UE23CS352A: MACHINE LEARNING
WEEK-SVM

Name : C Yogesh Reddy

SRN : PES2UG23CS159

Sec:C

<u>MOONS DATASET</u>

Classification reports

1) LINEAR SVM

```
SVM with LINEAR Kernel PES2UG23CS159
              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) RBF SVM

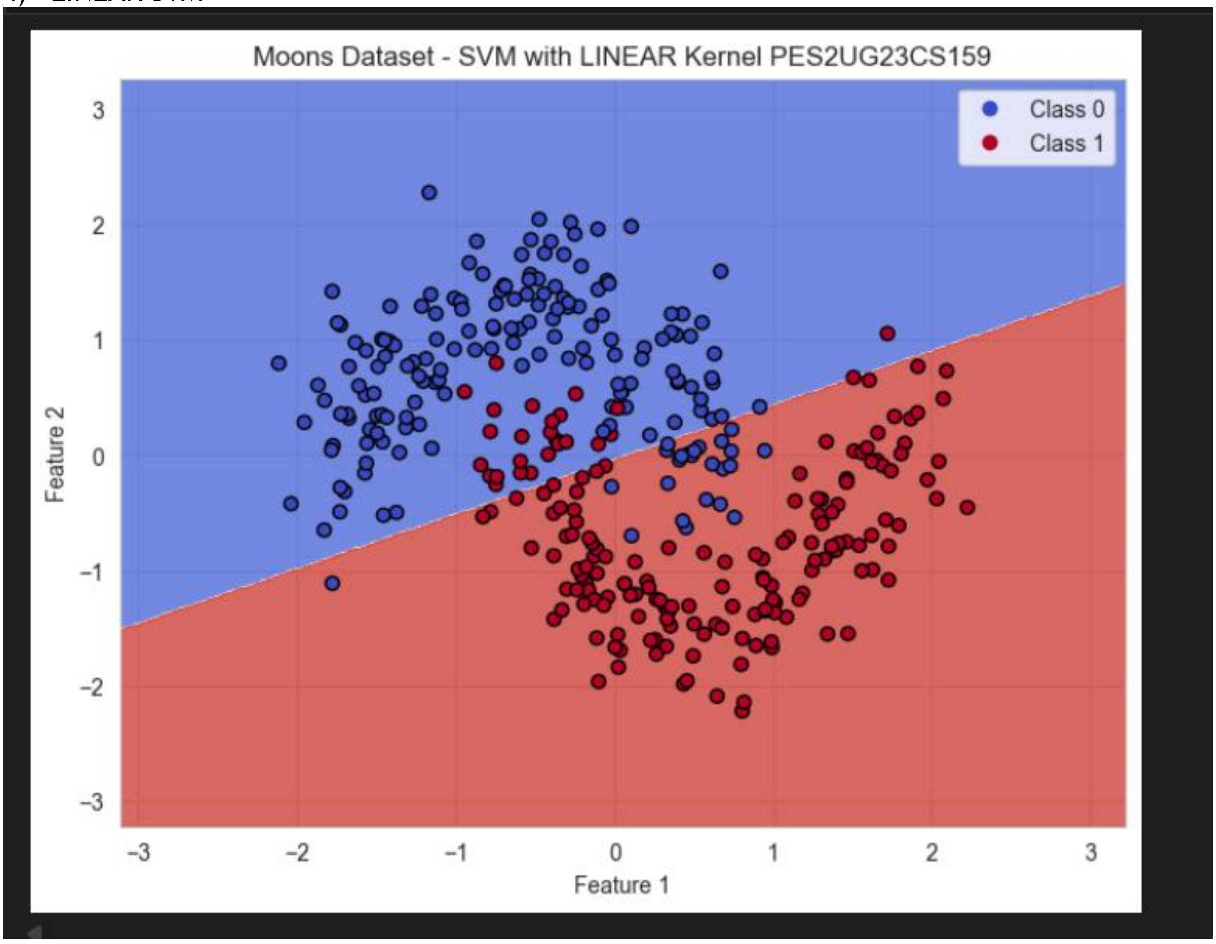
```
SVM with RBF Kernel PES2UG23CS159
              precision    recall  f1-score   support

           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) POLY SVM

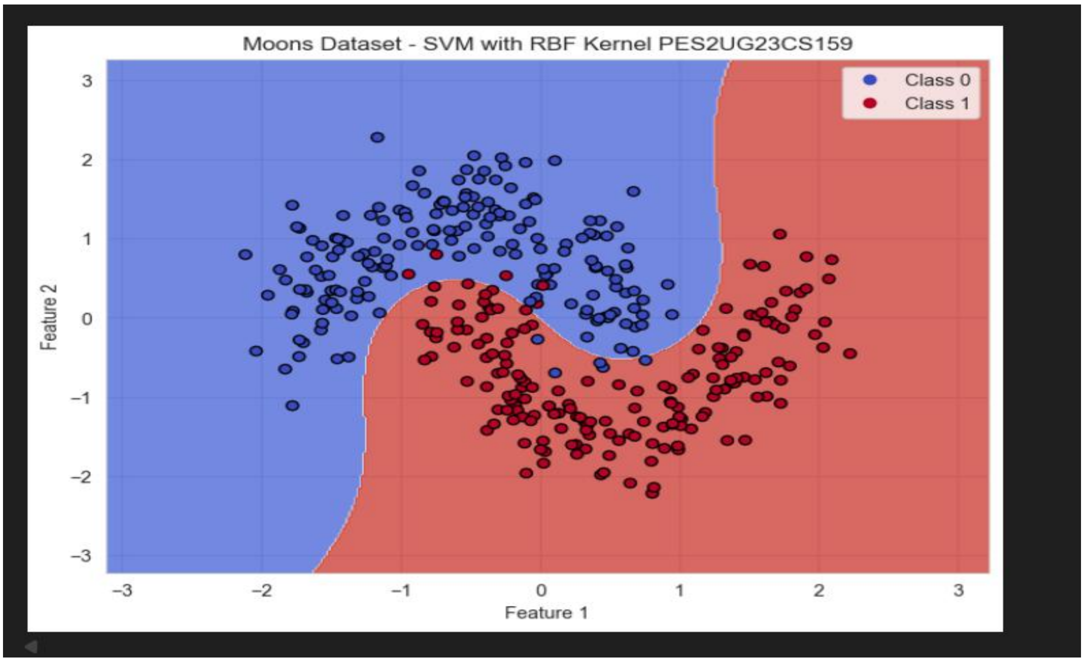
```
SVM with POLY Kernel PES2UG23CS159
              precision    recall  f1-score   support

           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
```

Decision Boundaries

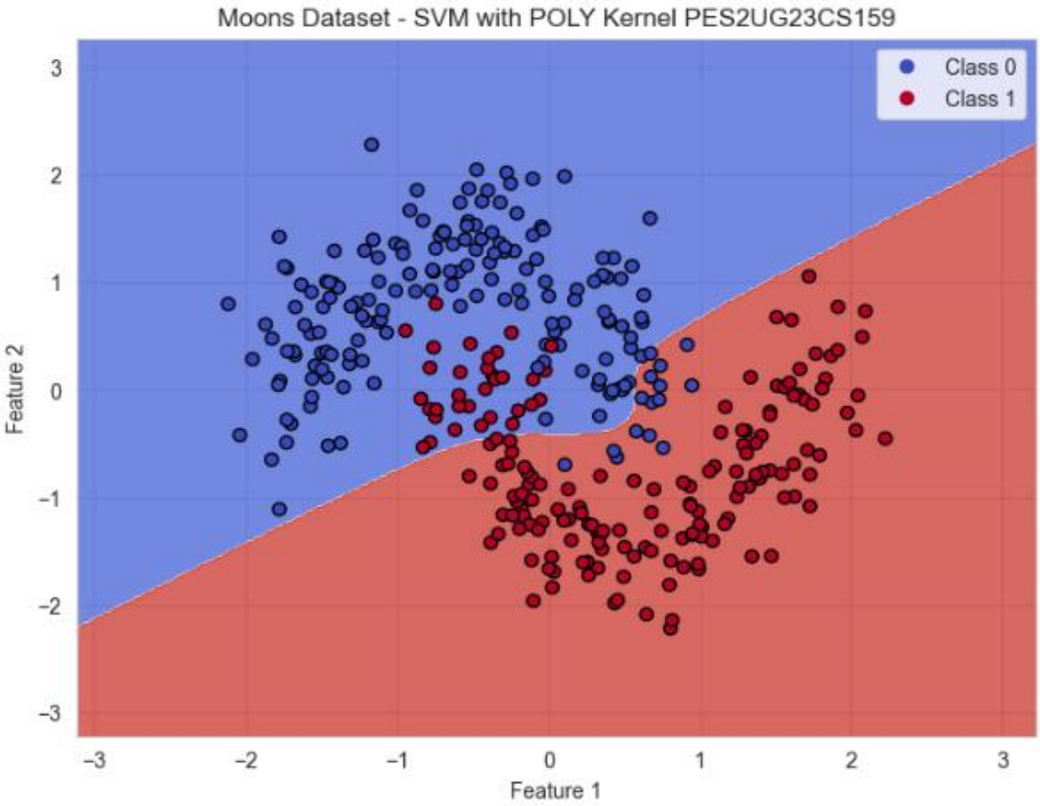
4) LINEAR SVM



5)RBF SVM



6)POLY SVM



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

Ans.

The Linear Kernel performed well on the Banknote dataset since the data is almost linearly separable, leading to high accuracy and a clear decision boundary. However, it underperformed on the Moons dataset, where the classes are not linearly separable. The straight decision line fails to capture the complex curve of the data, resulting in lower classification accuracy

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

Ans.
The RBF kernel forms smooth and flexible decision boundaries that closely follow the true shape of the data.
The Polynomial kernel creates more rigid and sometimes unnecessarily complex curves, which may lead to overfitting, especially in noisy datasets like Moons.

Banknote Dataset:

Classification Report for SVM with LINEAR Kernel:

SVM with LINEAR Kernel PES2UG23CS159

	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

Classification Report for SVM with RBF Kernel:

SVM with RBF Kernel PES2UG23CS159

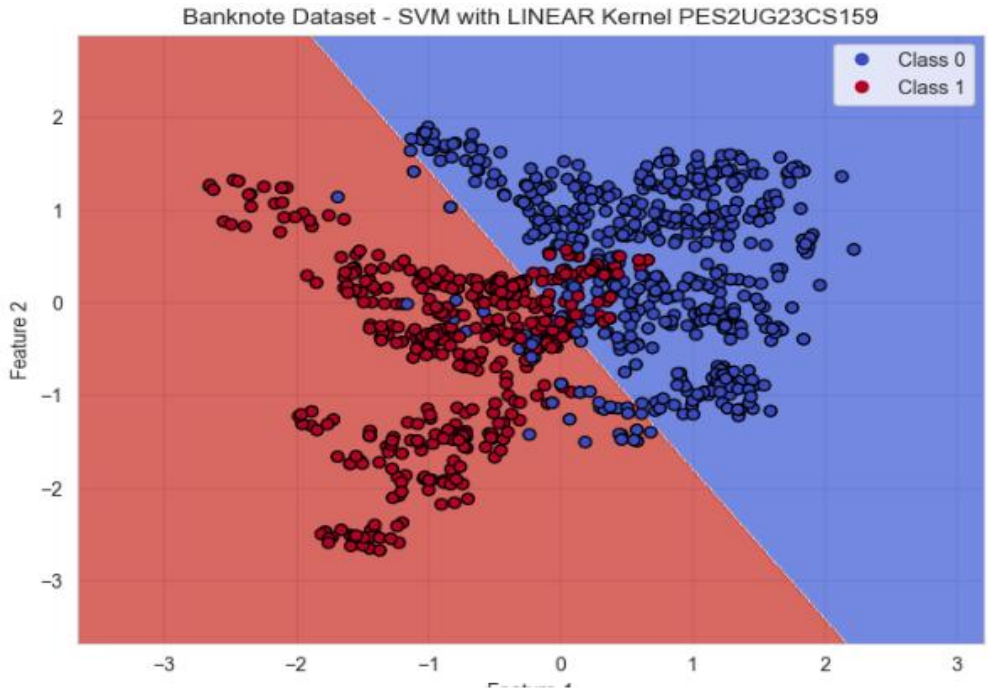
	precision	recall	f1-score	support
Forged	0.96	0.91	0.94	229
Genuine	0.90	0.96	0.93	183
accuracy			0.93	412
macro avg	0.93	0.93	0.93	412
weighted avg	0.93	0.93	0.93	412

Classification Report for SVM with POLY Kernel:

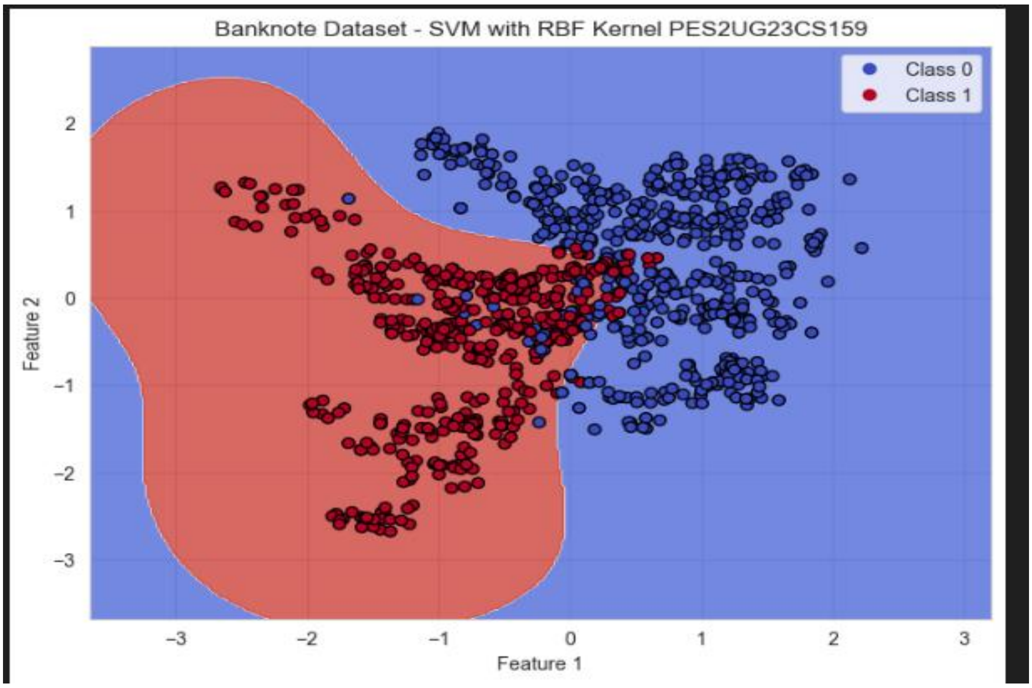
SVM with POLY Kernel PES2UG23CS159

	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

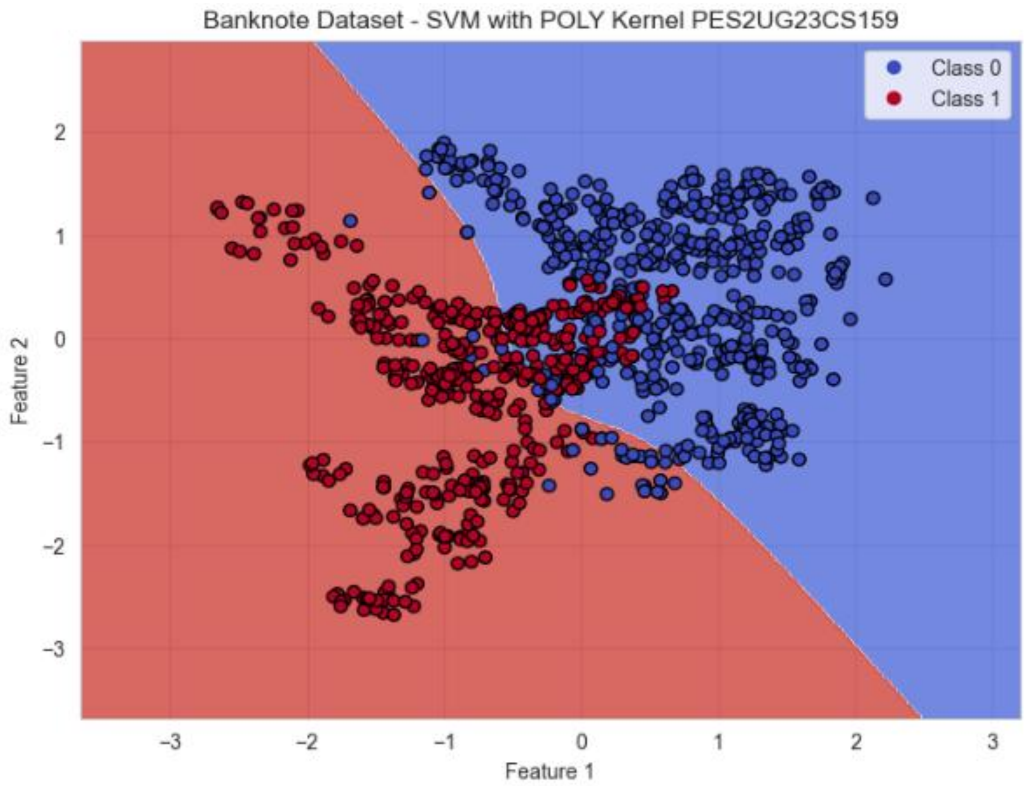
SVM with LINEAR Kernel:



SVM Kernel with RBF:



SVM Kernel with POLY:



1) In this case, which kernel appears to be the most effective?

Ans.
The Linear kernel performed best on the Banknote dataset.
This is because the dataset is almost linearly separable, so a simple linear hyperplane is enough to achieve high classification accuracy without adding unnecessary complexity.

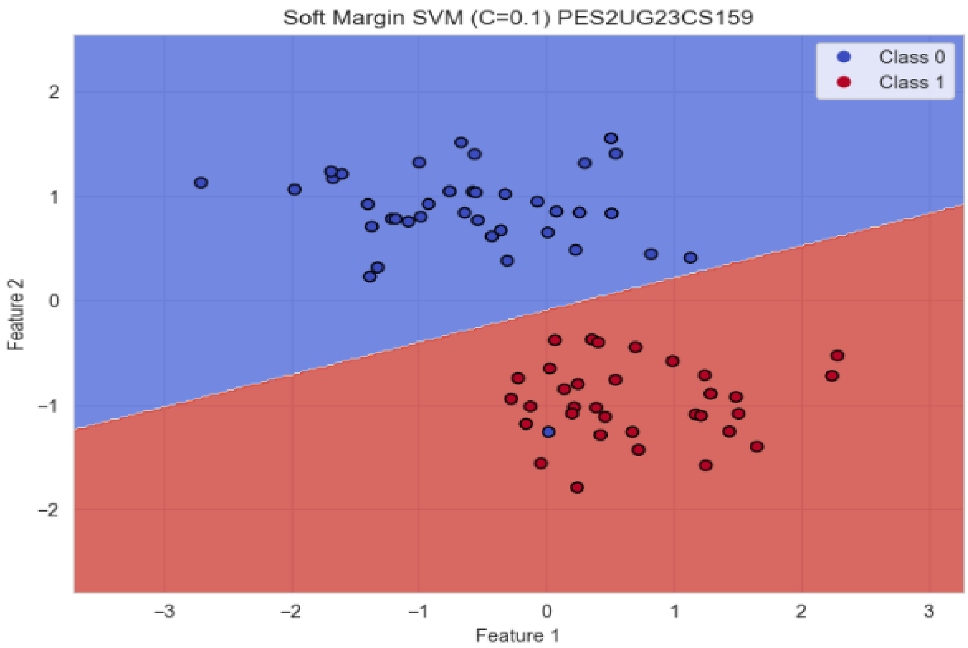
2) The Polynomial kernel shows lower performance here compared to the Moons dataset. What might be the reason for this?

Ans.

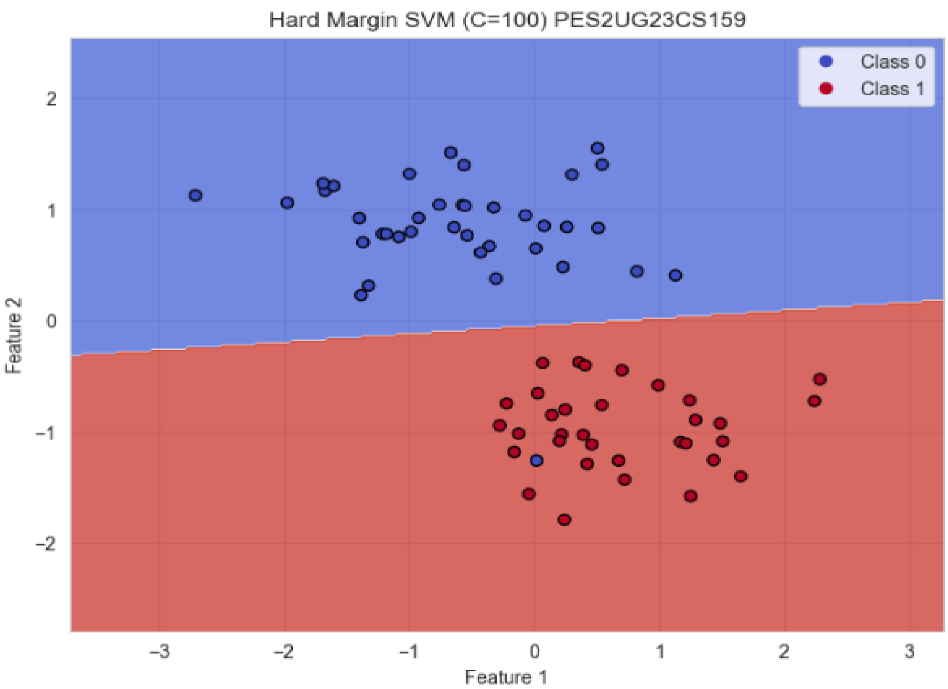
The Polynomial kernel introduces additional non-linear transformations that are not required for this dataset.
Since the classes are already well separated, these extra transformations only add computational complexity and increase the chance of overfitting, which results in slightly lower accuracy than the Linear or RBF kernels

Hard vs. Soft Margin Dataset:

Soft Margin SVM (C=0.1):



Soft Margin SVM (C=0.1):



1) **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.
A lower value of C allows the SVM to focus on maximizing the margin, even if a few data points are misclassified. This leads to a smoother and more generalizable boundary.

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?

   Ans.
   The Soft Margin model allows some misclassifications because with a low C value, the penalty for these errors is smaller.
   This helps the SVM maximize the margin and avoid overfitting. The primary goal is better generalization rather than perfectly classifying every training point.

3) 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.
   It tries to classify all training points perfectly, forcing a very tight decision boundary. This makes it less flexible and more sensitive to noise in the data.

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?

   Ans.
   The Soft Margin model (C = 0.1) is more trustworthy for new data.
   Because it allows flexibility, it can generalize better to unseen data rather than memorizing the training set.
   In real-world noisy datasets, it's usually better to start with a low C value, then adjust if needed.