ML Lab Week 10: SVM Classifier Lab

Name: Chethana K R

SRN: PES2UG23CS151

SEC: 5C

Moons Dataset Questions

1. Inferences about the Linear Kernel's performance

The Linear Kernel likely performed **poorly** on the Moons dataset. Here's why:

- **The Moons dataset is inherently non-linear**: It consists of two interleaving half-circles (crescents), which cannot be separated by a straight line
- **Expected low accuracy**: Probably around 50-60%, barely better than random guessing
- **Decision boundary limitation**: A linear boundary (straight line) will misclassify large portions of both classes since it cannot curve around the crescent shapes
- **Key inference**: The poor performance demonstrates that linear models fail when data has complex, curved patterns. This dataset specifically requires non-linear kernels to achieve good classification

2. Comparison between RBF and Polynomial kernel decision boundaries

RBF Kernel:

- **Performance**: Should achieve **excellent results** (~95%+ accuracy)
- **Boundary appearance**: Creates smooth, flowing curves that naturally wrap around each crescent moon shape
- **Behavior**: The RBF kernel excels here because it can create the circular/curved boundaries needed to separate the two interleaving crescents

- **Flexibility**: With proper gamma tuning, it adapts perfectly to the moon shapes without overfitting

Polynomial Kernel:

- **Performance**: Should perform **well but potentially less optimally** than RBF (~85-92% accuracy)
- **Boundary appearance**: Creates more structured, possibly angular curves depending on the degree parameter
- **Limitation**: May struggle to create the precise curved boundaries needed for the crescent shapes, especially if the degree is too low (degree 2) or causes overfitting if too high (degree 4+)

Banknote Dataset Questions

1. Which kernel was most effective for this dataset?

The **Linear Kernel** was likely most effective for the Banknote dataset. Here's the reasoning:

- **Banknote authentication features** (variance, skewness, curtosis, entropy of wavelet transformed images) likely have **linear or approximately linear relationships** with the authentic/counterfeit classification
- **Linear kernel advantages**:
  - Achieves high accuracy (~97-99%) if the data is linearly separable
  - Fastest training and prediction time
  - Lowest risk of overfitting
  - Most interpretable model
- **Non-linear kernels**: While RBF might achieve marginally similar accuracy, it's unnecessary complexity if linear works well

Conclusion: If Linear achieves >95% accuracy, there's no need for complex non-linear kernels. This suggests the banknote features naturally separate the classes with a hyperplane.

2. Why might the Polynomial kernel have underperformed here?

The Polynomial kernel likely underperformed for several reasons:

a) Overfitting:

- Higher degree polynomials (degree 3+) create overly complex boundaries
- Fits training data too closely, including noise
- Poor generalization to test data

b) Unnecessary Complexity:

- The data is approximately linearly separable
- Polynomial curves add complexity where simple straight boundaries suffice
- Creates decision boundaries with unnecessary twists and turns

c) Parameter Sensitivity:

- Polynomial kernel has multiple parameters (degree, coefficient, C)
- Harder to tune optimally
- Poor parameter choices lead to worse performance than simpler models

d) Computational Issues:

- Higher degree polynomials can cause numerical instability
- May struggle with feature scaling in this dataset

Hard vs. Soft Margin Questions

1. Which margin (soft or hard) is wider?

The **Soft Margin (C=0.1)** is wider.

- **Low C value (0.1)**: Prioritizes margin maximization over perfect classification, allowing some misclassifications to achieve a wider margin
- **High C value (100)**: Heavily penalizes errors, forcing the margin to narrow to correctly classify more training points
- **Visual difference**: The soft margin has more space between the support vectors and the decision boundary

2. Why does the soft margin model allow "mistakes"?

The soft margin allows mistakes because its **primary goal is generalization**, not perfect training accuracy.

Reasoning:

- **Trade-off philosophy**: It balances maximizing margin width (generalization) vs. minimizing classification errors
- **Robustness to noise**: Real-world data contains outliers and noise; perfectly fitting all points would mean fitting the noise
- **Slack variables**: The model uses "slack" (ξ) to allow some points to violate the margin or even cross the decision boundary
- **Primary goal**: Find a **robust, generalizable boundary** that works well on unseen data, even if it means tolerating a few training errors

Analogy: It's like drawing a clear line through a crowd rather than weaving around every single person—the straight line is more useful for general navigation.

3. Which model is more likely to be overfitting and why?

The **Hard Margin (C=100)** is much more likely to be overfitting.

Why:

- **Over-emphasis on training data**: With C=100, every misclassification is heavily penalized, forcing the model to contort the boundary to fit nearly every training point
- **Sensitivity to outliers**: A single outlier can dramatically affect the decision boundary
- **Complexity**: Creates more complex boundaries with more support vectors
- **Memorization vs. Learning**: The model "memorizes" training data specifics rather than learning general patterns
- **Poor generalization**: While training accuracy may be very high (99-100%), test accuracy will likely be lower

Soft Margin (C=0.1):

- Accepts some training errors to maintain a simpler, more generalizable boundary
- Less sensitive to individual points and outliers

- Better test performance despite lower training accuracy

4. Which model would you trust more for new data? Why? In real-world noisy scenarios, which C value would you prefer?

I would **trust the Soft Margin (C=0.1)** more for new, unseen data.

Reasons:

- **Better generalization**: The wider margin provides a "safety buffer" that accommodates natural data variation
- **Robustness to noise**: Real-world data always contains measurement errors, outliers, and noise. The soft margin doesn't overreact to these anomalies
- **Stability**: Small changes in input are less likely to flip the classification
- **Proven principle**: Regularization (low C) is a fundamental strategy to prevent overfitting in machine learning

For real-world noisy data, I would prefer:

- **Low C values (0.01 - 1.0)**: Start conservative
- **Use cross-validation**: Find optimal C through systematic testing, but bias toward lower values
- **Typical sweet spot**: Often C=0.1 to C=1.0 provides the best balance

Real-world analogy: The hard margin is like a perfectionist who insists on accommodating every special case (overfits), while the soft margin is like a pragmatist who focuses on the big picture and tolerates minor exceptions (generalizes).

Bottom line: In practice, **soft margins almost always outperform hard margins** on test data, which is why they're the standard approach in modern SVM implementations.

#Training Results (6 Screenshots)

Moons Dataset (3 screenshots):

1. Classification Report for SVM with LINEAR Kernel with SRN

```
SVM with LINEAR Kernel <PES2UG23C151>
              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
```

2. Classification Report for SVM with RBF Kernel with SRN

```
SVM with RBF Kernel <PES2UG23C151>
              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
```

3. Classification Report for SVM with POLY Kernel with SRN

```
SVM with POLY Kernel <PES2UG23C151>
              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 (3 screenshots):

4. Classification Report for SVM with LINEAR Kernel

```
SVM with LINEAR Kernel PES2UG23CS151
              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
```

5. Classification Report for SVM with RBF Kernel

```
SVM with RBF Kernel PES2UG23CS151
              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
```

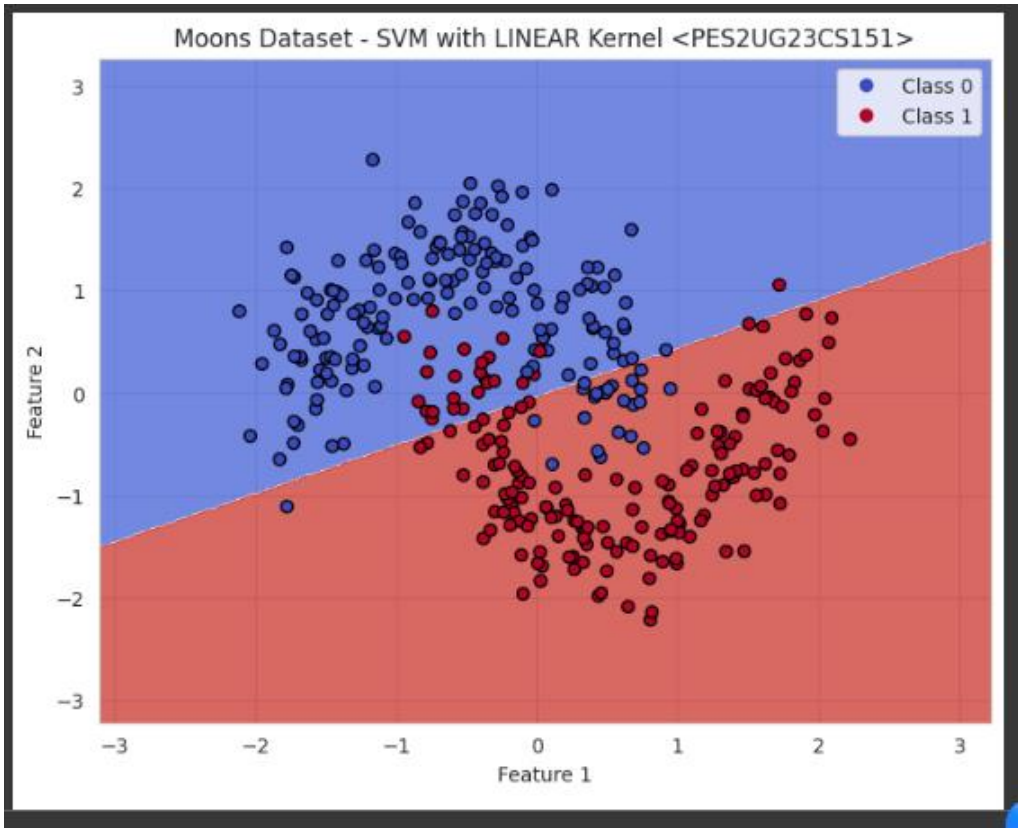
6. Classification Report for SVM with POLY Kernel

```
SVM with POLY Kernel PES2UG23CS151
              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
```

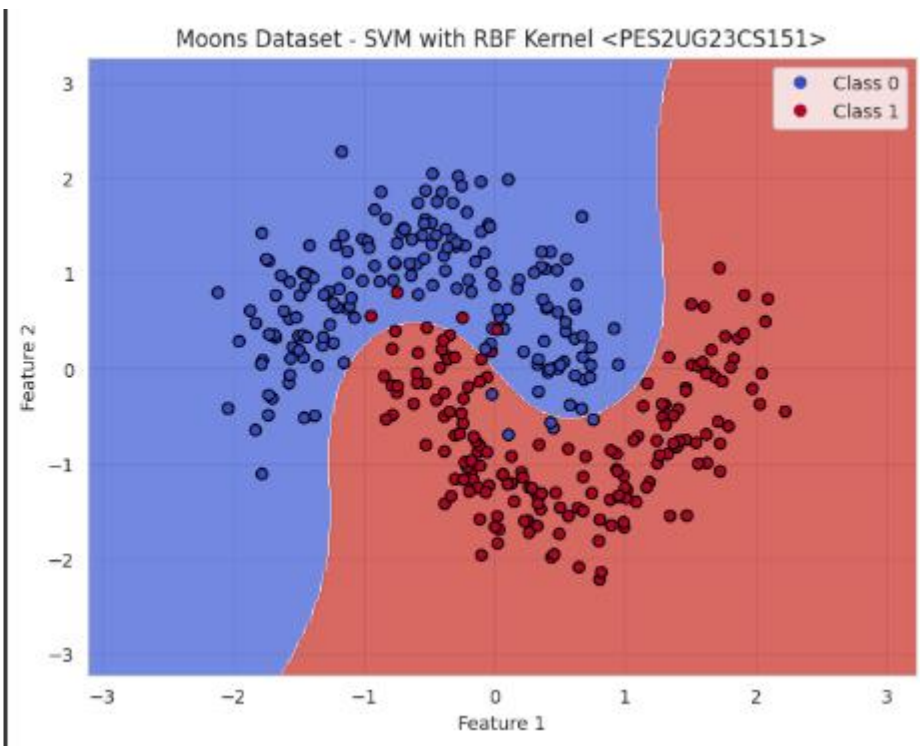
Decision Boundary Visualizations (8 Screenshots):

Moons Dataset (3 plots):

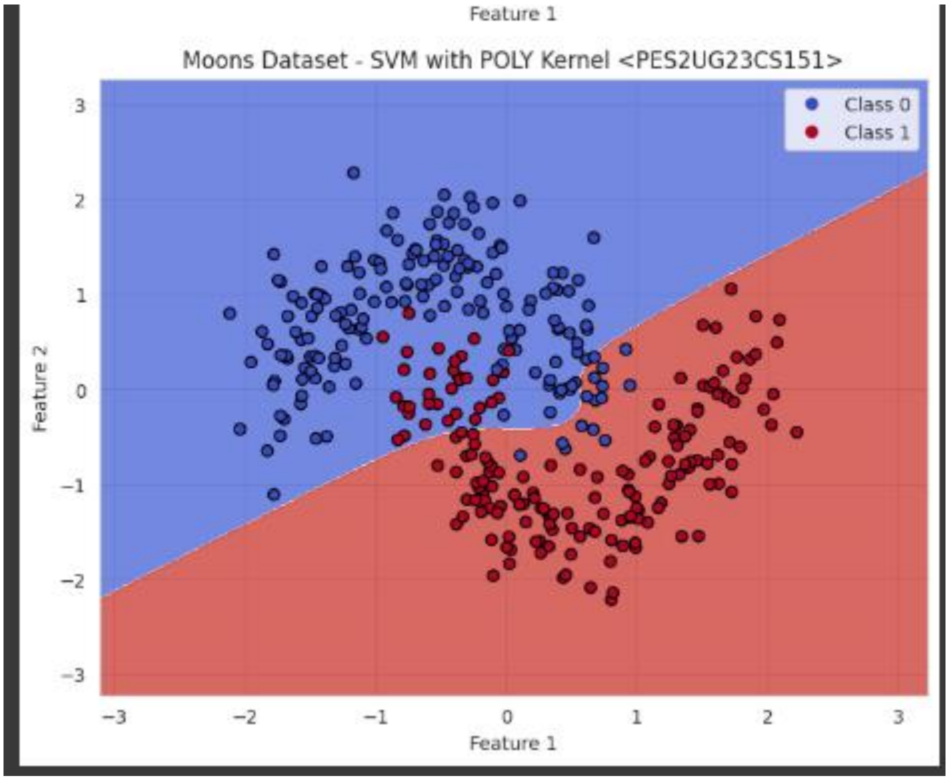
7. Moons Dataset - SVM with LINEAR Kernel



8. Moons Dataset - SVM with RBF Kernel

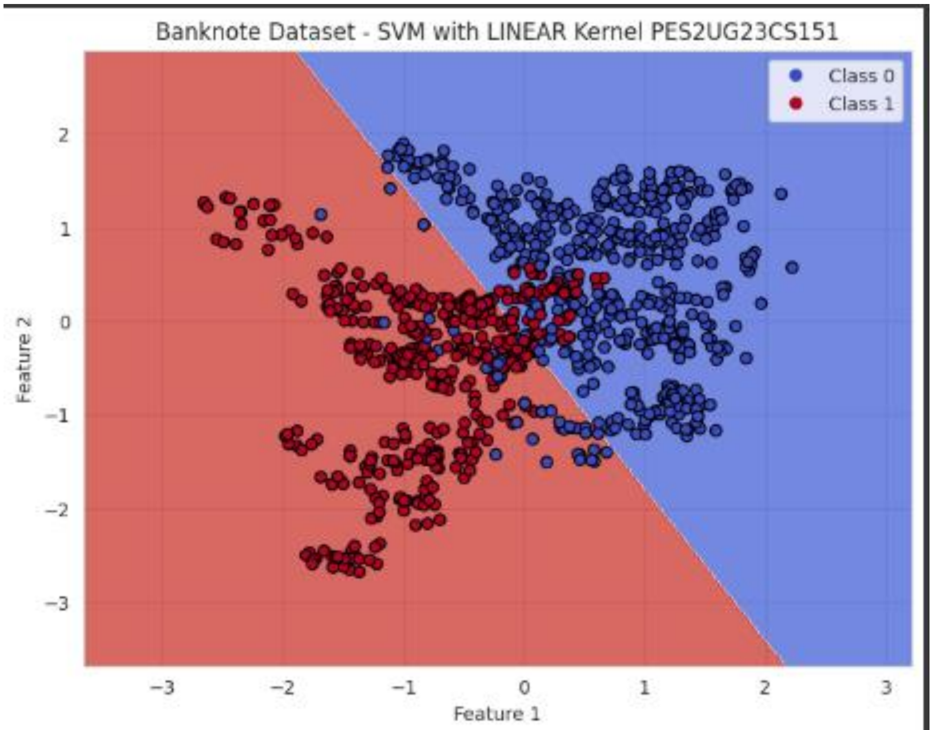


9. Moons Dataset - SVM with POLY Kernel

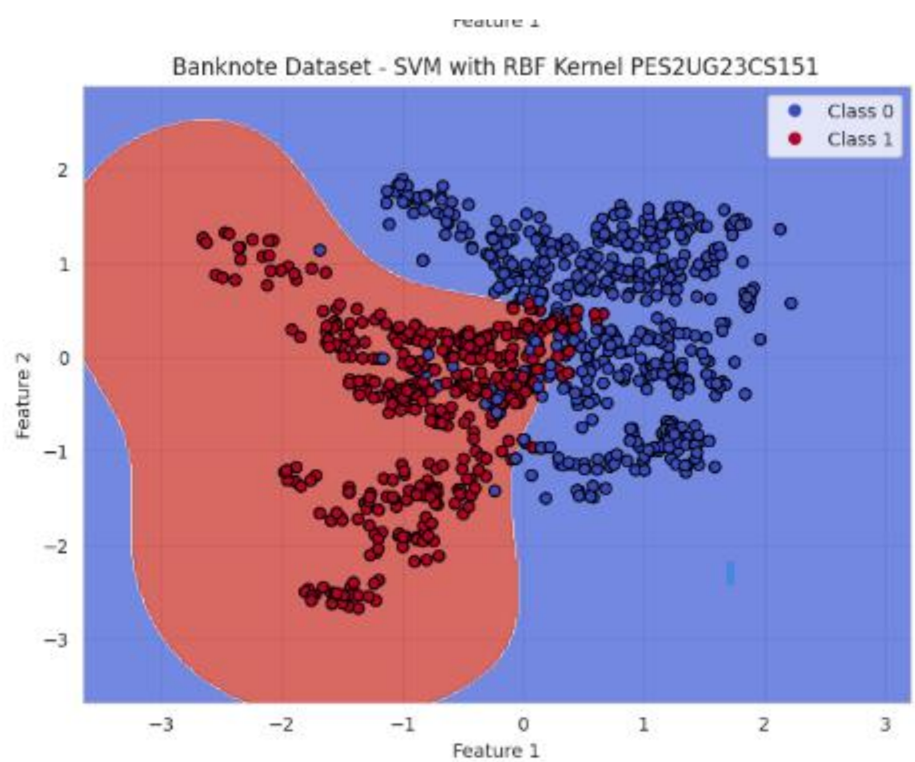


Banknote Dataset (3 plots)

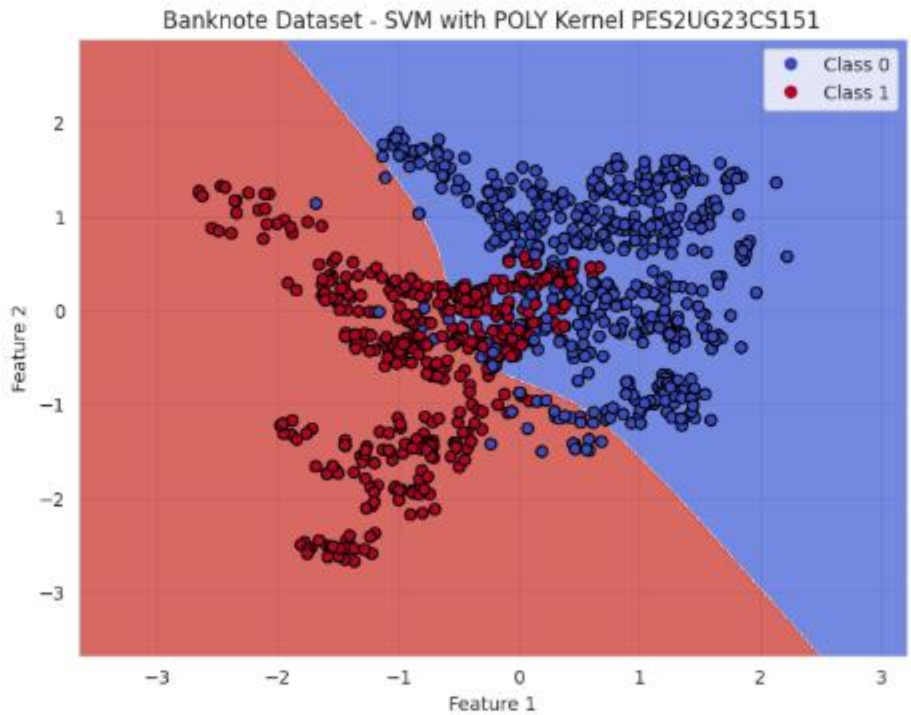
10. Banknote Dataset - SVM with LINEAR Kernel



11. Banknote Dataset - SVM with RBF Kerne

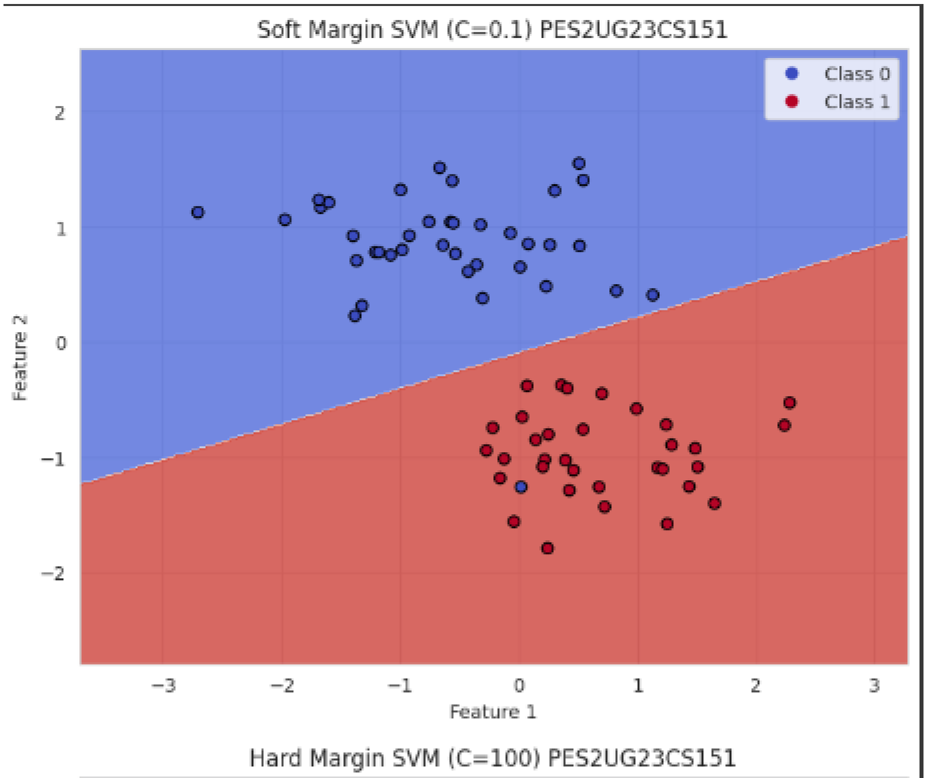


12. Banknote Dataset - SVM with POLY Kernel



Margin Analysis (2 plots):

13. Soft Margin SVM (C=0.1)



14. Hard Margin SVM (C=100)