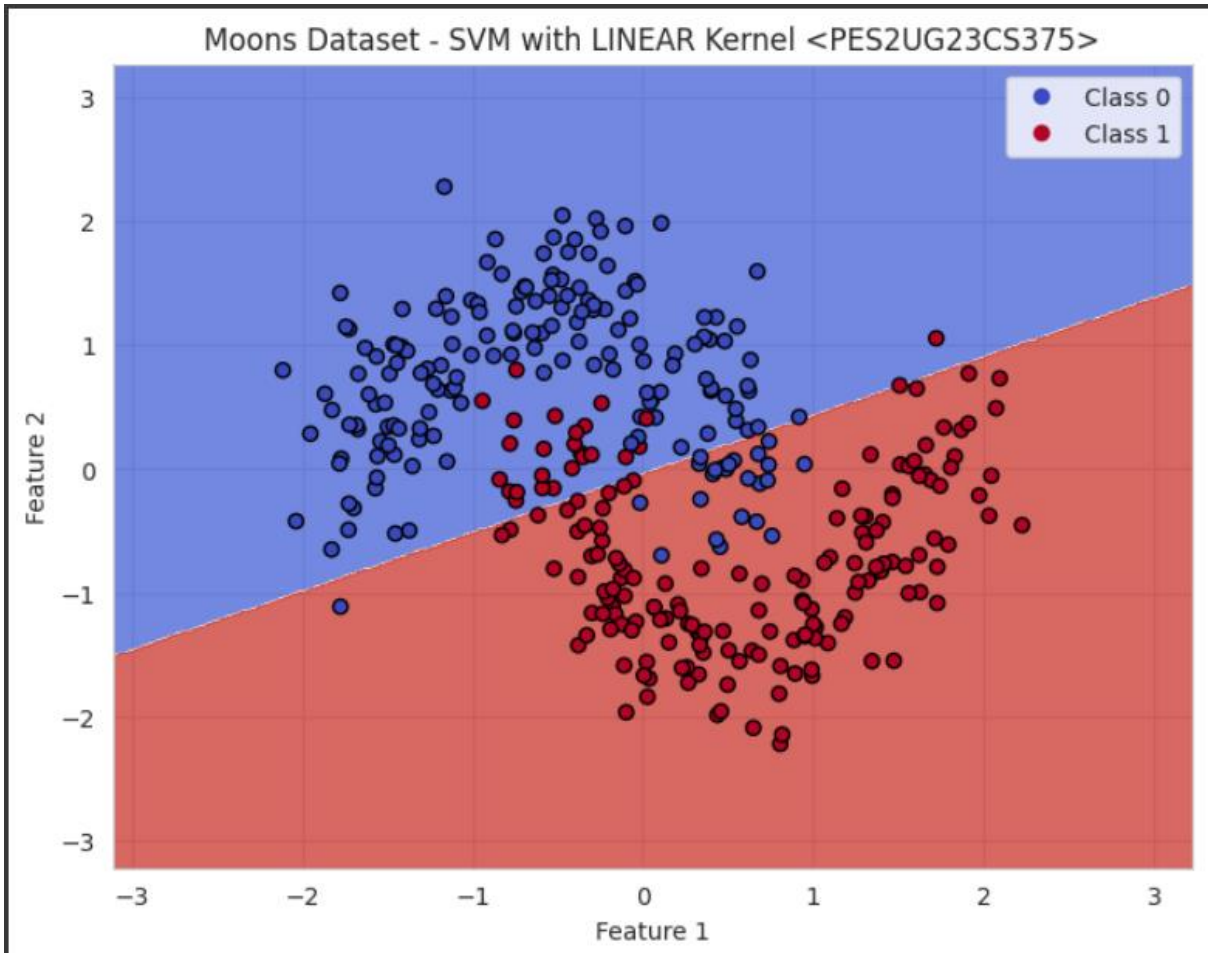
Ans: The Hard Margin model (C = 100) is more likely to overfit the training data. Since it penalizes misclassifications very strongly, it tries to perfectly separate the training points, even if that means forming a decision boundary that fits the noise in the data. This leads to high training accuracy but poor performance on new or slightly different data.

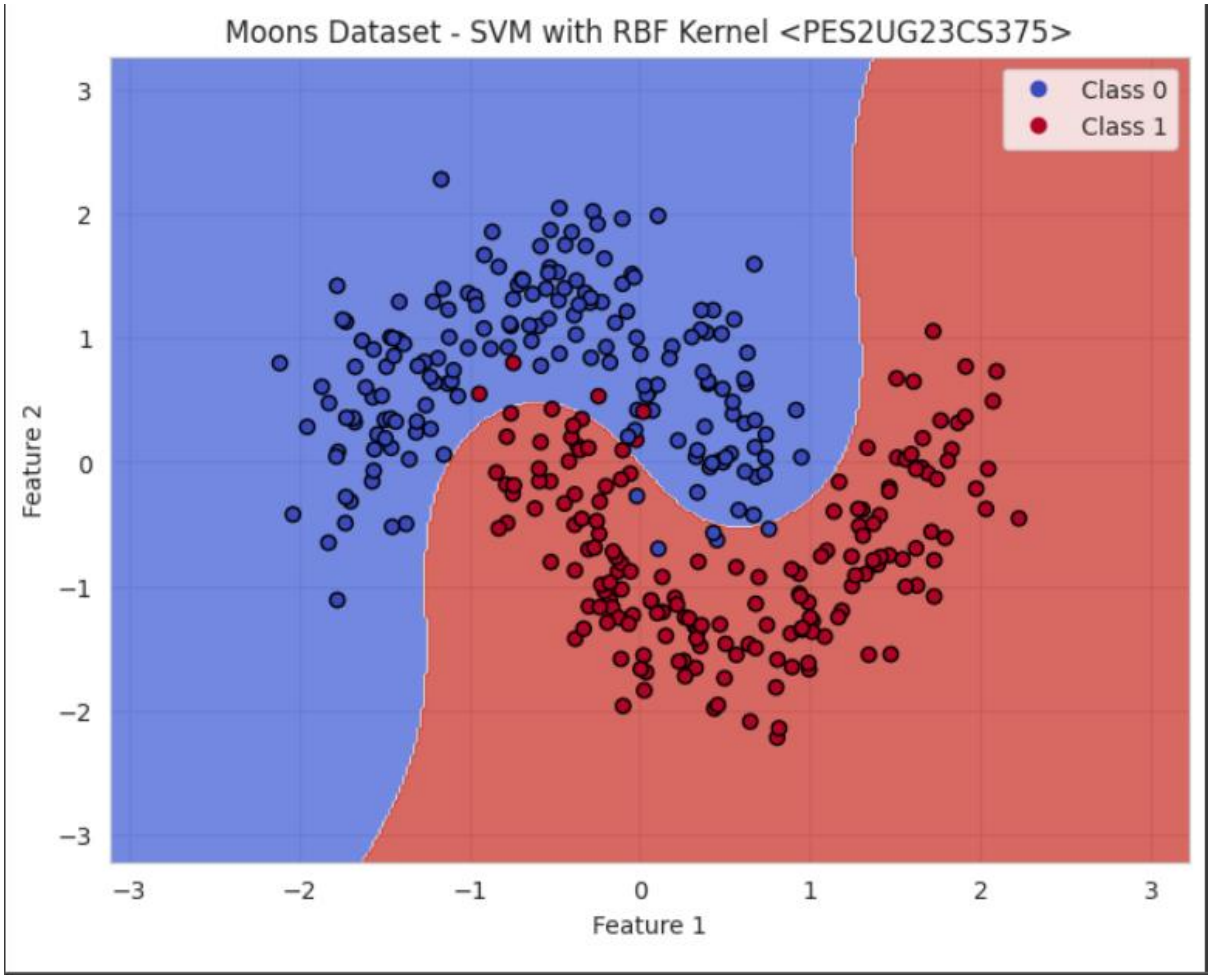
Q9. Imagine you receive a new, unseen data point. Which model do you trust more to classify it correctly? Why? In a real-world scenario where data is often noisy, which value of C (low or high) would you generally prefer to start with?

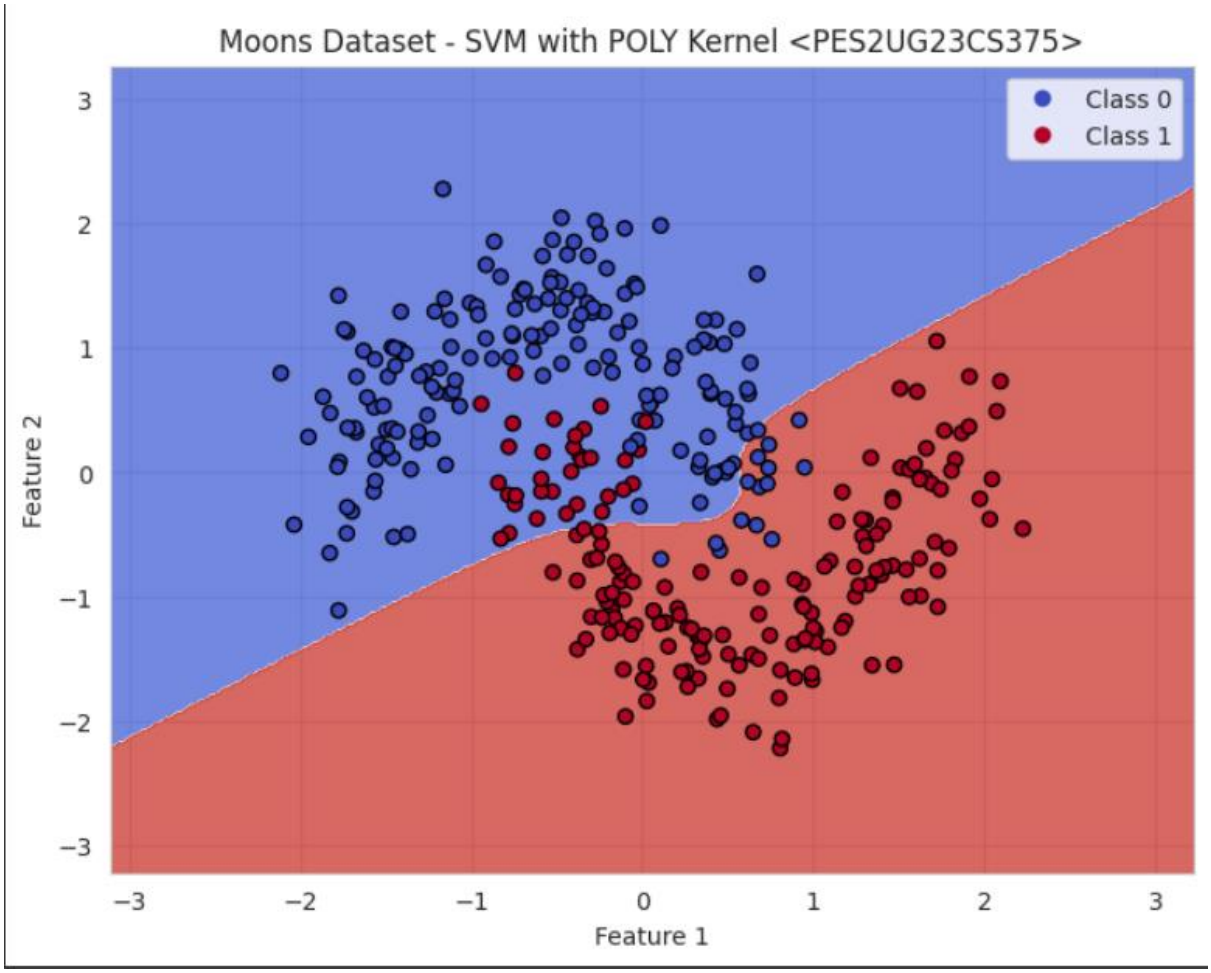
Ans: For new, unseen data, the Soft Margin model (C = 0.1) is generally more reliable because it focuses on generalization rather than perfect separation. In real-world scenarios where data is often noisy or overlapping, it is better to start with a lower C value, as it produces a simpler and more robust decision boundary that handles variability better.

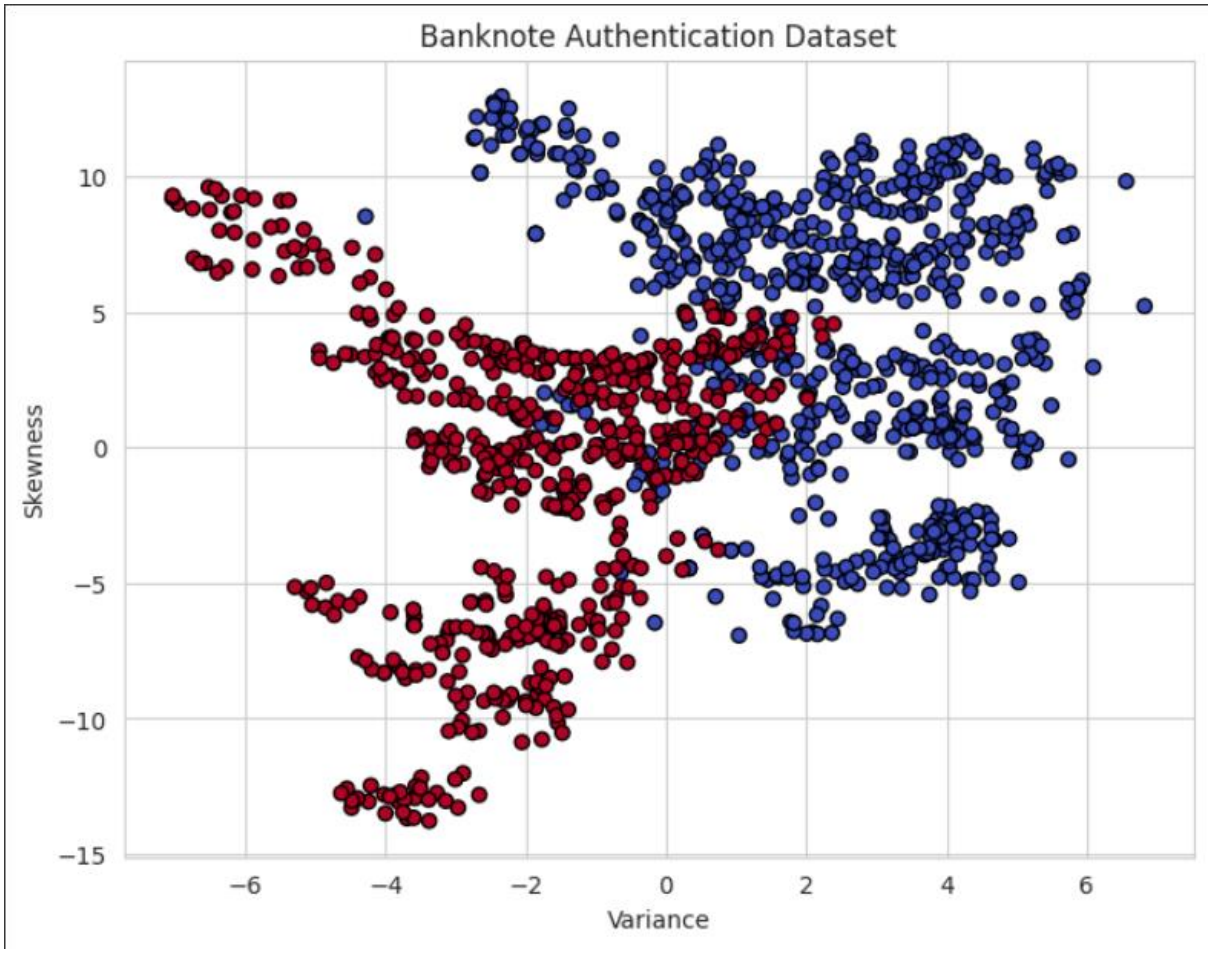


```
SVM with LINEAR Kernel <PES2UG23CS375>
              precision    recall  f1-score   support

           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 <PES2UG23CS375>
              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 <PES2UG23CS375>
              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
```

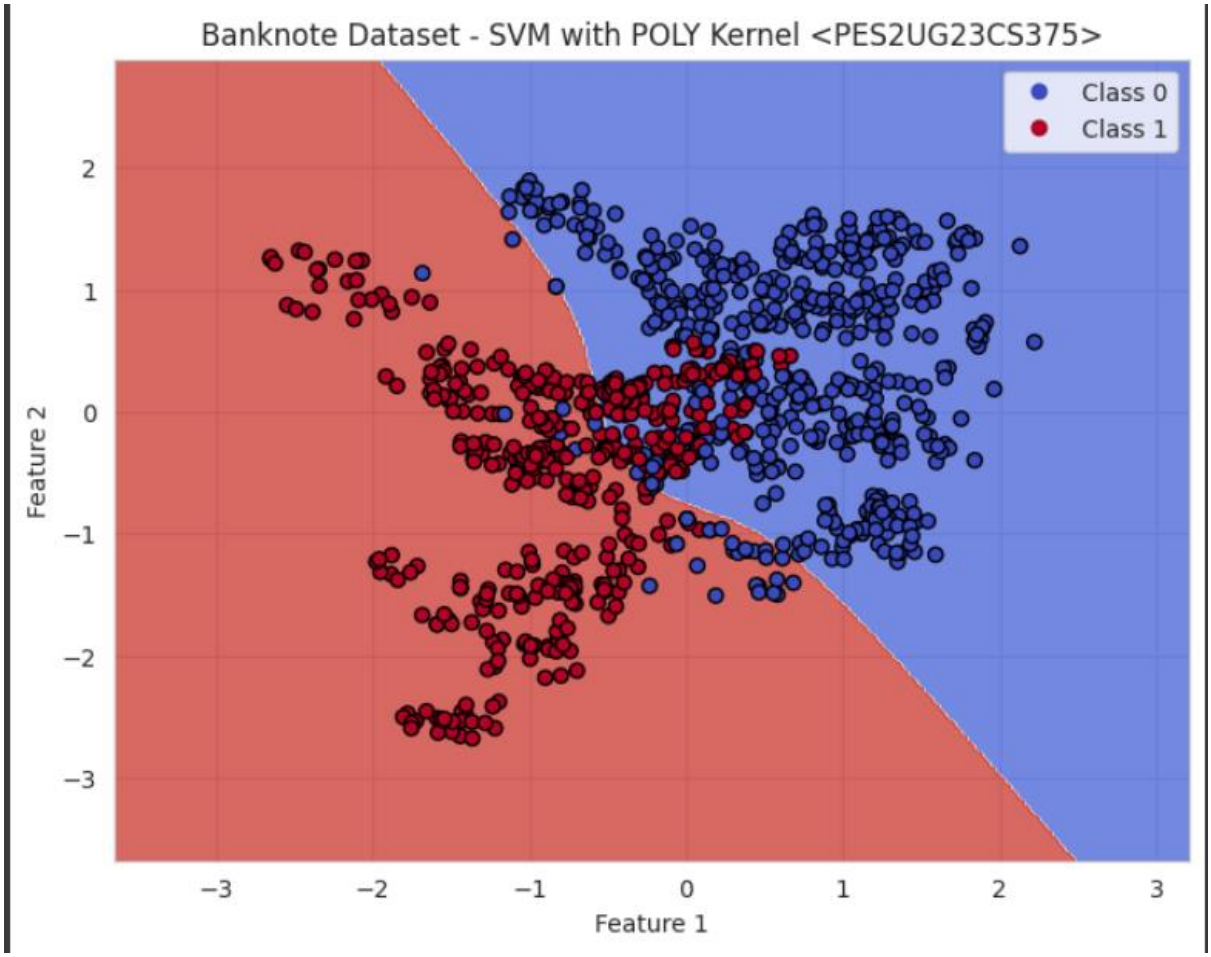

Moons Dataset - SVM with LINEAR Kernel <PES2UG23CS375>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

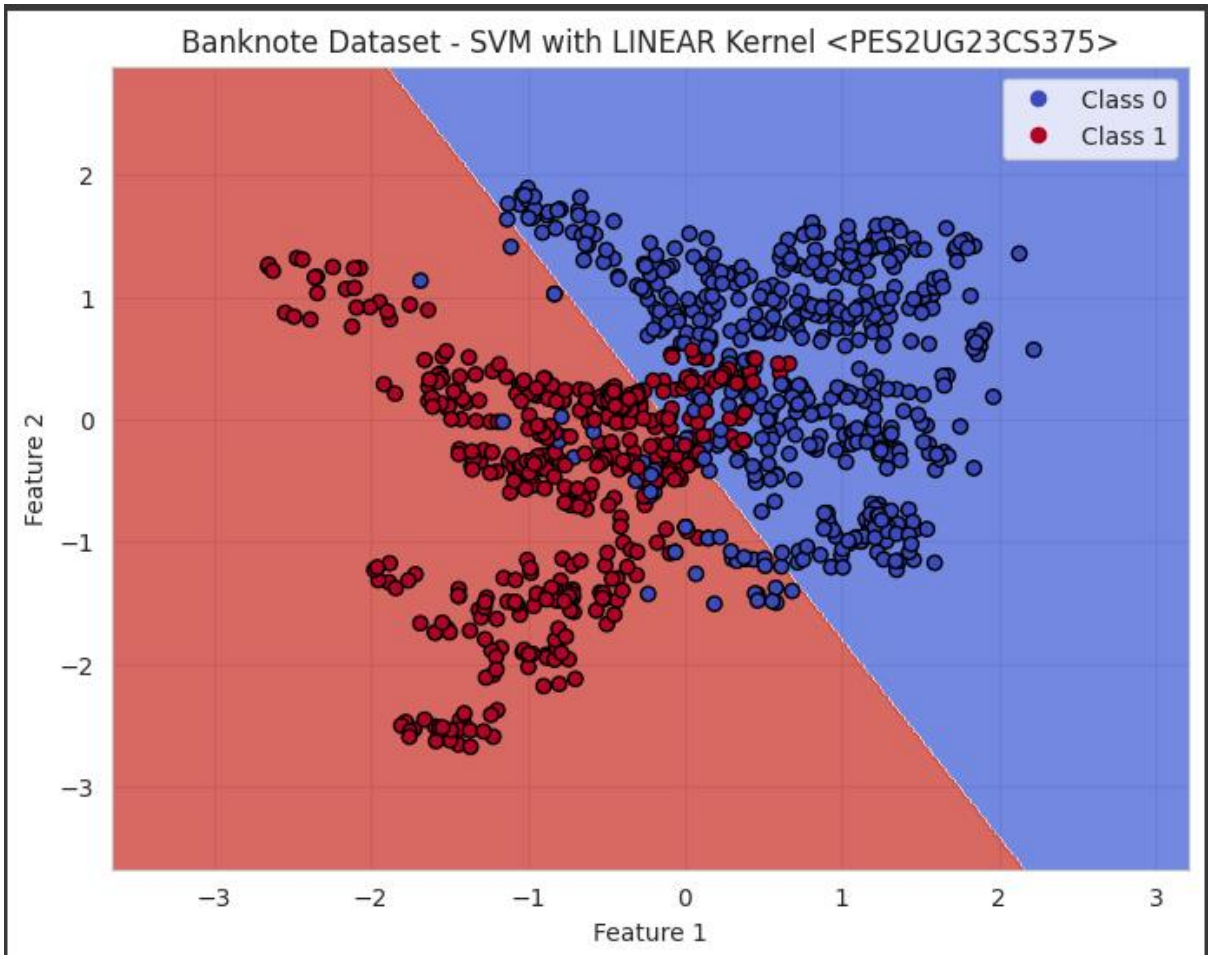

Moons Dataset - SVM with RBF Kernel <PES2UG23CS375>
Class 0
Class 1
Feature 1
Feature 2
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

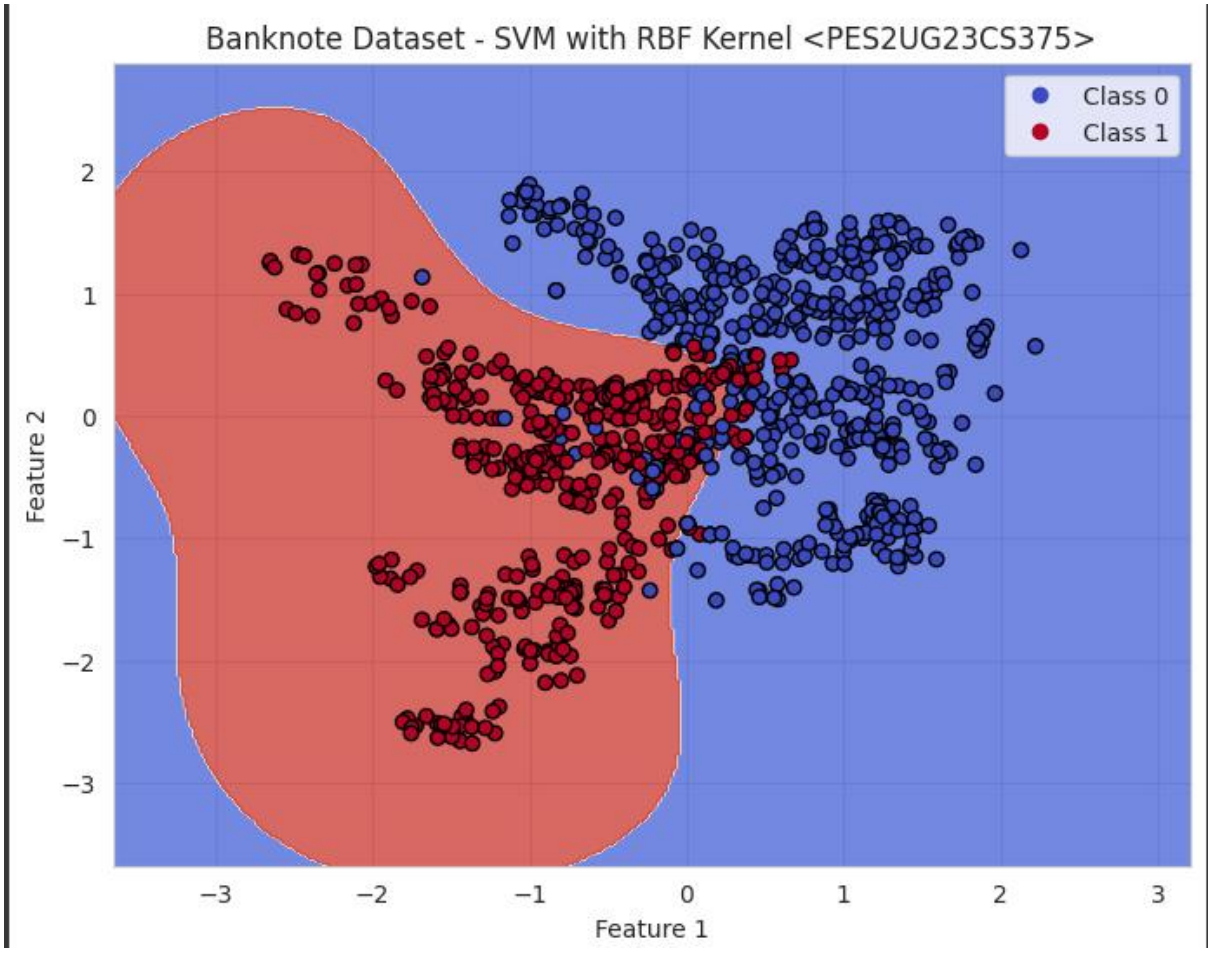

Moons Dataset - SVM with POLY Kernel <PES2UG23CS375>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

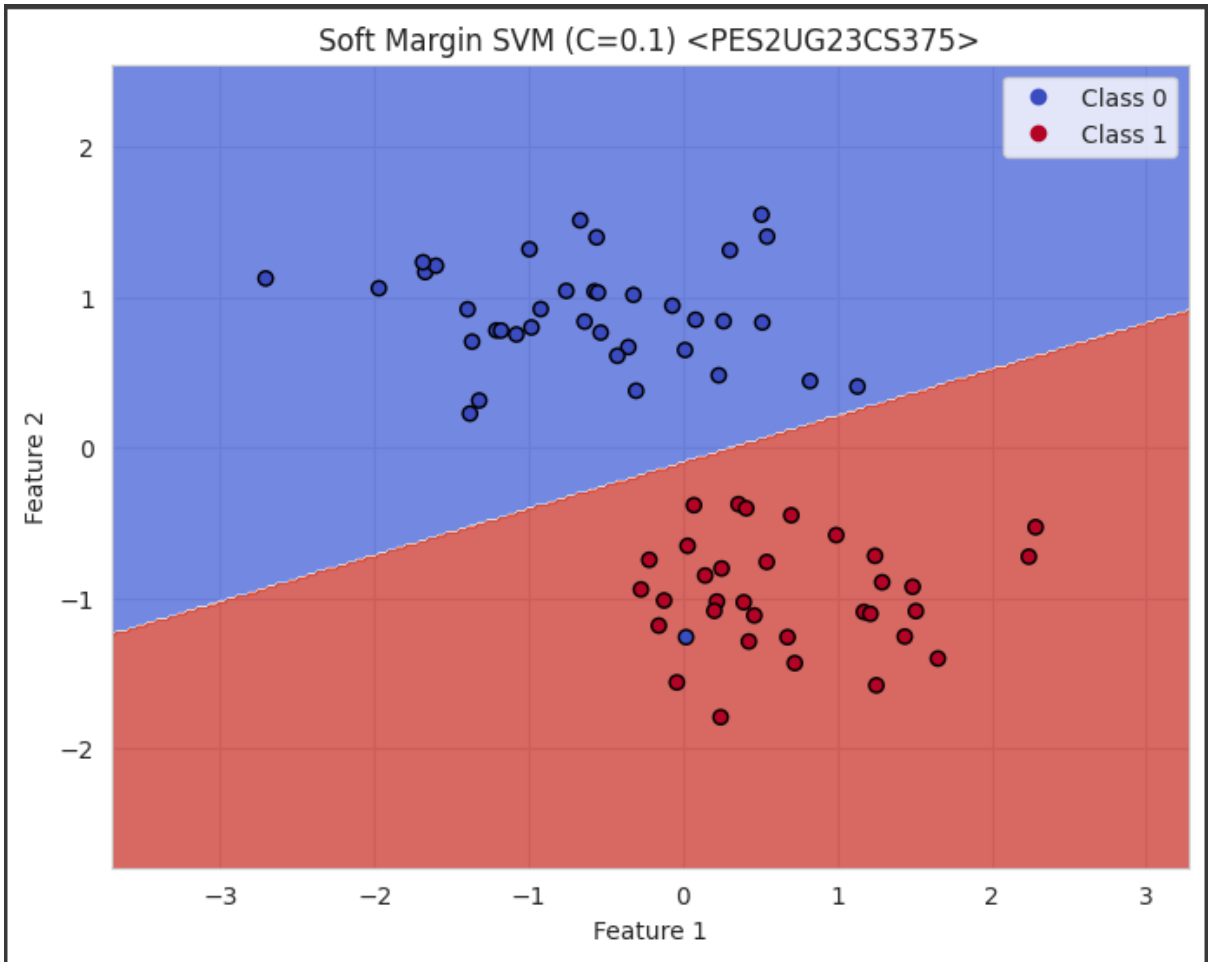

Banknote Authentication Dataset
Skewness
Variance
10
5
0
−5
−10
−15
−6
−4
−2
0
2
4
6

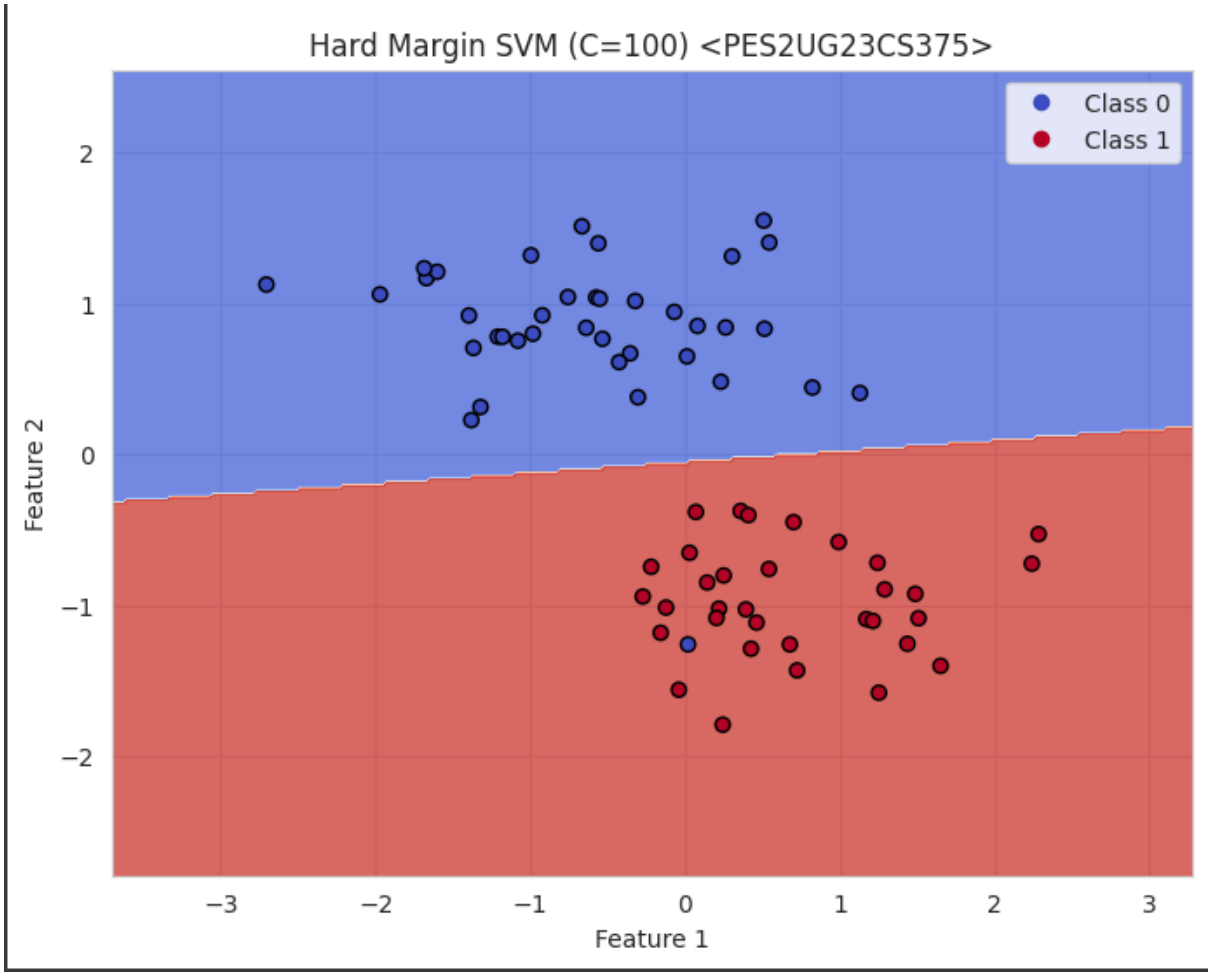
```
SVM with LINEAR Kernel <PES2UG23CS375>
              precision    recall  f1-score   support

      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 <PES2UG23CS375>
              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 <PES2UG23CS375>
              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
```


Banknote Dataset - SVM with POLY Kernel <PES2UG23CS375>
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 LINEAR Kernel <PES2UG23CS375>
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 RBF Kernel <PES2UG23CS375>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


Soft Margin SVM (C=0.1) <PES2UG23CS375>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−2
−1
0
1
2
3


Hard Margin SVM (C=100) <PES2UG23CS375>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−2
−1
0
1
2
3