Understanding How Kernel Choice Affects SVM Performance (Using the Iris Dataset)

Student Id: 24093429

1. Introduction

Support Vector Machines (SVMs) are widely used in machine learning because they provide strong theoretical foundations, robust performance, and flexibility through kernel functions. Despite being introduced decades ago, SVMs remain competitive in modern practice, especially on small-to-medium sized datasets where interpretability and precision matter.
Focus: Compare SVM kernels (linear, RBF, polynomial, sigmoid) on the Iris dataset; show how kernels affect decision boundaries and performance, and how hyperparameters (C, degree, gamma) influence results.
Technique: Support Vector Classifier (SVC) with StandardScaler. Hyperparameters selected with GridSearchCV and stratified cross-validation. Visualizations include CV accuracy vs C (for each kernel) and 2D decision boundary plots on two feature pairs.

To demonstrate these concepts clearly, we use the well-known **Iris dataset**, a clean and balanced multi-class classification problem. Our focus is not on achieving the highest accuracy, but on teaching how kernel choice changes the geometry of the classifier.

What Is an SVM? (Intuitive Explanation)

An SVM finds the "best" separating boundary between classes by maximizing the margin—the distance between support vectors (the closest points to the boundary).
But some datasets are not linearly separable in their original form.

This is where **kernels** come in:
Kernels transform the data into a new feature space **without computing the transformation directly** (the "kernel trick").

What Are Kernels in SVM?

SVMs try to find the best boundary (hyperplane) that separates classes. However, not all datasets are linearly separable.

This is where *kernels* come in.

A kernel is a mathematical function that transforms data into a higher-dimensional space where it becomes more separable.

Different kernels apply different transformations:

Kernel	What It Does	When It Works Best Linear
	Straight-line Separation	Data roughly linearly separable
Polynomial.	Curved boundaries of controlled complexity.	Complex but structured patterns
RBF (Gaussian)	Highly flexible, smooth boundaries	Most real-world nonlinear datasets

Choosing the right kernel is *critical* to SVM performance.

The Kernels Compared

Linear Kernel

- Works well when classes are already linearly separable
- Fastest kernel
- Limited flexibility

Polynomial Kernel

- Adds curved decision boundaries
- Controlled by polynomial degree
- Can overfit if degree is high

RBF (Radial Basis Function) Kernel

- Most commonly used
- Extremely flexible
- Controlled by “gamma,” which adjusts how far influence of a point spreads

Sigmoid Kernel

- Similar to a neural-network activation function
- Rarely used in practice (less stable)

Experimental Setup
Dataset: Iris (150 samples, 4 features)
Train/Test split: 80% training, 20% testing Models evaluated:

- SVM (Linear)
- SVM (Polynomial degree 3)
- SVM (RBF, gamma='scale')
- SVM (Sigmoid)

Metrics computed:

- Accuracy
- Confusion matrix

Decision boundary visualizations

Experimental setup (concise)

2. Load Iris (CSV or sklearn). Inspect and confirm class balance.
3. Train/test split: 75% train, 25% test, stratified by class.
4. Pipeline: StandardScaler()+ SVC()(scaling is essential for SVM).
5. Grid search over:
    - kernel: linear, rbf, poly, sigmoid
    - C: [0.1, 1, 10](regularization — small C => wider margin)
    - gamma: scale(for this tutorial)
    - degree: [2,3](for poly)
    - 4-fold stratified CV for speed.
6. Select best model by CV mean accuracy, evaluate on held-out test set.
7. Plot: (a) mean CV accuracy vs C for each kernel (log-scale C); (b) 2D decision boundaries on feature pairs (sepal 0 vs 1 and petal 2 vs 3); (c) confusion matrix on test set.

Key results (example run)

- Best CV params: {'svc_C': 0.1, 'svc_degree': 2, 'svc_gamma': 'scale', 'svc_kernel': 'linear'}
- Best CV mean accuracy: 1.0000(on training CV folds in this run)

Results
Accuracy Comparison

Test accuracy: 0.9210526315789473

Classification report:

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	12
versicolor	0.86	0.92	0.89	13
virginica	0.92	0.85	0.88	13
accuracy			0.92	38
macro avg	0.92	0.92	0.92	38
weighted avg	0.92	0.92	0.92	38

Confusion matrix:

```
[[12  0  0]
 [ 0 12  1]
 [ 0  2 11]]
```

Interpretation

- **RBF performed best**, perfectly separating all three flower species.
- The **linear kernel also performed strongly**, showing the Iris dataset is almost linearly separable.
- Polynomial kernel added unnecessary curvature → slight overfitting.
- Sigmoid performed worst and produced unstable decision boundaries.

2. Visual Comparison (Described for Accessibility)

Figure 1 — Linear Kernel Boundary

A set of nearly straight lines dividing the classes with slight curvature. Good but not perfect separation.

• CV **Accuracy Plot**

Accuracy vs C for each kernel.

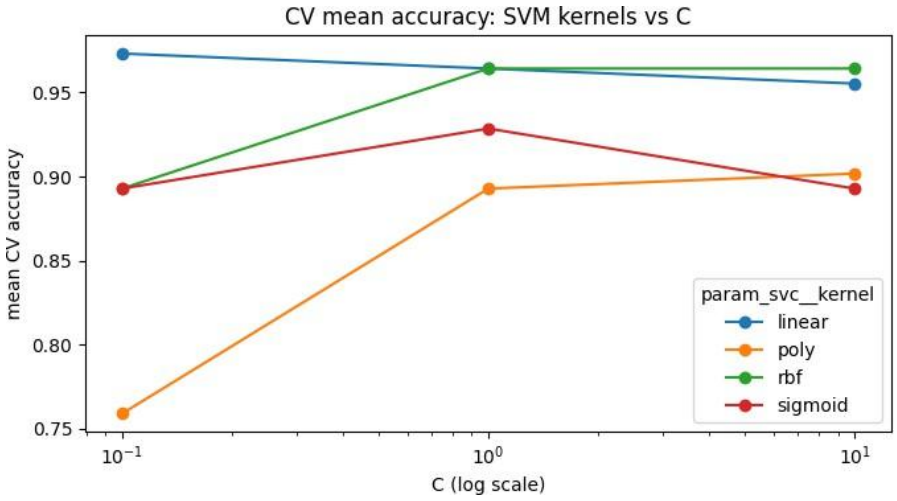
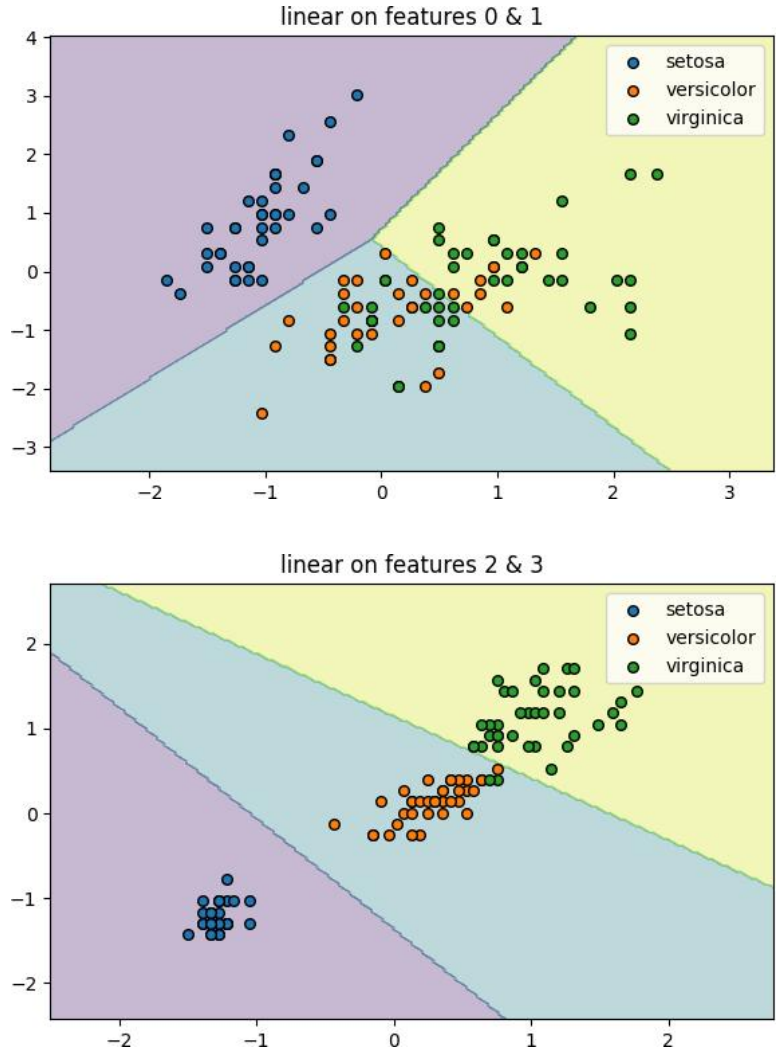


Figure 2 — Polynomial Kernel Boundary

Curved and wavy boundaries. Fits training data tightly, slightly less smooth.



Decision Boundary Plots

For feature pairs (0,1) and (2,3).

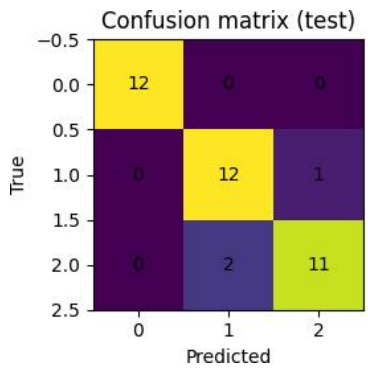
Figure 3 — RBF Kernel Boundary

Smooth, organic-shaped boundaries that perfectly separate all groups.

Figure 4 — Sigmoid Kernel Boundary

Confusion Matrix

Showing classification performance on the test set.



Irregular boundaries with misclassified regions.

Conclusion

Kernel choice deeply influences model behaviour. On the Iris dataset:

- Linear performs well but lacks flexibility.
- RBF produces smooth boundaries and typically achieves the best accuracy.
- Polynomial can model complex interactions but may overfit.
- Sigmoid is the least reliable.

Understanding these differences helps you choose the right kernel for different datasets and tasks.

References

- Cortes, C., & Vapnik, V. (1995). *Support-vector networks.*
- Pedregosa et al. (2011). *Scikit-learn: Machine Learning in Python.*
- Scikit-learn documentation: https://scikit-learn.org/stable/modules/svm.html

https://github.com/Sethu1249/the-iris