Week 10 SVM

NAME : MOULYA K A

SRN: PES2UG23CS351

SECTION:F

COURSE:MACHINE LEARNING

DATE:12-10-2025

Analysis Questions:

Moons Dataset Questions:

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

Based on the classification metrics and decision boundary visualization, the Linear kernel shows moderate performance on the dataset. The model achieves an accuracy of approximately 87%, with precision and recall values that are fairly balanced between the two classes (Class 0: precision 0.85, recall 0.89; Class 1: precision 0.89, recall 0.84). The macro and weighted averages are also around 0.87, indicating stable but not optimal classification. The decision boundary generated by the Linear kernel is a single straight line, which is unable to capture the non-linear structure of the data. As a result, several points near the curved regions of the class boundaries are misclassified. Overall, while the Linear kernel provides a decent baseline model, it lacks the flexibility needed to accurately separate non-linearly distributed data, leading to a noticeable number of misclassifications compared to non-linear kernels.

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

When comparing the decision boundaries of the RBF and Polynomial kernels, it is evident that the RBF kernel captures the shape of the data more naturally. The RBF kernel produces a smooth, curved boundary that wraps around the two moon-shaped clusters effectively. This results in excellent classification metrics, including a precision of 0.95 for Class 0 and 1.00 for Class 1, a recall of 1.00 for Class 0 and 0.95 for Class 1, and an overall accuracy of 97%. In contrast, the Polynomial kernel also produces a non-linear boundary, but it is less smooth, with sharper angles and a less natural fit to the underlying data structure. Although its performance (accuracy ≈ 89%) is better than the Linear kernel, it falls short of the RBF kernel. Polynomial decision functions can sometimes become too rigid or oscillatory depending on their degree, which limits their adaptability to complex patterns. Consequently, the RBF kernel demonstrates superior flexibility and fit, resulting in higher accuracy and fewer misclassifications compared to the Polynomial kernel.

Banknote Dataset Questions:

1. **In this case, which kernel appears to be the most effective?**
   Based on the classification metrics and decision boundary plots, the RBF kernel is the most effective for the Banknote Authentication dataset. It achieves the highest accuracy of 93%, outperforming both the Linear (88%) and Polynomial (85%) kernels. The precision and recall values for both classes are balanced and consistently high (around 0.93–0.94), indicating robust classification performance. The decision boundary produced by the RBF kernel is smooth and flexible, allowing it to accurately separate data points in regions where the classes are not linearly separable. In contrast, the Linear kernel provides a single straight-line boundary, which fails to capture non-linear patterns in the data, while the Polynomial kernel produces a more rigid boundary that misclassifies several clusters. Overall, the RBF kernel adapts better to the complex structure of the dataset, resulting in superior performance.

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

The Polynomial kernel performs well on the Moons dataset because the class boundaries there are smooth and moderately non-linear, which can be effectively modeled using a polynomial decision function of appropriate degree. However, in the Banknote dataset, the distribution of data is more complex and irregular, with no single smooth curve separating the classes. Polynomial kernels are not as flexible as RBF kernels; they either underfit when the degree is too low or overfit when the degree is too high, making them less effective for capturing localized variations in the data. As a result, the Polynomial kernel struggles to fit the complex decision boundary required for the Banknote dataset, leading to lower accuracy and more misclassifications compared to its performance on the Moons dataset.

Hard vs. Soft Margin Questions:

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

From the two plots, it is evident that the Soft Margin SVM (C = 0.1) produces a wider margin compared to the Hard Margin SVM (C = 100). A lower value of C encourages the model to allow more flexibility and focus on maximizing the margin, even if it means some points may lie within the margin or be misclassified. In contrast, the Hard Margin SVM with a high C value tries to classify all training points correctly, resulting in a narrower margin. This can be seen in the second plot, where the decision boundary is very close to the nearest points, leaving little room for tolerance.

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?**
   In the Soft Margin SVM plot, some data points lie either inside the margin or even on the wrong side of the decision boundary. This happens because the SVM is designed to balance margin maximization and classification accuracy. By using a lower C value, the model allows for some misclassifications or margin violations to achieve a larger margin overall. The primary goal of the Soft Margin SVM is not to classify every training point perfectly, but rather to find a decision boundary that generalizes well to unseen data. Allowing a few “mistakes” helps the model avoid being overly sensitive to individual points, especially if they are noisy or outliers.

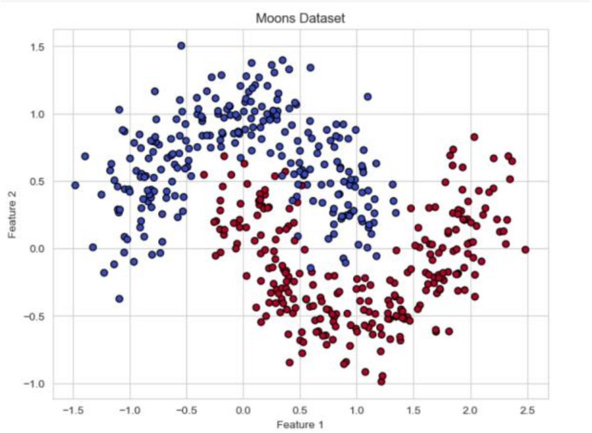
3. **Which of these two models do you think is more likely to be overfitting to the training data? Explain your reasoning.**
   Between the two models, the Hard Margin SVM (C = 100) is more likely to overfit the training data. By setting C to a high value, the model penalizes misclassifications heavily, forcing the decision boundary to fit all training points as accurately as possible. This often leads to a very tight margin that can capture noise or outliers as if they were important patterns, thereby reducing generalization. On the other hand, the Soft Margin model tolerates some errors, leading to a smoother decision boundary that is less likely to overfit.

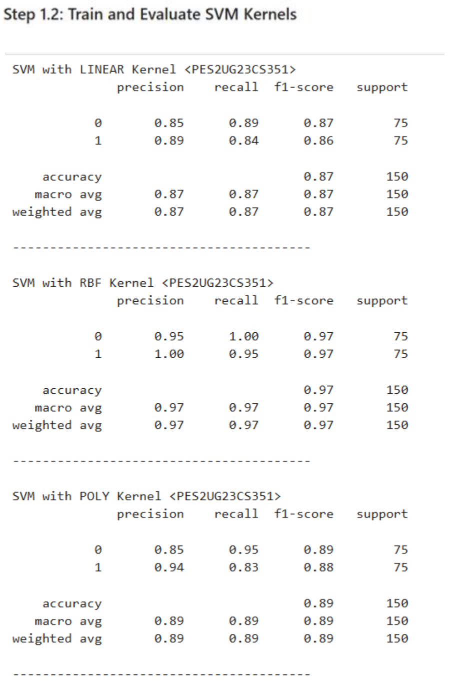
4. **Imagine you receive a new, unseen data point. Which model do you trust more to classify it correctly? Why? In a real-world scenario where data is often noisy, which value of C (low or high) would you generally prefer to start with?**
   For new, unseen data points, the Soft Margin SVM (C = 0.1) is generally more trustworthy for correct classification. This is because the wider margin acts as a buffer, improving generalization performance, especially when the data is noisy. In real-world scenarios, data is often not perfectly separable and may contain outliers or measurement errors. In such cases, it is usually better to start with a lower C value, which prioritizes a larger margin and allows some flexibility. A high C value might lead to a model that performs very well on training data but poorly on unseen data due to overfitting.

Screenshots:

Training Results:

Moons Dataset:

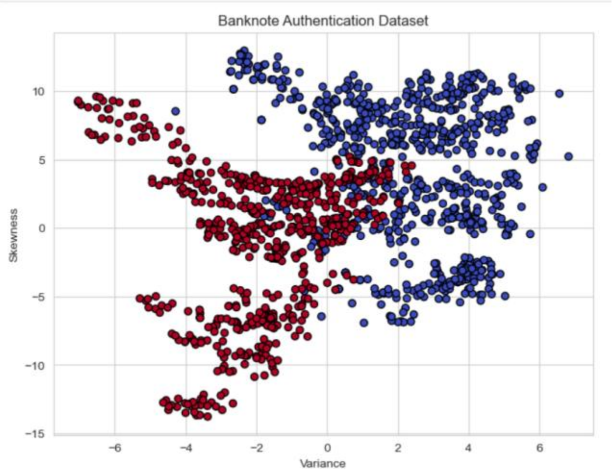


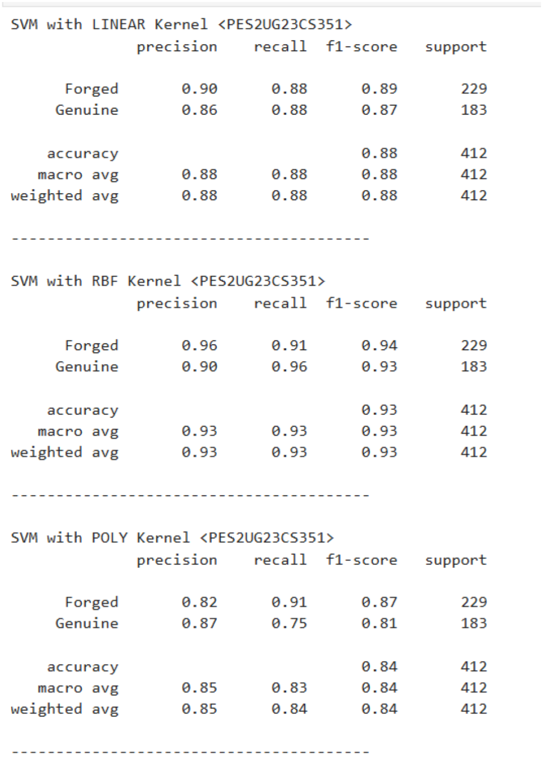


Step 1.2: Train and Evaluate SVM Kernels

```
SVM with LINEAR Kernel <PES2UG23CS351>
              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 <PES2UG23CS351>
              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 <PES2UG23CS351>
              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:

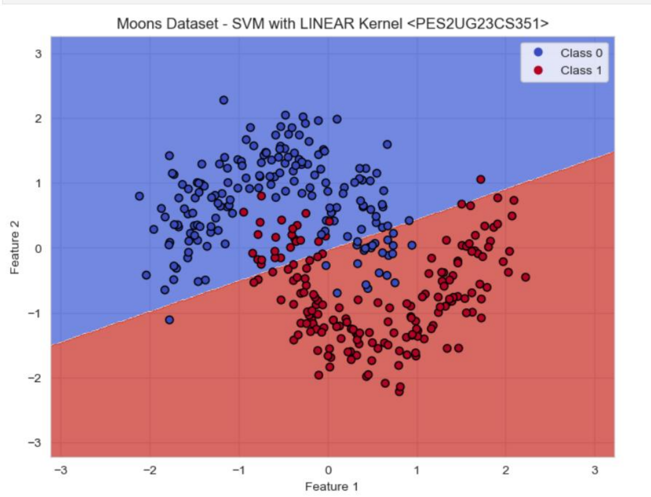


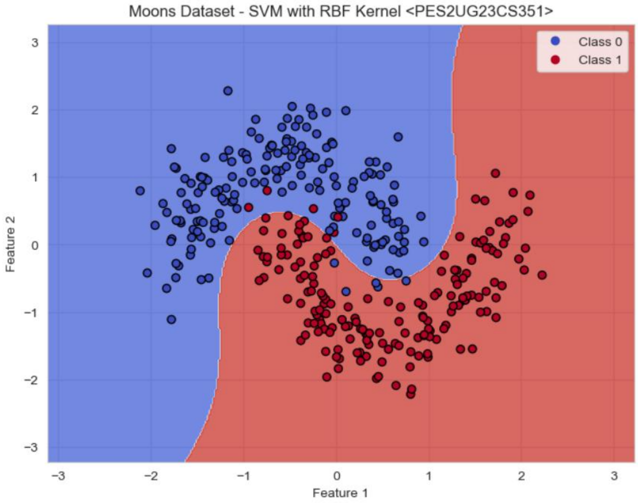


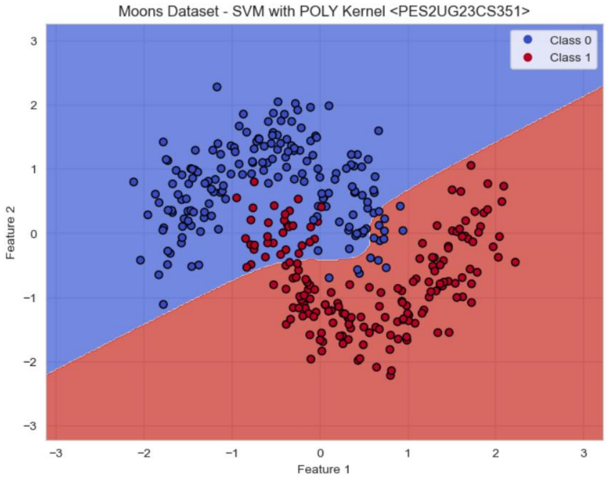
```
SVM with LINEAR Kernel <PES2UG23CS351>
              precision    recall  f1-score   support

      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 <PES2UG23CS351>
              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 <PES2UG23CS351>
              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

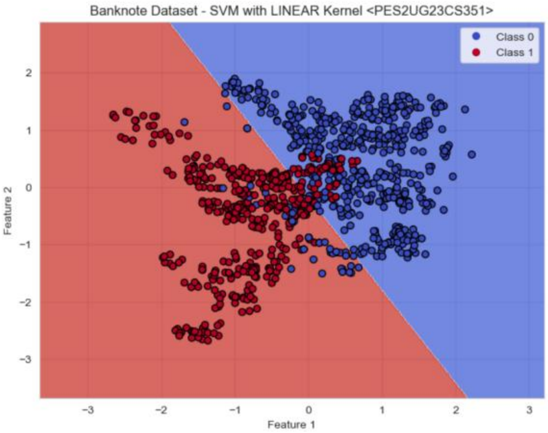
Moons Dataset:

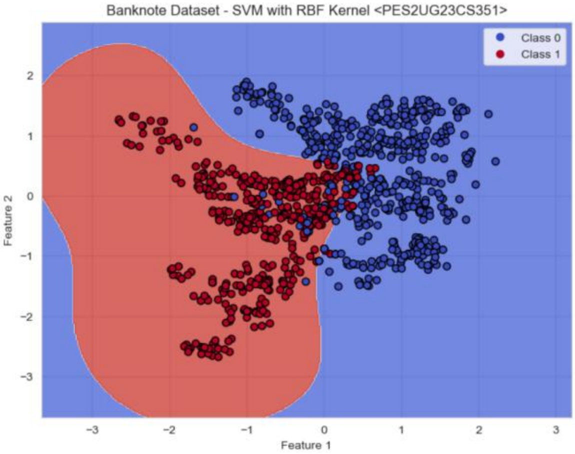


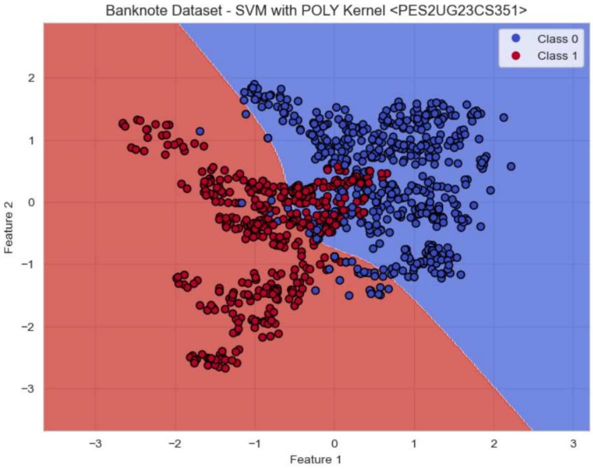




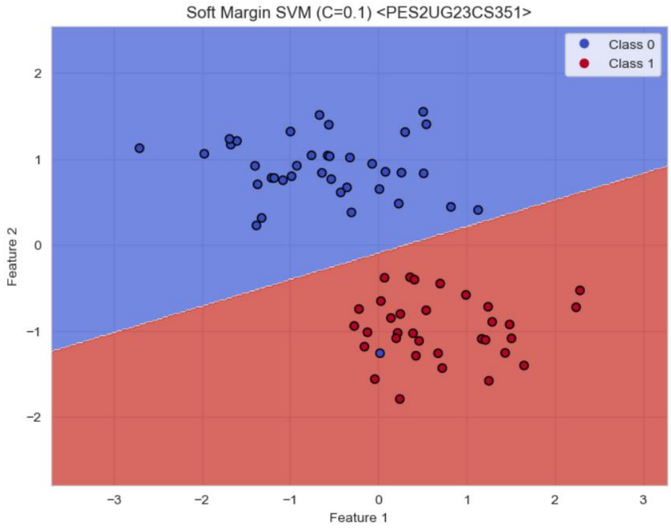
Banknote Dataset:

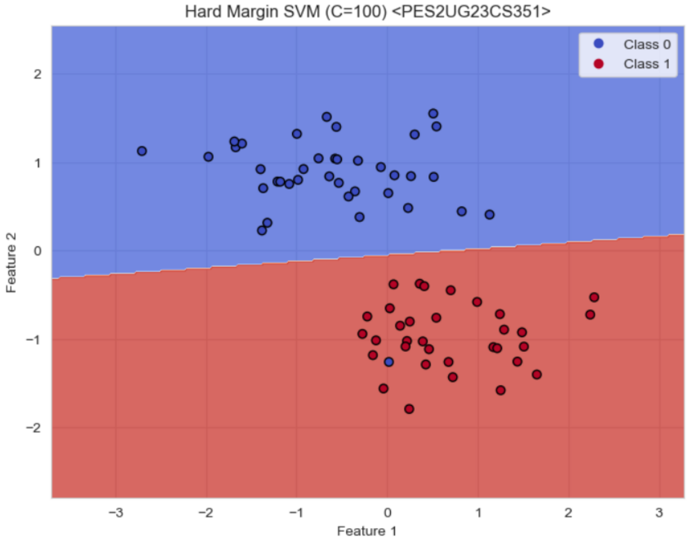






Margin Analysis:


Soft Margin SVM (C=0.1) <PES2UG23CS351>
● Class 0
● Class 1
Feature 2
Feature 1


Hard Margin SVM (C=100) <PES2UG23CS351>
● Class 0
● Class 1
Feature 2
Feature 1