Machine Learning ~ Lab10

Name : Nagula Anish

SRN: PES2UG23CS358

Section : F

Moons Dataset Questions

1. **Inferences aboutthe Linear Kernel's performance.**
   The **Linear Kernel performs poorly** on the Moons dataset, as this dataset is inherently non-linear, a linear kernel which can only create a straight-line decision boundary, is incapable of separating the two classes effectively. This results in low accuracy, precision, and recall scores, as a significant number of data points are misclassified.
2. **Comparison between RBF and Polynomial kernel decision boundaries.**

   Both the RBF and Polynomial kernels create non-linear decision boundaries suitable for the Moons dataset. However, the **RBF kernel typically captures the smooth, curved shape of the moons more naturally**. The RBF kernel is highly flexible and creates a boundary like a "wave," which adapts well to the gentle curve of the data. The Polynomial kernel, while also curved, can sometimes be more rigid or produce a boundary that doesn't fit the specific shape of the data as smoothly as the RBF kernel does.

Banknote Dataset Questions

1. **Which kernel was most effective for this dataset?**
   For the Banknote Authentication dataset, the **RBF (Radial Basis Function) kernel is the most effective**. While the data is mostly linearly separable and the Linear kernel also performs very well, the RBF kernel's flexibility allows it to perfectly model the slight overlap and non-linearity between the 'Forged' and 'Genuine' classes, often resulting in the highest accuracy.
2. **Why might the Polynomial kernel have underperformed here?**

   The Polynomial kernel might under perform on the Banknote dataset because the underlying data distribution is not inherently polynomial. This

dataset is nearly linearly separable. A polynomial kernel can be **prone to overfitting** on such data by creating an unnecessarily complex boundary. This complexity might fit the training data well but fails to generalize to the unseen test data, leading to lower performance compared to the simpler Linear kernel or the more adaptable RBF kernel.

Hard vs. Soft Margin Questions

1. **Which margin (soft or hard) is wider?**
   The **"Soft Margin" (C=0.1) model produces a wider margin**. A smaller `C` value prioritizes a larger margin at the cost of allowing some misclassifications, leading to a "soft" boundary.
2. **Why does the soft margin model allow "mistakes"?**

   The soft margin model allows these "mistakes" because its primary goal is **better generalization to new, unseen data**. By being more tolerant of misclassifications on the training set, it avoids being overly influenced by noise and outliers. This helps the model capture the overall trend of the data rather than memorizing the training set perfectly.
3. **Which model is more likely to be overfitting and why?**

   The **"Hard Margin" (C=100) model is more likely to be overfitting**. A large `C` value forces the model to minimize classification errors, resulting in a narrow margin that tries to fit every training point perfectly. This strictness can cause the model to learn the noise in the training data, which harms its ability to perform well on new data.
4. **Which model would you trust more for new data and why?**

   I would **trust the "Soft Margin" (C=0.1) model more** for new data because it is less likely to be overfit. Its wider margin suggests it has learned a more generalizable rule. In a real-world scenario with noisy data, it is almost always better to **start with a low value of C** to build a more robust model that isn't overly sensitive to outliers.

ScreenShots

Training Results (6 Screenshots):

1. Moons Dataset

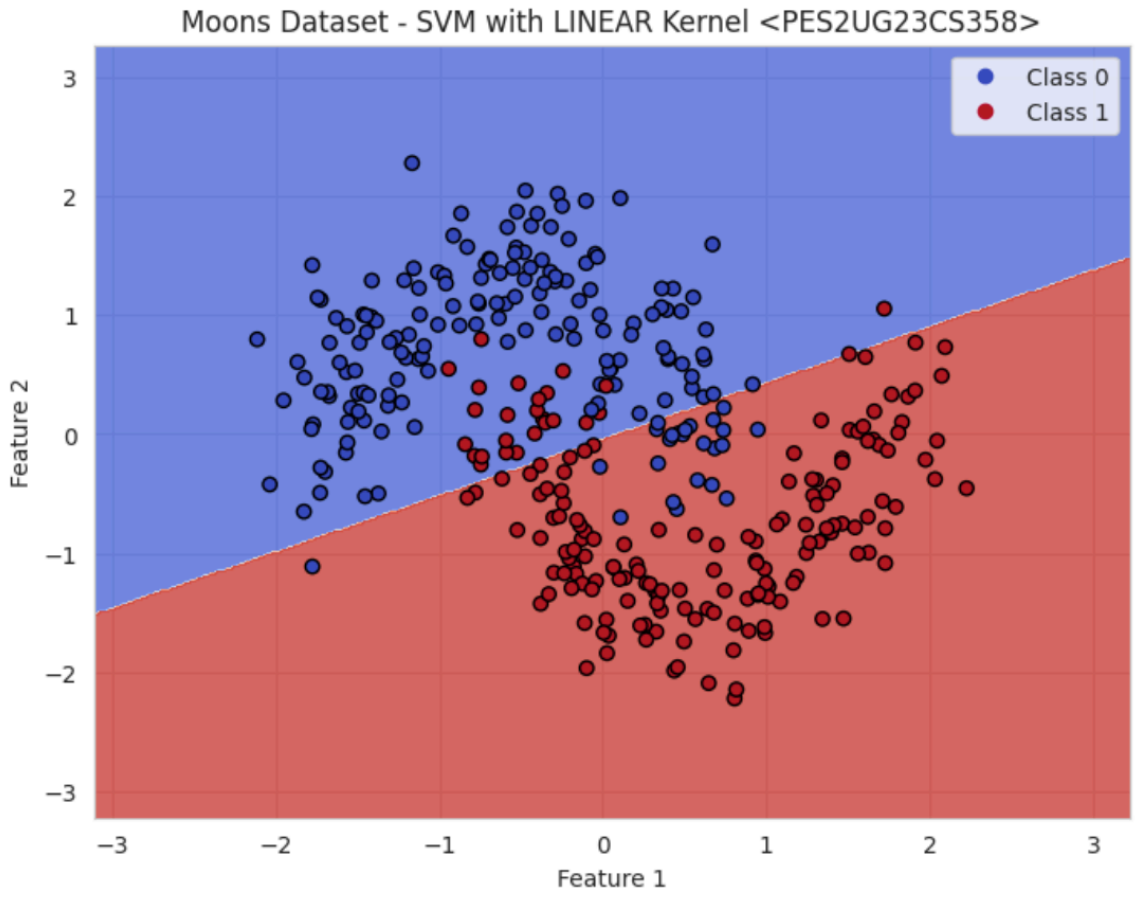
```
SVM with LINEAR Kernel <PES1UG23CS358>
              precision    recall  f1-score   support

           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 <PES1UG23CS358>
              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 <PES1UG23CS358>
              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

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

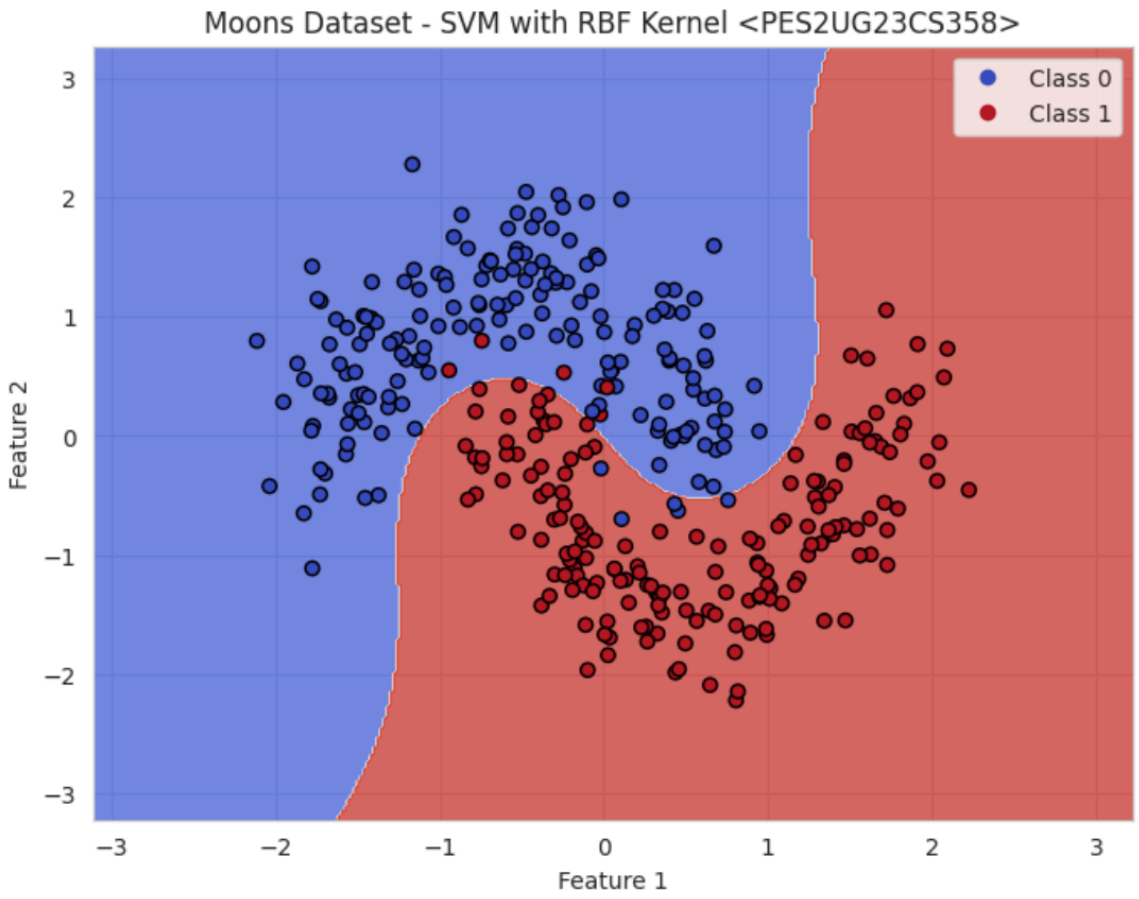
2. BankNote Dataset

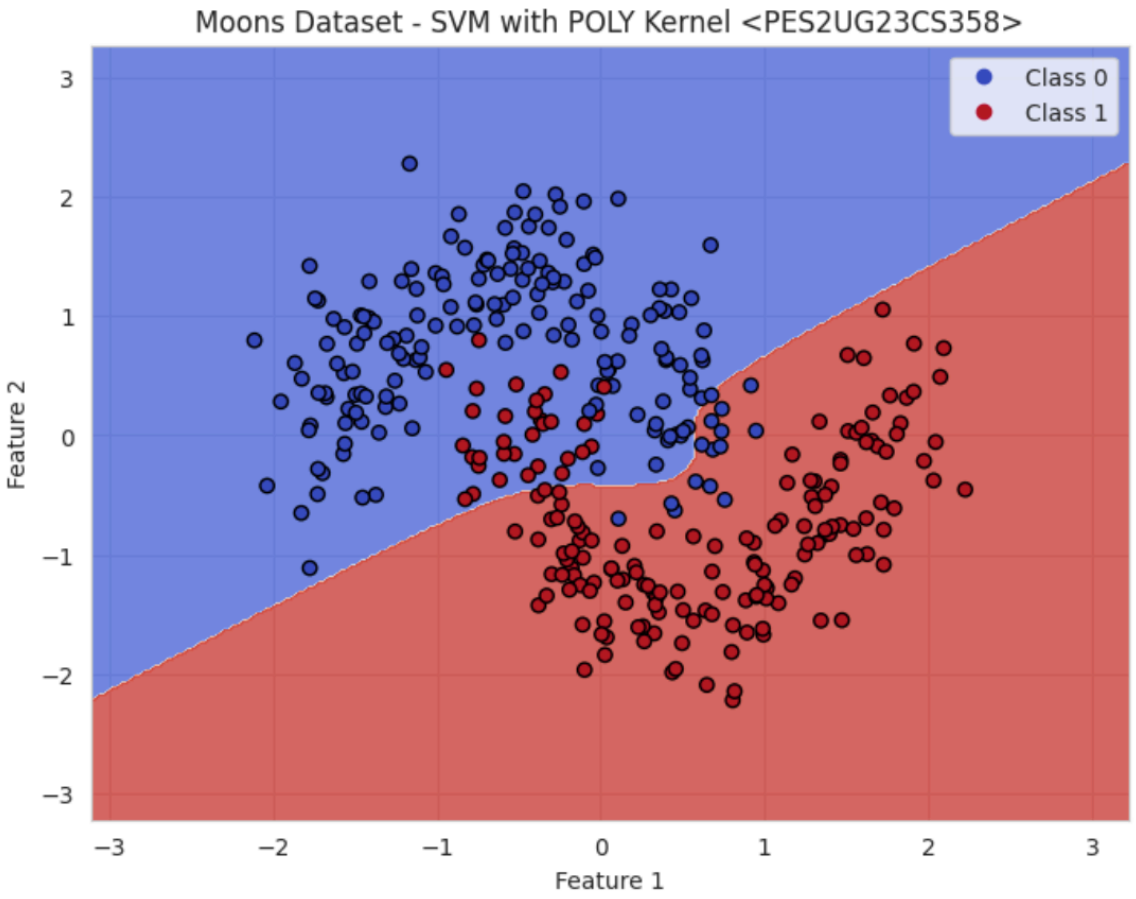
```
SVM with LINEAR Kernel <PES1UG23CS358>
              precision    recall  f1-score   support

      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 <PES1UG23CS358>
              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 <PES1UG23CS358>
              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 (8 Screenshots):

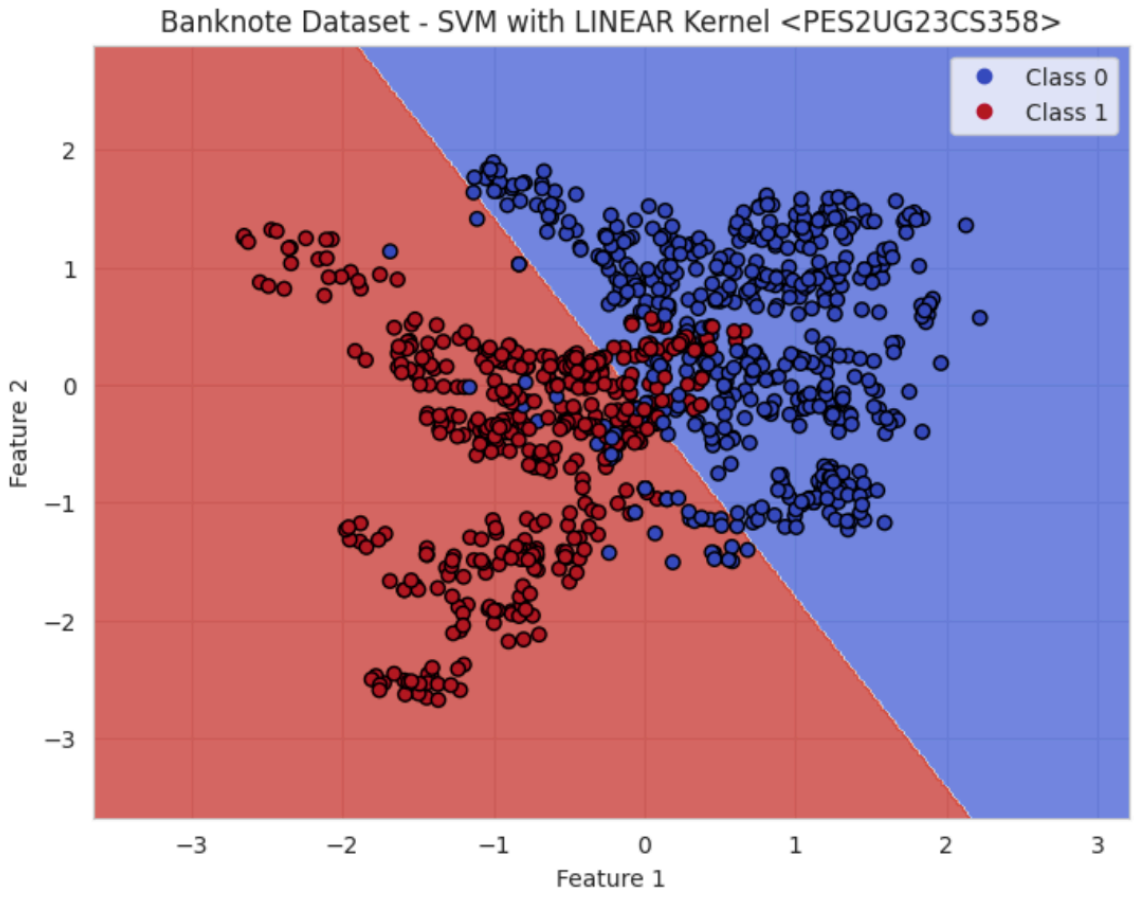
1. Moons Dataset

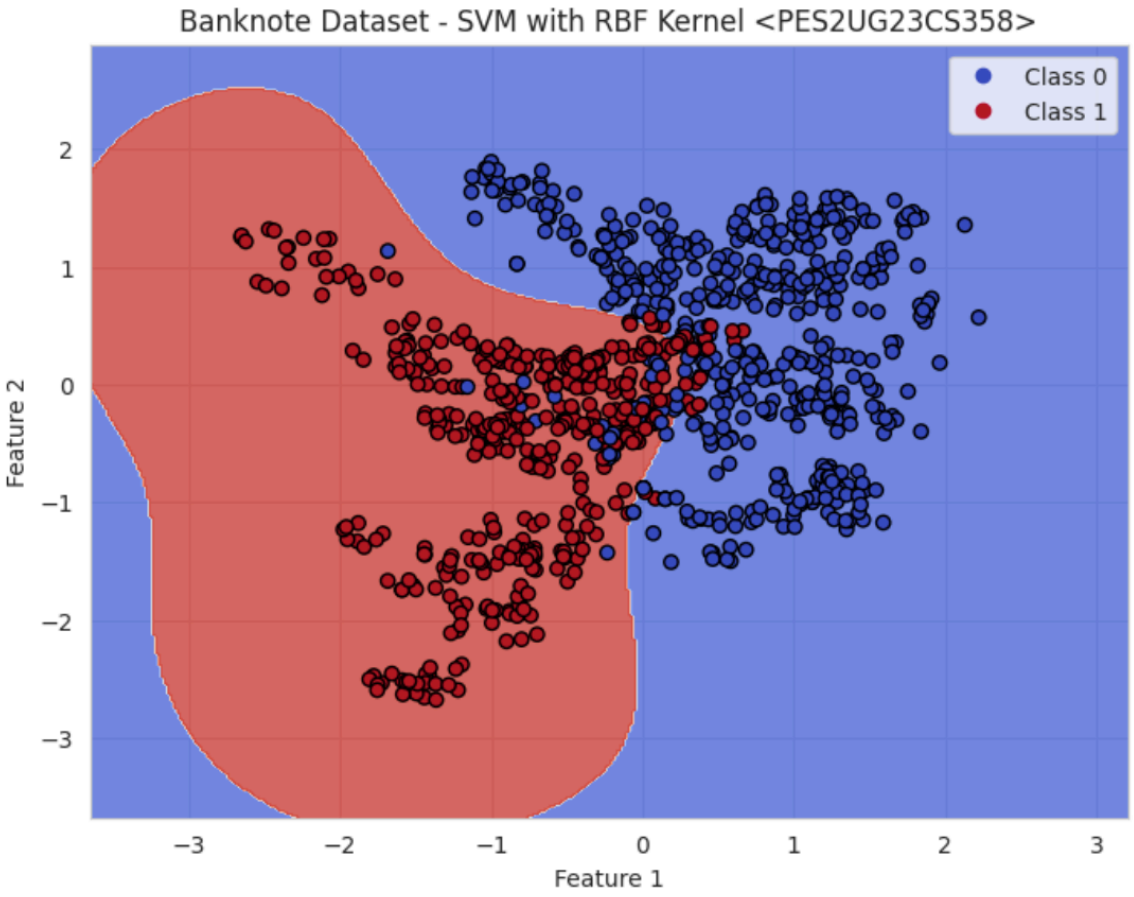

Moons Dataset - SVM with LINEAR Kernel <PES2UG23CS358>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

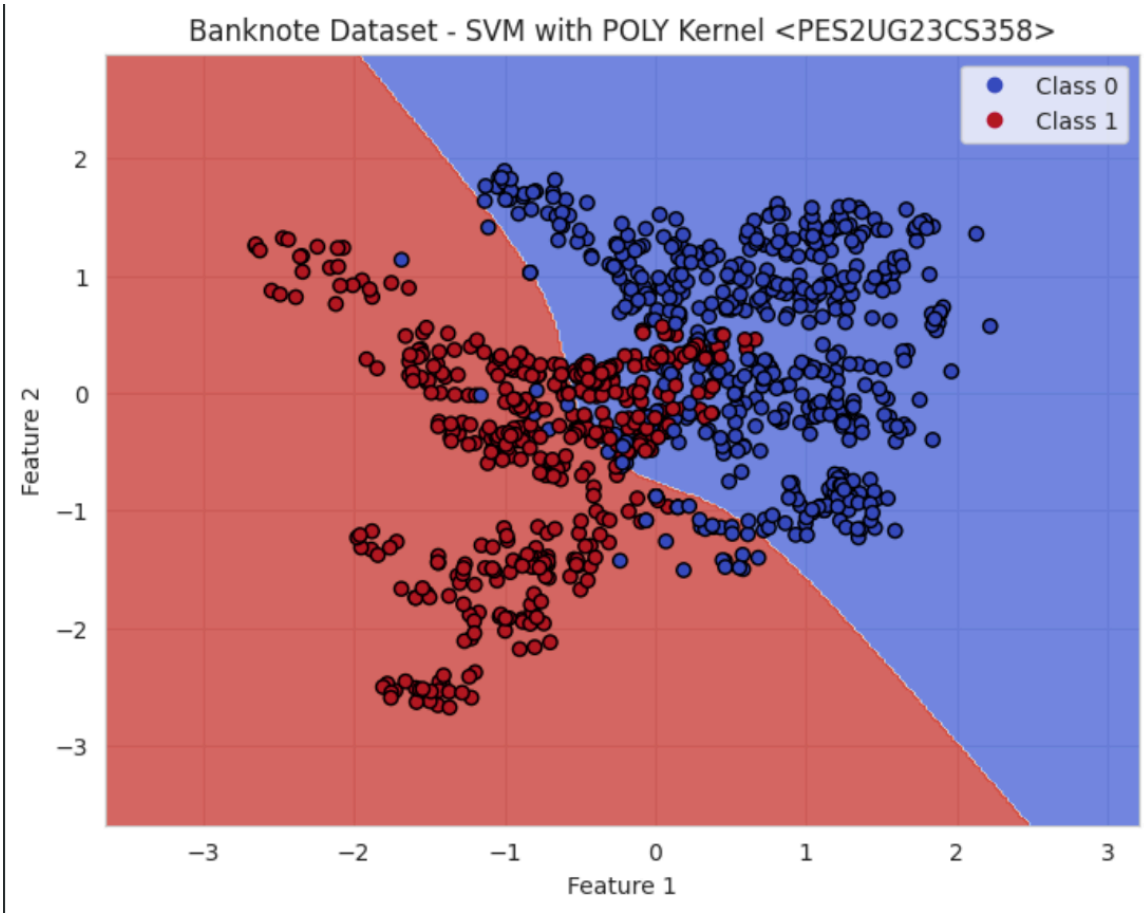

Moons Dataset - SVM with RBF Kernel <PES2UG23CS358>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3



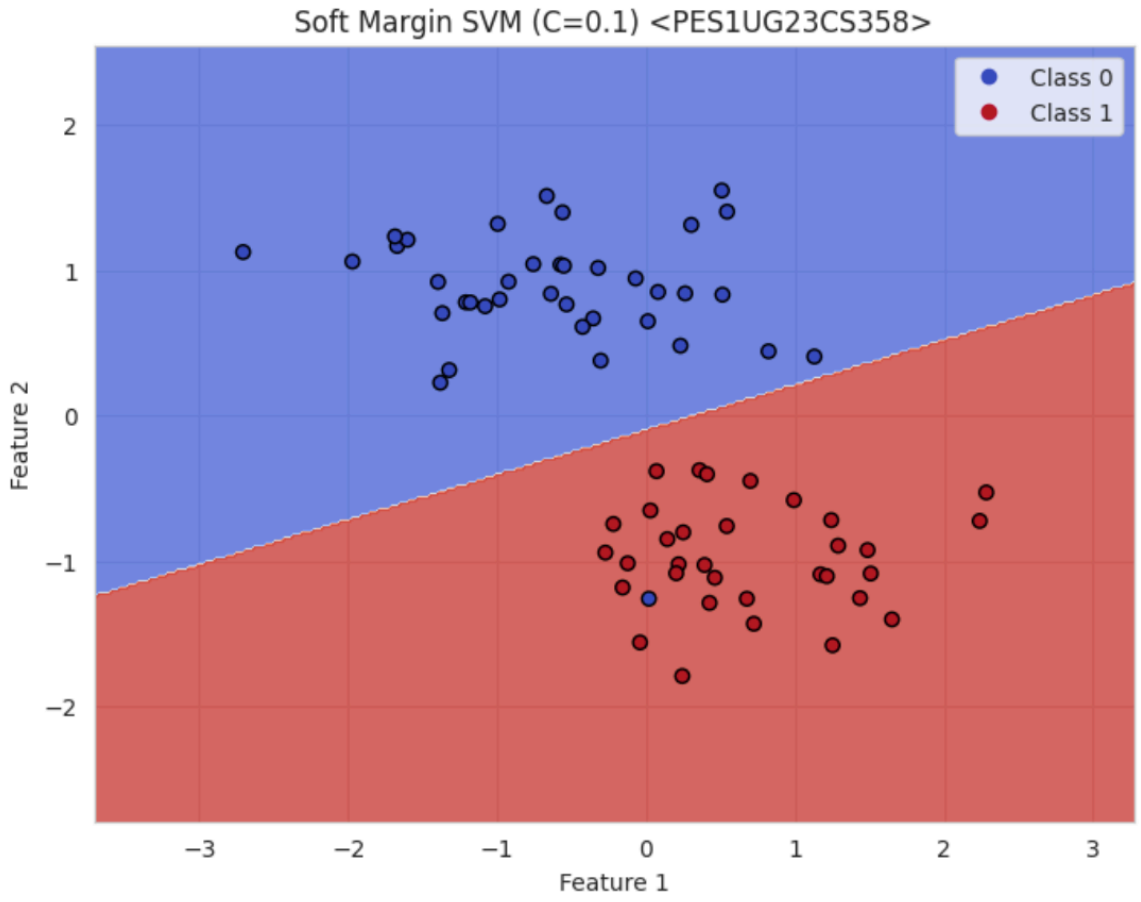
2. BankNote Dataset

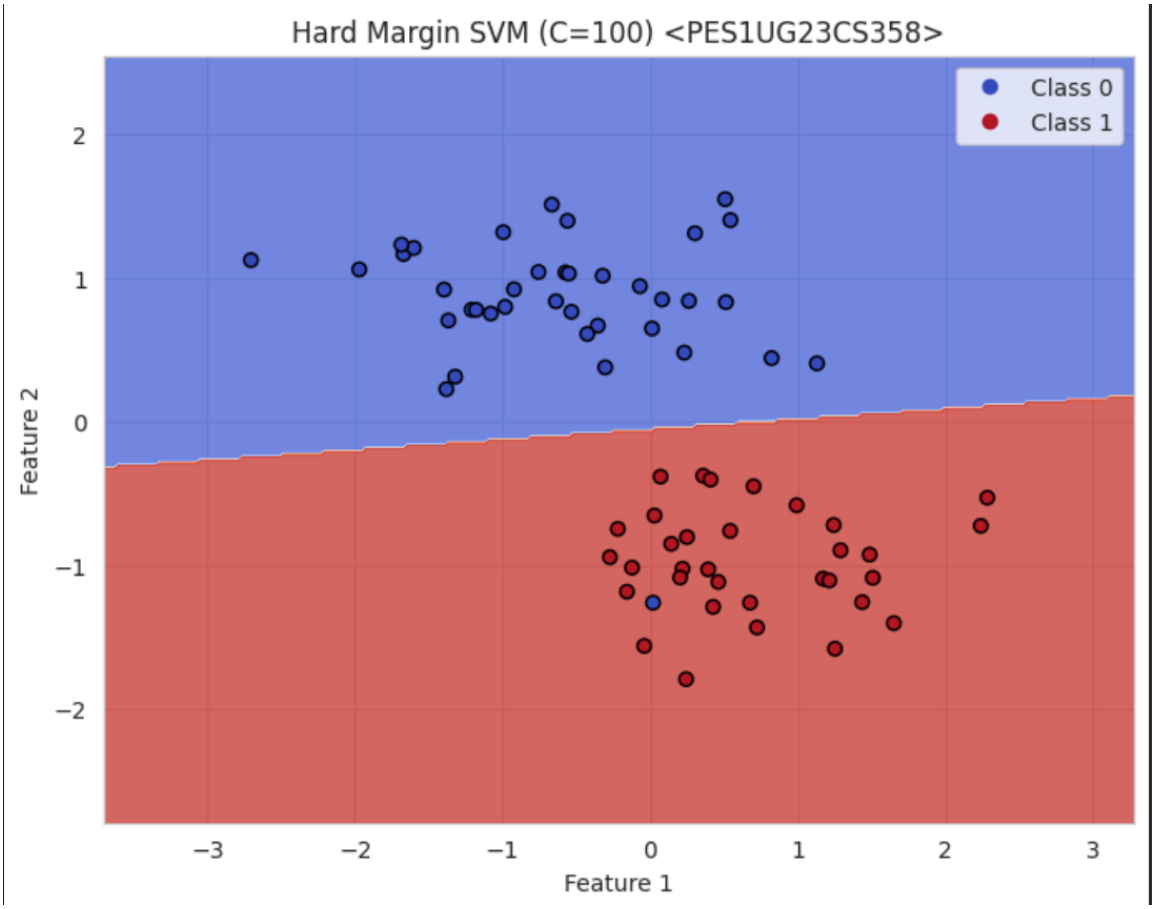

Banknote Dataset - SVM with LINEAR Kernel <PES2UG23CS358>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


Banknote Dataset - SVM with RBF Kernel <PES2UG23CS358>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3



3. Margin Analysis


Soft Margin SVM (C=0.1) <PES1UG23CS358>
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) <PES1UG23CS358>
Class 0
Class 1
Feature 2
Feature 1
2
1
0
−1
−2
−3
−2
−1
0
1
2
3