ML Lab Week 10 — SVM Classifier Lab

Name: Diya Prakash SRN: PES2UG23CS184
Section: 5C CSE

Answers to Analysis Questions:

Moons Dataset (2)

1. Inferences about the Linear Kernel's Performance

The **linear kernel performs poorly** on the Moons dataset because the data are **intrinsically non-linear**—it consists of **two interlocking half-moon shapes** that cannot be separated by a straight line. Since the linear kernel constructs only **a single straight decision boundary**, it **fails to follow the curved geometry** of the classes. As a result, its **decision boundary alignment is weak**, and this translates to **lower accuracy and poorer classification metrics** compared to non-linear kernels.
In essence, the linear kernel's limited flexibility causes **high bias** and **underfitting** on this dataset, since it cannot adapt to the non-linear separations required.

2. RBF vs Polynomial Decision Boundaries

- **RBF Kernel:** The RBF (Radial Basis Function) kernel produces **highly flexible and localized decision regions**, meaning it adjusts its boundary based on the local structure of the data. This makes it particularly effective at **capturing the fine-grained, curved shapes** seen in the Moons dataset. The RBF kernel's Gaussian nature allows it to smoothly adapt to the contours of each moon, fitting the non-linear boundaries naturally even with **default hyperparameters**.
- **Polynomial Kernel (degree = 3 by default):** The polynomial kernel, on the other hand, constructs a **global polynomial surface** across the entire feature space. It can capture some degree of curvature, but its performance **depends strongly on tuning** parameters such as the **degree** and **coef0**. With default settings, the polynomial kernel often **fails to capture the intricate curvature** of the Moons data, resulting in less optimal boundaries.

Conclusion:
While both are non-linear, the **RBF kernel aligns more naturally** with the Moons dataset's geometry under default conditions. The **polynomial kernel can approximate** similar patterns but typically requires **manual hyperparameter tuning** (e.g., increasing degree or adjusting coef0) to reach comparable performance.

Banknote Dataset (2)

1. Which Kernel Was Most Effective?

The **RBF kernel** was the **most effective** on the Banknote dataset, achieving approximately **0.93 accuracy** and the **highest per-class F1-scores** among all tested kernels.
This suggests that the banknote data contain **nonlinear patterns and local variations** that are well

captured by the RBF's localized similarity measure. The RBF kernel's ability to model subtle nonlinear relationships makes it better suited for differentiating between **authentic and counterfeit** notes.

2. Why Might the Polynomial Kernel Have Underperformed?

The **polynomial kernel** underperformed mainly due to a **mismatch in inductive bias** — the assumptions it makes about data structure. The **relationships among features** in the banknote dataset, such as **variance, skewness, and kurtosis**, are **not well represented by a low-degree polynomial function**.
Additionally, the **default polynomial hyperparameters** (degree and coef0) are often **suboptimal** for this dataset's feature distribution.
Unlike the RBF kernel, which can **flexibly model clustered or nonlinear pockets of data**, the polynomial kernel tends to either **underfit (if the degree is too low)** or **overfit (if too high)**.
Thus, RBF's **localized adaptability** allows it to generalize better, explaining its superior performance compared to the polynomial kernel.

Hard vs Soft Margin (4)

1. Which Margin Is Wider?

The **soft margin (C = 0.1)** produces the **wider margin**.
A lower C value relaxes the model's strictness, encouraging the SVM to **maximize the separation gap** between classes even if it misclassifies some training points. This creates **a broader, more forgiving boundary** that improves generalization.

2. Why Does the Soft Margin Model Allow "Mistakes"?

In a soft-margin SVM, a **small C value** reduces the **penalty for misclassification**, effectively **introducing slack variables**.
These slack variables allow some data points to fall on the wrong side of the margin (or even the boundary itself). This tolerance enables the model to **focus on overall structure rather than every individual point**, improving its robustness to **noisy or overlapping data**.
In short, the model **intentionally allows some mistakes** to achieve a larger, more stable margin.

3. Which Model Is More Likely to Overfit and Why?

The **hard margin model (C = 100)** is **more likely to overfit** because it heavily penalizes any misclassification. This strictness forces the decision boundary to **perfectly separate** the training data—even if some points are **outliers or noisy samples**.
As a result, the model's margin becomes **narrower**, and it starts fitting to the training data's noise rather than the underlying trend. Hence, while it may achieve perfect training accuracy, its **test accuracy suffers** due to poor generalization.

4. Which Model Would You Trust More for New Noisy Data and Why?

The **soft margin model (low C)** would be **more trustworthy** for **new or noisy data**.
Because it allows a few training errors, it avoids overfitting and maintains **better generalization** capability. A smaller C encourages the model to **learn smoother, more general decision boundaries**, making it resilient to random fluctuations or measurement noise.
In practical tuning, one usually **starts with a low C** and increases it gradually if the model **underfits**, achieving a good trade-off between margin width and training accuracy.

Summary Insight:
Across both datasets, **non-linear kernels (especially RBF)** consistently outperform linear and low-degree polynomial ones when data exhibit curved or clustered boundaries. Similarly, **soft margins** strike a better generalization balance in noisy scenarios, while **hard margins** risk overfitting due to their rigidity.

Required Screenshots:

Training Results (6 Screenshots): Capture the classification report output for each model.

Moons Dataset:

```
SVM with LINEAR Kernel <PES2UG23CS184>
              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 <PES2UG23CS184>
              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 <PES2UG23CS184>
              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:

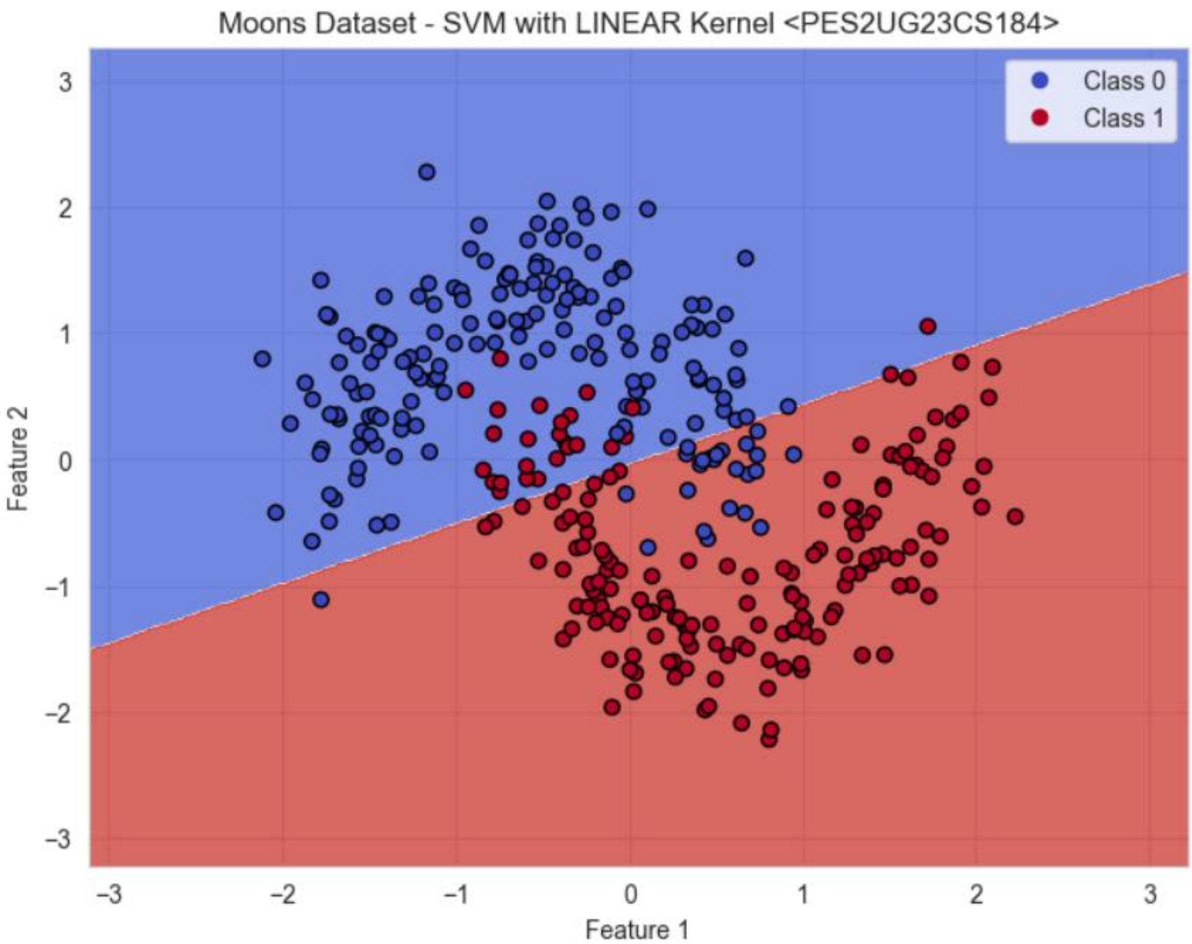
```
SVM with LINEAR Kernel <PES2UG23CS184>
              precision    recall  f1-score   support

      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
```

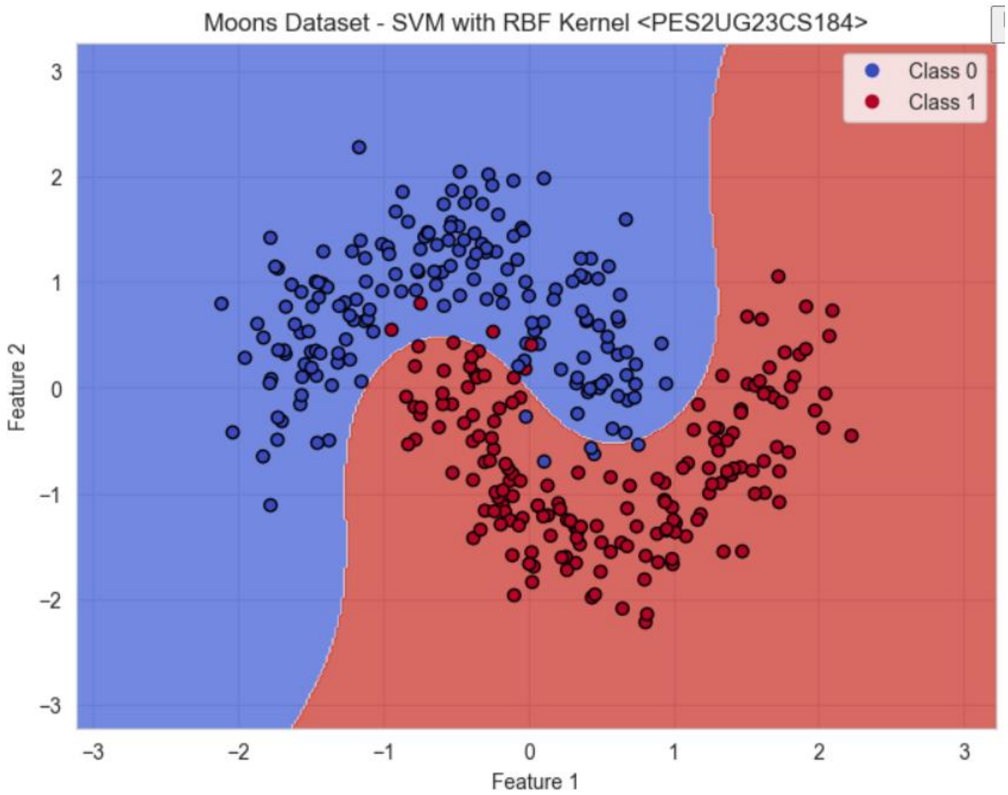
```
SVM with RBF Kernel <PES2UG23CS184>
              precision    recall  f1-score   support

      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 <PES2UG23CS184>
              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): Capture the plot for each model's decision boundary

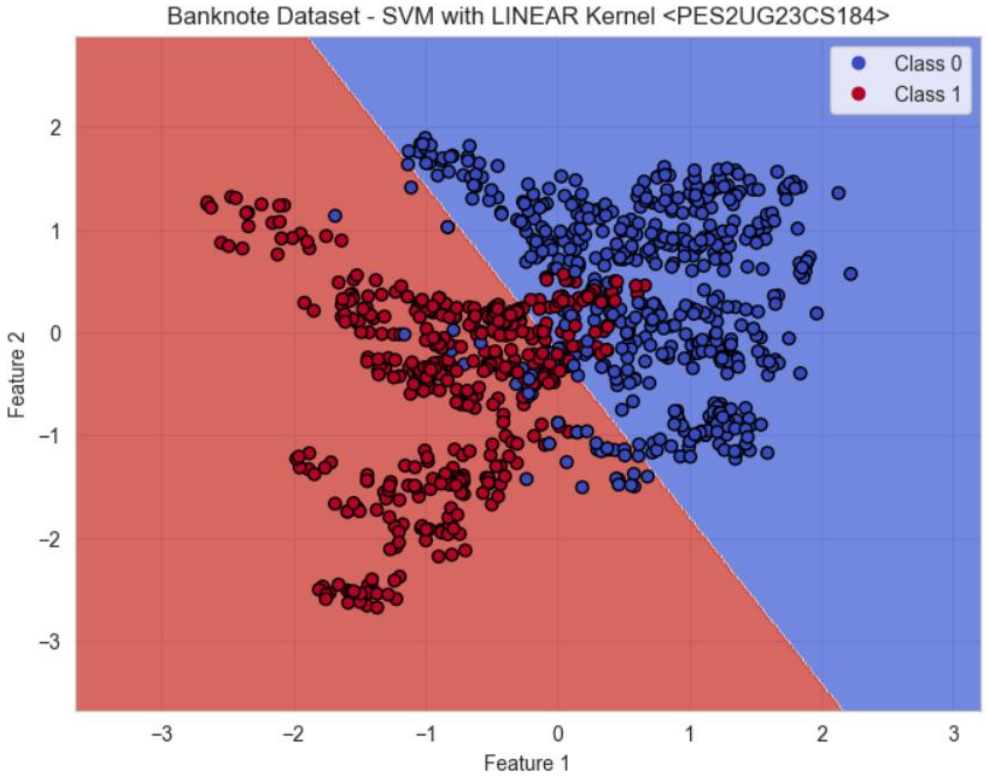
Moons Dataset (3 plots):

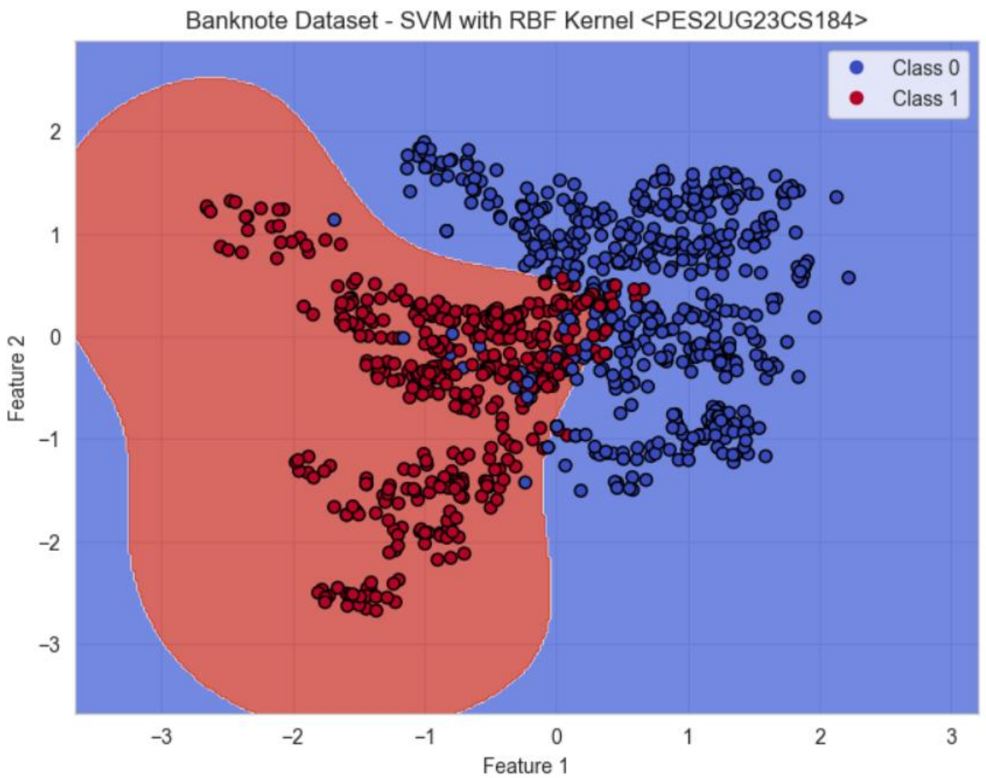


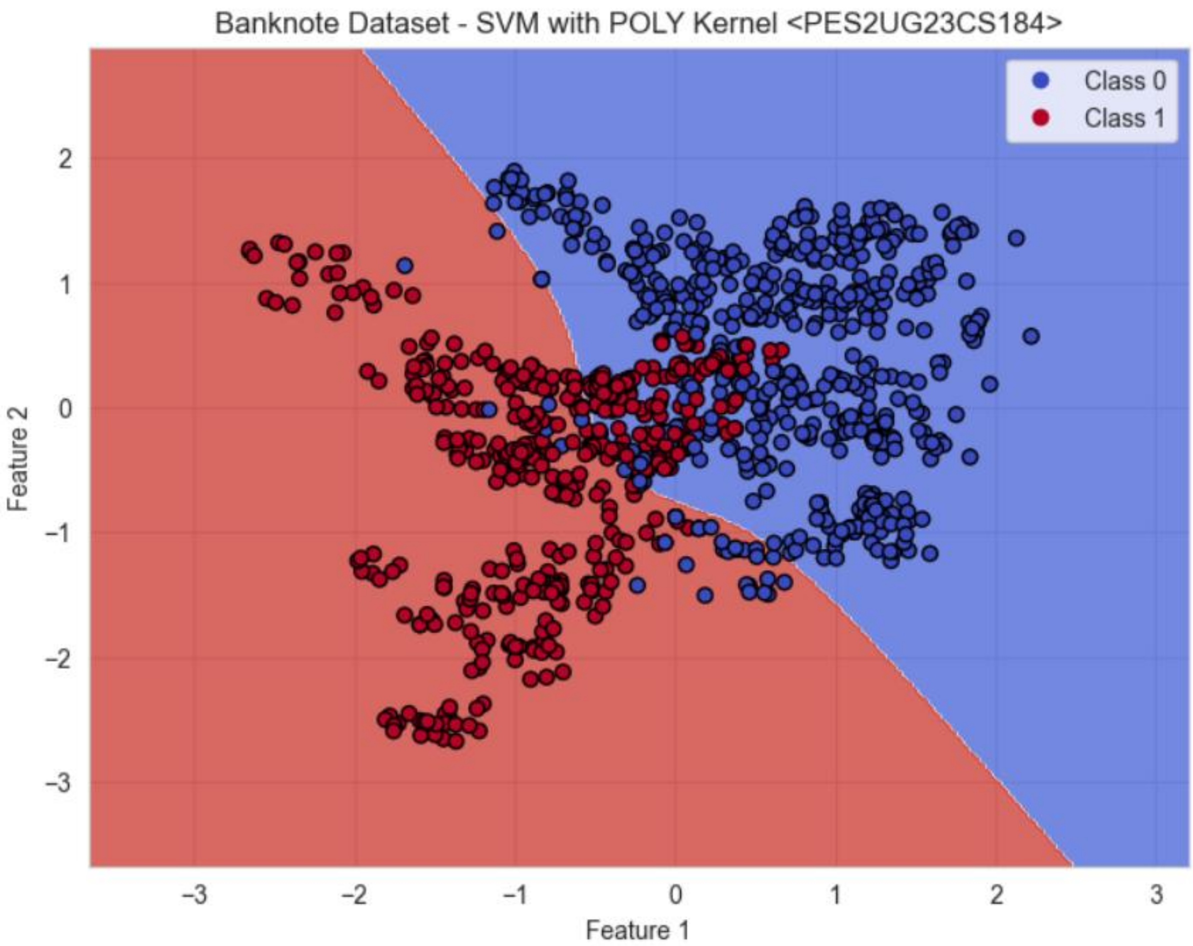

Moons Dataset - SVM with RBF Kernel <PES2UG23CS184>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


Moons Dataset - SVM with POLY Kernel <PES2UG23CS184>
Class 0
Class 1
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Banknote Dataset (3 plots):

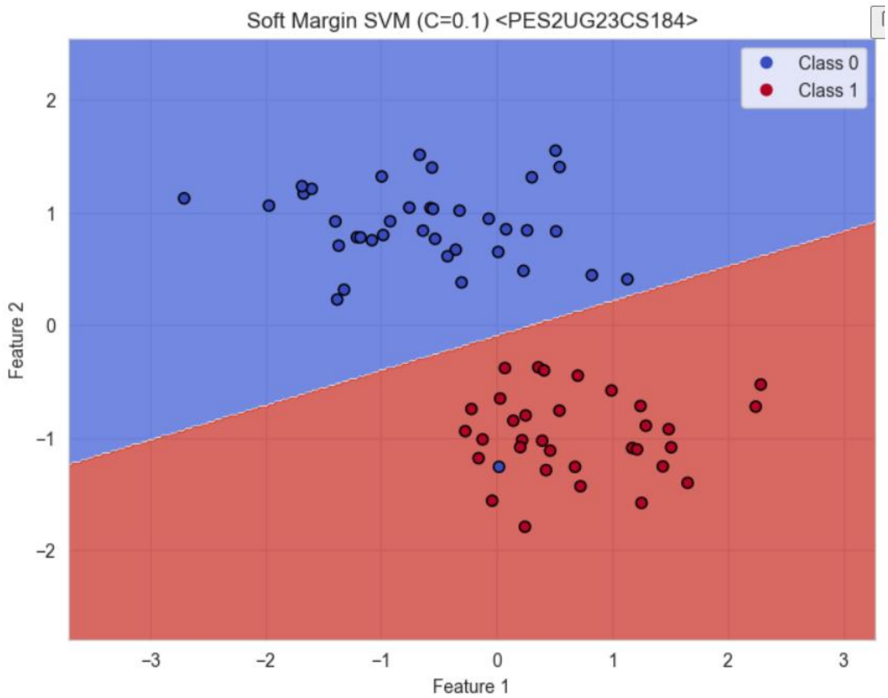






Margin Analysis (2 plots):

Soft Margin SVM (C = 0.1)



Hard Margin SVM (C = 100):