SVM Model Analysis Report: Moons Dataset

Based on the experiment conducted in the code notebook, here is the analysis report.

Model Comparison

This table shows the validation accuracy for the four models tested on the 30% hold-out validation set.

Model	Parameters	Validation Accuracy
1. Linear SVM	kernel='linear', C=1.0	85.33%
2. RBF SVM	kernel='rbf' (default params)	94.67%
3. Polynomial SVM	kernel='poly', degree=3	87.33%
4. Tuned RBF SVM	C=10, gamma=1 (from GridSearchCV)	**97.00%**

The tuned RBF SVM (Model 4) achieved the highest accuracy, demonstrating the importance of both kernel selection and hyperparameter tuning.

Final Model Performance (Tuned RBF SVM)

The final model, selected by GridSearchCV (C=10, gamma=1), was evaluated on the 30% hold-out validation set.

Classification Report

```
              precision    recall  f1-score   support

           0       0.94      1.00      0.97        75
           1       1.00      0.93      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
```

Confusion Matrix

The confusion matrix shows the model's performance in detail.

```
[[75  0]
 [ 5 70]]
```

- **True Negatives (Top-Left):** 75 - All 75 instances of class 0 were correctly classified.
- **False Positives (Top-Right):** 0 - No instances of class 1 were misclassified as class 0.
- **False Negatives (Bottom-Left):** 5 - 5 instances of class 0 were misclassified as class 1.
- **True Positives (Bottom-Right):** 70 - 70 of 75 instances of class 1 were correctly classified.

The model is exceptionally good at identifying class 0 but makes a few errors in classifying class 1.

Decision Boundary Plots

The 1x3 subplot visualization (from Task 5 in the notebook) clearly shows:

1. **Linear SVM:** Fails to separate the non-linear "moons" with a straight line, resulting in significant misclassification.
2. **Default RBF SVM:** Creates a much better, non-linear boundary that effectively captures the general shape of the data.
3. **Tuned RBF SVM:** Provides the cleanest boundary, fitting the data tightly and accurately (C=10, gamma=1), which corresponds to its superior performance in the classification report.

Analysis & Hyperparameter Effects

● Why did the Linear SVM fail, and why did the RBF kernel succeed?

The linear SVM failed because the moons dataset is **non-linearly separable**. The data forms two moon-shaped curves, and there is no single straight line (linear hyperplane) that can separate the two classes correctly. As a result, the linear model misclassifies many points, leading to comparatively low accuracy and a messy boundary.

In contrast, the **RBF kernel succeeded** because:

- It uses the "kernel trick" to map the data into a **higher-dimensional space**.
- In that transformed space, the moons become linearly separable.
- The RBF kernel draws a smooth, **curved boundary** that wraps around the moon shape.

- This dramatically improves classification accuracy, as seen in the model comparison.

● What did GridSearchCV tell you? What were the best C and gamma?

GridSearchCV performs a systematic search over combinations of hyperparameters and selects the model that gives the highest cross-validated accuracy.

- **Best C = 10**: This value means the model allows fewer misclassifications (a higher penalty), resulting in a clearer and stricter boundary.
- **Best Gamma = 1**: This parameter influences how far the effect of a single data point spreads.
  - If gamma is too small, the boundary is too smooth and underfits.
  - If gamma is too large, the boundary is too sharp and overfits.

GridSearchCV finds the "sweet spot": a boundary that is flexible enough to capture the data's patterns but not so complex that it overfits.

● What happens if gamma is too high (e.g., 1000)?

If gamma is extremely high:

- The influence of each training point becomes very small and localized.
- The model tries to fit every single training point exactly.
- The decision boundary becomes very "wiggly" and extremely complex. It overfits noise and looks like it draws tight loops around individual points.

Effect:

- Training accuracy becomes very high.
- Validation accuracy decreases.
- The decision boundary looks irregular and overfitted.

● What happens if C is too low (e.g., 0.01)?

C controls the penalty for misclassification.

- When C is small (a low penalty), the model allows many misclassifications to keep the boundary as simple as possible.
- The decision boundary becomes too smooth and broad, failing to fit the moon shapes properly. This results in **underfitting**.

Effect:

- Many errors are allowed.
- The boundary is too simple.
- Accuracy drops.
- The model becomes very generalized but is not accurate.