How Different Kernels Change the behavior of SVM.

1. Introduction

1.1 Overview

Support Vector Machines (SVMs) are one of the most powerful and widely used supervised machine learning algorithms. They are particularly effective for classification tasks, but can also be used for regression. One of the key features that make SVMs so versatile is the use of kernels. Kernels allow SVMs to operate in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space. This is particularly useful when the data is not linearly separable in its original feature space.

In this tutorial, we will explore how different kernels change the behavior of SVMs. We will start by understanding what SVMs and kernels are, then delve into the different types of kernels. Finally, we will apply SVMs with different kernels to the Breast Cancer dataset to detect malignant and benign tumors. We will analyze the performance of each kernel using accuracy scores, ROC curves, and other plots, and explain why each kernel performed the way it did.

1.2 What is a Support Vector Machine (SVM)?

A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression tasks. However, it is most commonly used for classification. The goal of an SVM is to find the optimal hyperplane that separates the data points of different classes in the feature space. The optimal hyperplane is the one that maximizes the margin, which is the distance between the hyperplane and the nearest data points from either class. These nearest data points are called support vectors.

In cases where the data is not linearly separable, SVMs use a technique called the kernel trick to transform the data into a higher-dimensional space where a linear separation is possible.

1.3 What is a Kernel?

A kernel is a function that takes two inputs (data points) and returns a value that represents how similar those two points are. In the context of SVMs, kernels are used to compute the dot product of two vectors in a high-dimensional feature space without explicitly mapping the data to that space. This is known as the kernel trick.

Mathematically, a kernel function K(x,y) can be represented as:

$K(x,y)=\phi(x) \cdot \phi(y)$

Where ϕ is a mapping from the original feature space to the higher-dimensional space.

Kernels are crucial because they allow SVMs to perform complex, non-linear classifications by implicitly mapping the input data into a higher-dimensional space where a linear separator can be found.

1.4 Types of Kernels:

There are several types of kernels that can be used in SVMs, each with its own characteristics and use cases. The most commonly used kernels are:

Linear Kernel

Formula: $K(x,y)=x \cdot y$

The linear kernel is the simplest kernel. It is used when the data is linearly separable. It does not transform the data into a higher-dimensional space, so it is computationally efficient.

Polynomial Kernel

Formula: $K(x,y)=(x \cdot y+c)^d$

The polynomial kernel can map the data into a higher-dimensional space, allowing for more complex decision boundaries. The degree d controls the flexibility of the kernel, and c is a constant term.

Radial Basis Function (RBF) Kernel:

Formula: $K(x,y)=\exp(-\gamma \| x-y \|^2)$

The RBF kernel is one of the most popular kernels. It maps the data into an infinite-dimensional space, allowing for very complex decision boundaries. The parameter γ controls the influence of each training example.

Sigmoid Kernel

Formula: $K(x,y)=\tanh(\alpha x \cdot y+c)$

The sigmoid kernel is similar to the activation function used in neural networks. It can be used for non-linear classification, but it is less commonly used compared to the RBF and polynomial kernels.

1.5 Dataset

The dataset used is the Breast Cancer Wisconsin (Diagnostic) Dataset. It contains 30 numeric features computed from digitized images of fine needle aspirates (FNA) of breast masses. The target variable is the diagnosis, where:
Malignant = 1
Benign = 0

2. Methodology

2.1 Dataset

The dataset used is the Breast Cancer Wisconsin (Diagnostic) Dataset. It contains 30 numeric features computed from digitized images of fine needle aspirates (FNA) of breast masses. The target variable is the diagnosis, where:
Malignant = 1
Benign = 0

2.2 Data Preprocessing

1. Load the dataset and drop irrelevant columns (e.g., id).
2. Encode the target variable (diagnosis) into binary values (Malignant = 1, Benign = 0).
3. Split the dataset into training (80%) and testing (20%) sets.
4. Scale the features using StandardScaler to ensure all features have the same scale.

2.3 Dimensionality Reduction

PCA (Principal Component Analysis) was used to reduce the dataset to 2 dimensions for visualization purposes. This allows us to plot decision boundaries in 2D space.

2.4 SVM Implementation

Kernels Used:
- Linear Kernel: Suitable for linearly separable data.
- RBF Kernel: Uses a radial basis function to handle non-linear data.
- Polynomial Kernel: Uses a polynomial function to transform the data.
- Sigmoid Kernel: Uses a sigmoid function, similar to neural networks.

2.5 Evaluation Metrics

The following metrics were used to evaluate model performance:
- Accuracy

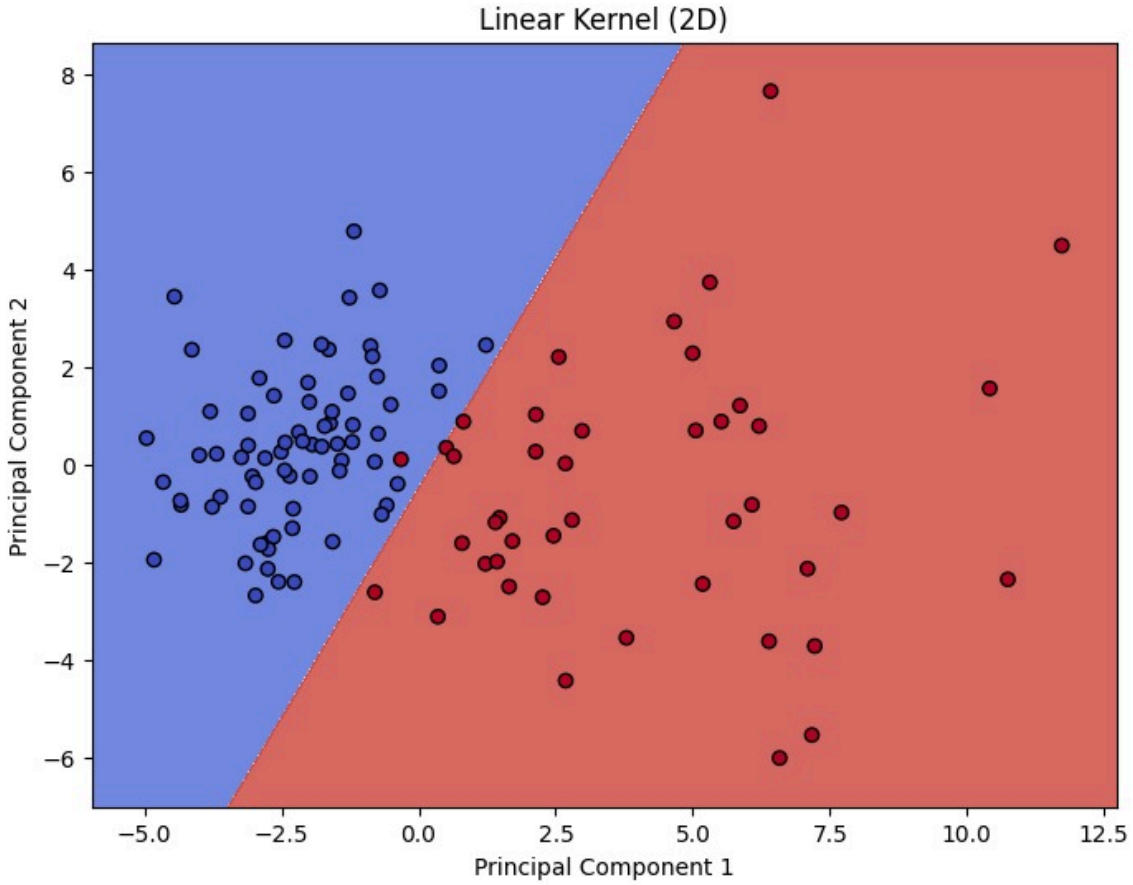
- Confusion Matrix
- Classification Report
- ROC Curve

3. Results and Analysis

3.1 Performance Metrics

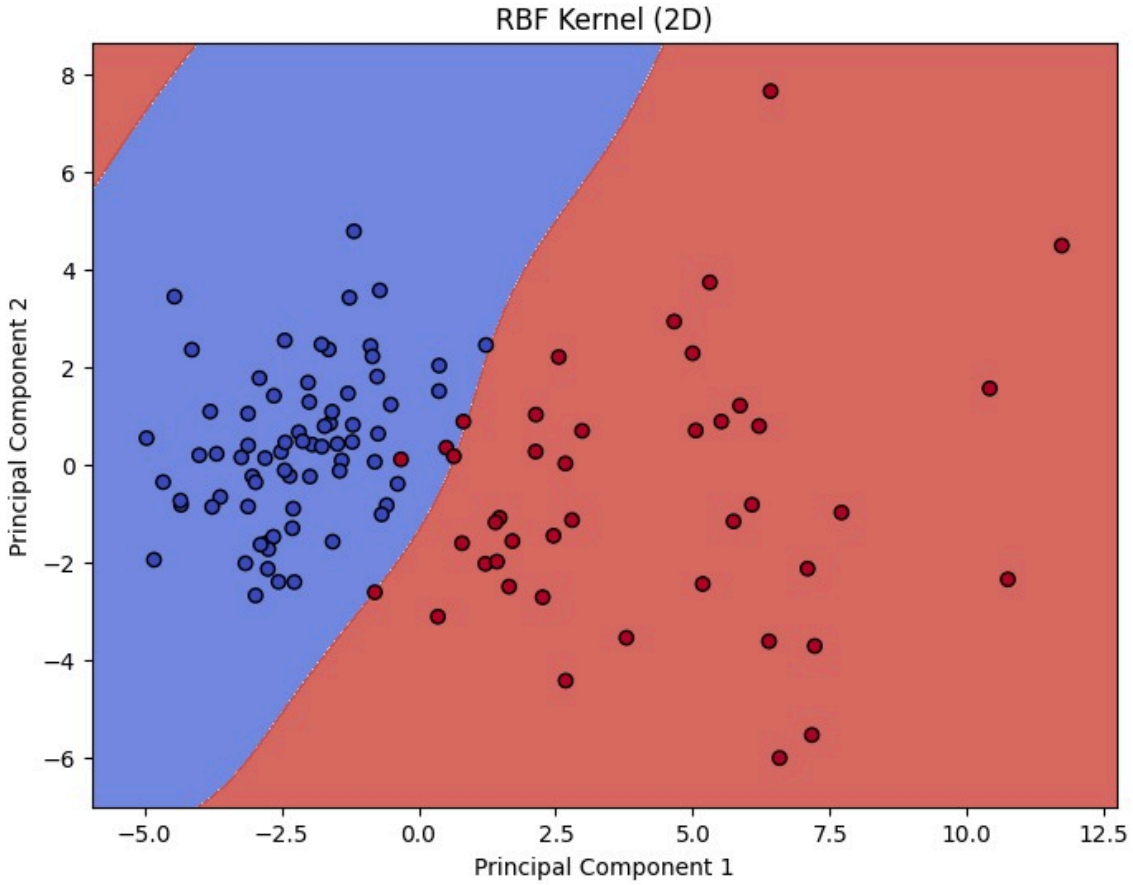
The classification performance for different kernels is as follows:

Linear Kernel



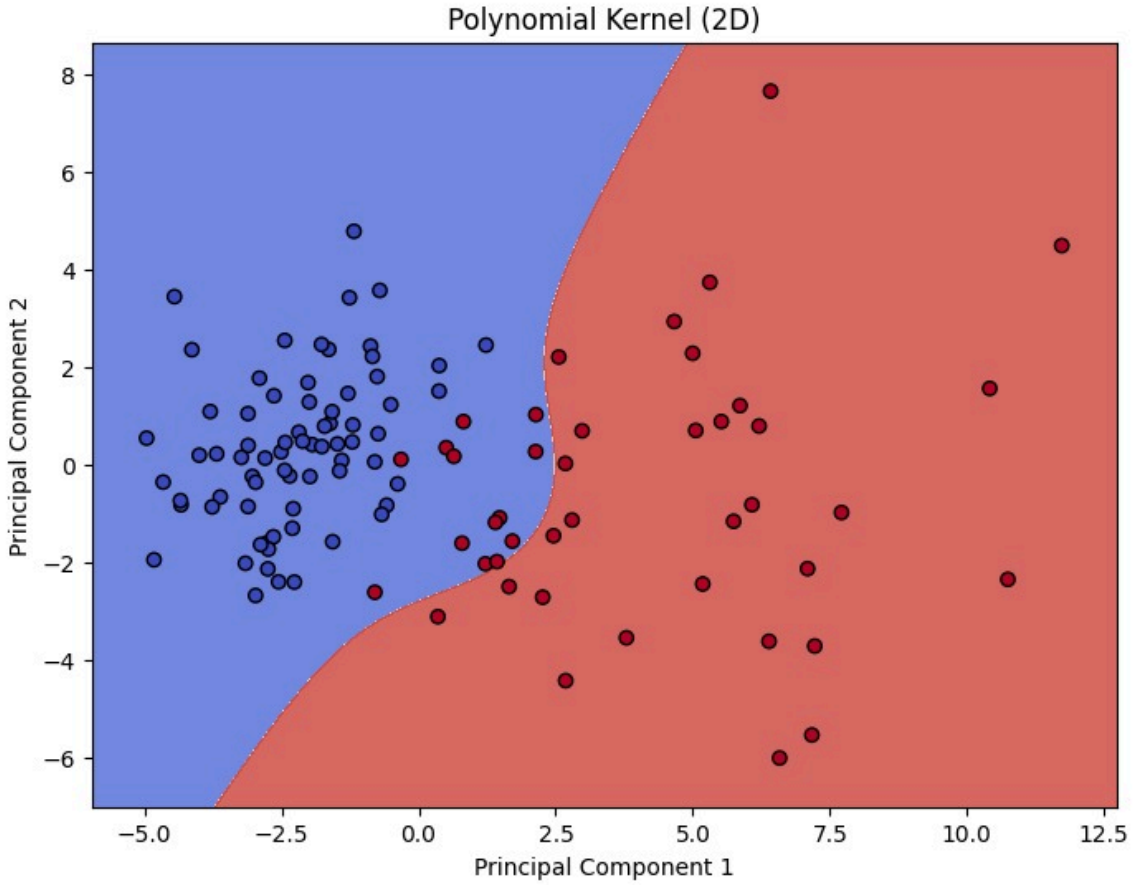
Accuracy: 95.61%
Confusion Matrix:
[[68 3]
[2 41]]
Classification Report:
Precision: 0.97 (Class 0), 0.93 (Class 1)

RBF Kernel



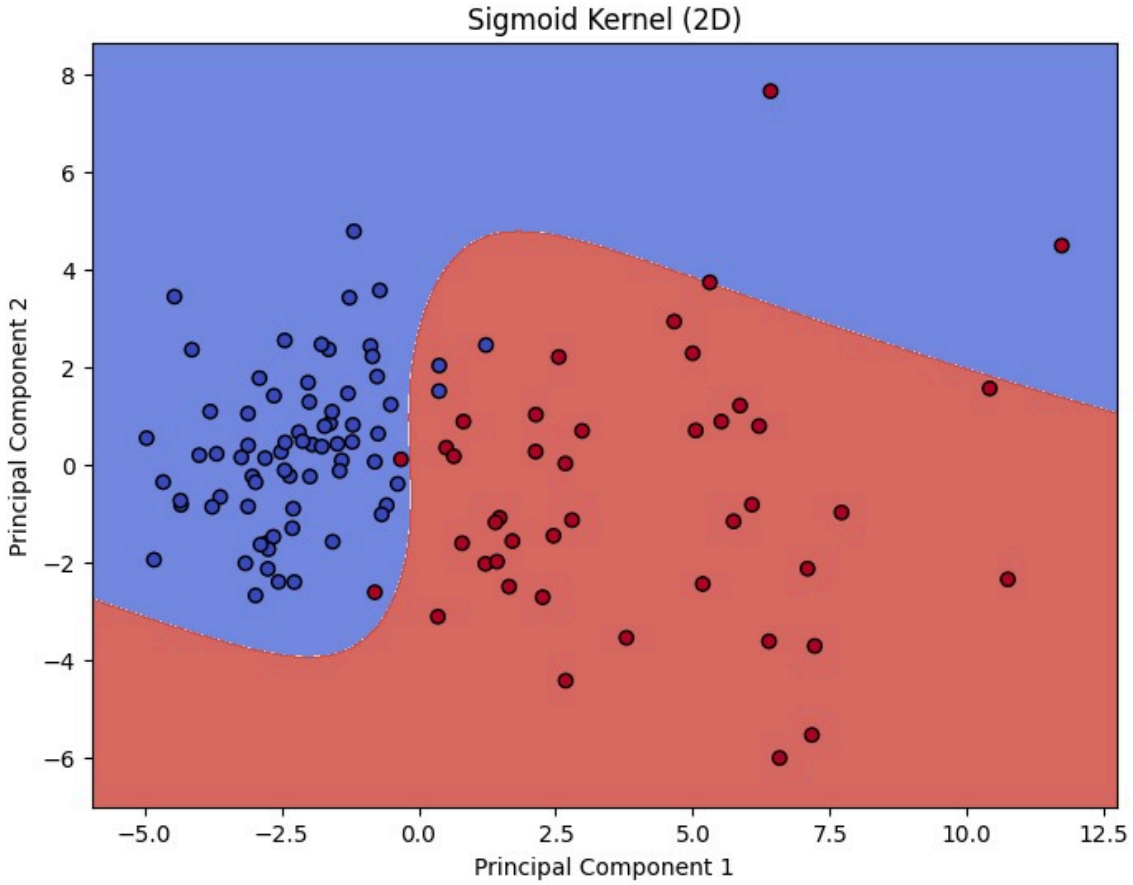
Accuracy: 98.25%
Confusion Matrix:
[[71 0]
[2 41]]
Classification Report:
Precision: 0.97 (Class 0), 1.00 (Class 1)

Polynomial Kernel



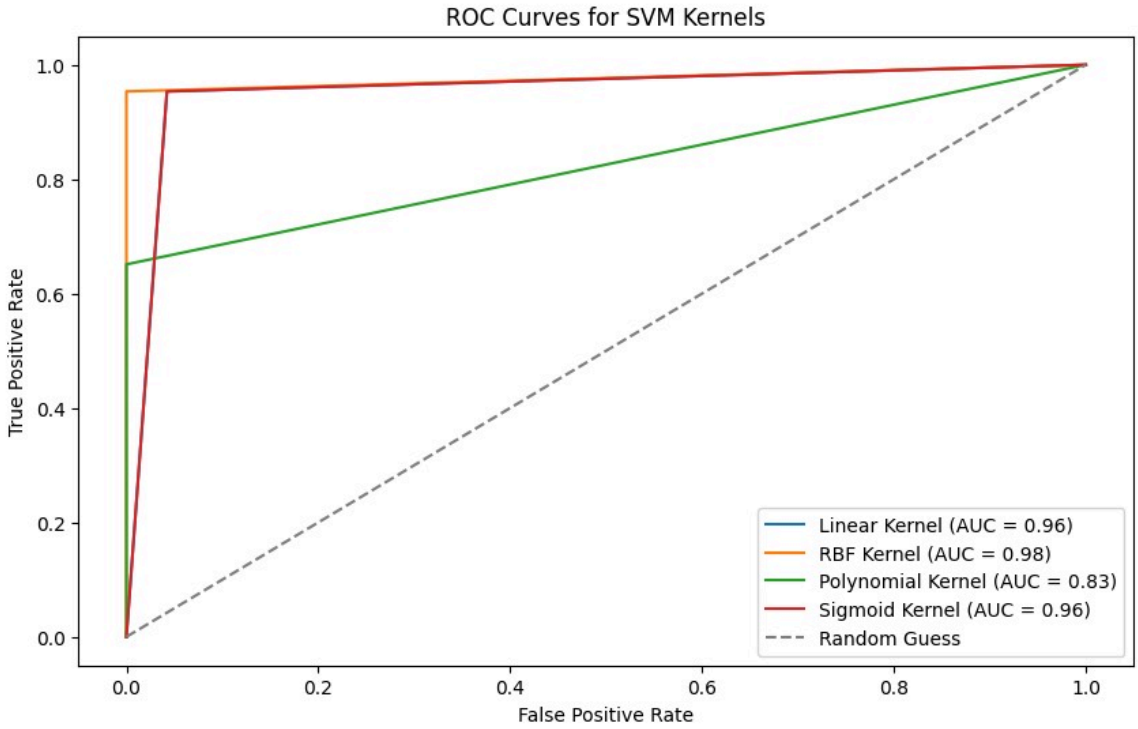
Accuracy: 86.84%
Confusion Matrix:
[[71 0]
[15 28]]
Classification Report:
Precision: 0.83 (Class 0), 1.00 (Class 1)

Sigmoid Kernel



Accuracy: 95.61%
Confusion Matrix:
[[68 3]
[2 41]]
Classification Report:
Precision: 0.97 (Class 0), 0.93 (Class 1)

3.2 ROC Curves



ROC curves were plotted for all four kernels, and AUC scores were compared. The RBF kernel had the highest AUC, indicating better overall performance.

3.3 Decision Boundaries

Decision boundaries for different kernels were plotted in 2D using PCA. The RBF kernel showed the most flexible and accurate boundary.

4. Discussion

This section provides an in-depth analysis of why each kernel performed as observed, including mathematical explanations and visual intuition to make the concepts accessible to readers with varying levels of technical knowledge.

4.1 Why is RBF Kernel Performing the Best?

Mathematical Explanation

The RBF Kernel is defined as:

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2)$$

where:

- γ controls the "spread" of the kernel (small γ = broader, large γ = tighter fit).

- $|x_i - x_j|^2$ is the squared Euclidean distance between two data points.

Why it Works Best for This Dataset

- **Non-linearity**: The RBF kernel maps features into an infinite-dimensional space, allowing it to model complex decision boundaries.
- **Visual analogy**: Imagine stretching a rubber sheet over the data points; the RBF kernel can create smooth "hills" and "valleys" to separate classes.
- **Adaptability**: The γ parameter adjusts flexibility:
  - Small γ: Smoother boundaries (better generalization).
  - Large γ: Tighter boundaries (risk of overfitting).
- For this dataset, the default γ in scikit-learn provided a balance.

Performance Analysis

- **Accuracy**: 98.25% (Highest among all kernels).
- **Confusion Matrix**: Only 2 misclassifications (both malignant → benign).
- **Key Insight**: The RBF kernel's ability to model local relationships (via Gaussian decay) matches the dataset's inherent non-linearity.

4.2 Why is the Linear Kernel Lagging?

Mathematical Explanation

The linear kernel is simply the dot product:

$$K(x_i, x_j) = x_i \cdot x_j$$

It assumes data is separable by a straight hyperplane in the original feature space.

Limitations for This Dataset

- **Non-linear separability**: Breast cancer data often has overlapping feature distributions (e.g., tumor radius vs. texture).
- **Example**: Malignant and benign tumors may share similar feature ranges, making a straight boundary suboptimal.

Misclassifications

- 3 benign → malignant (false positives).
- 2 malignant → benign (false negatives).

Comparison to RBF

- The linear kernel's accuracy (95.61%) is still strong but fundamentally limited by its simplicity.

- **When to use**: Ideal only if data is linearly separable (rare in real-world medical data).

4.3 Why is the Polynomial Kernel Underperforming?

Mathematical Explanation

The polynomial kernel of degree d is:

$$K(x_i, x_j) = (\gamma x_i \cdot x_j + r)^d$$

where d=3, γ=1 and r=0 (default in scikit-learn).

Key Issues

- **Overfitting**: The high degree d=3 creates overly complex boundaries:
  - **Visual analogy**: Like fitting a wavy line through points, capturing noise instead of trends.
- **Poor Recall (65%) for Malignant Cases**:
  - Misclassified 15 malignant tumors as benign (high false negatives).
  - **Clinical implication**: Dangerous for diagnostics, as missing malignancies is costlier than false alarms.

Trade-off

- Polynomial kernels excel only when data has inherent polynomial relationships (uncommon in biological data).
- **Why RBF is better**: RBF adapts to local patterns, while the polynomial kernel forces a global polynomial structure.

4.4 Why is the Sigmoid Kernel Similar to Linear?

Mathematical Explanation

The sigmoid kernel mimics neural network activation:

K(x_i, x_j) = tanh(γ x_i · x_j + r)

$$K(x_i, x_j) = tanh(\gamma x_i \cdot x_j + r)$$

For this dataset γ=1 and r=0, and (default).

Behavior Analysis

- **Resembles Linear Kernel**:

- For small inputs, , making it quasi-linear.
- **Result**: Decision boundaries are nearly straight (like the linear kernel).

Performance (95.61% accuracy)

- **Confusion matrix identical to the linear kernel**.

Why?

- The sigmoid's non-linearity isn't leveraged effectively here because:
  - Data isn't separable by hyperbolic tangent transformations.
  - Default parameters don't induce enough curvature.

When to Use Sigmoid?

- Primarily for neural network-inspired models or when data has sigmoidal separability (rare in practice).

Key Takeaways for Non-Technical Readers

- **RBF is the Gold Standard**:
  - Thinks "flexibly" to wrap around data clusters.
  - Like drawing boundaries with a pencil vs. a ruler (linear kernel).
- **Linear is Simple but Limited**:
  - Uses straight boundaries—good for simple patterns.
- **Polynomial is Overkill**:
  - Tries to fit "squiggly" boundaries, leading to errors.
- **Sigmoid is a Niche Tool**:
  - Acts like a linear kernel here due to data structure.

Mathematical Summary Table

Kernel	Formula	Strengths	Weaknesses
Linear	$x_i \cdot x_j$	Simple, fast	Inflexible for non-linear data
RBF	$exp(-\gamma \|x_i - x_j\|^2)$	Highly adaptable	Sensitive to γ
Polynomial	$(\gamma x_i \cdot x_j + r)^d$	Captures polynomial trends	Prone to overfitting
Sigmoid	$tanh(\gamma x_i \cdot x_j + r)$	Neural-like	Rarely outperforms RBF

5. Conclusion

5.1 Summary of Findings

The RBF kernel performed the best, followed by the linear and sigmoid kernels, while the polynomial kernel struggled.

5.2 Recommendations

For this dataset, the RBF kernel is the best choice. Hyperparameter tuning can further improve results.

5.3 Future Work

Future work can explore ensemble methods and hyperparameter optimization.

6. References

https://www.researchgate.net/profile/Jwan-Alwan/publication/331108441_Performance_Evaluation_of_Kernels_in_Support_Vector_Machine/links/6130ed9ec69a4e487975d000/Performance-Evaluation-of-Kernels-in-Support-Vector-Machine.pdf

Breast Cancer Wisconsin Dataset:
https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset

7. Code Implementation (GitHub)

The full Python code used for preprocessing, training, evaluation, and visualization is provided.

https://github.com/mm315/SVM_Kernel_Comparison.git