

ROHITH RAMESH

www.linkedin.com/in/rohithr24

rohithrc8464@gmail.com

Understanding Support Vector Machines (SVMs) Made Simple

What is SVM?

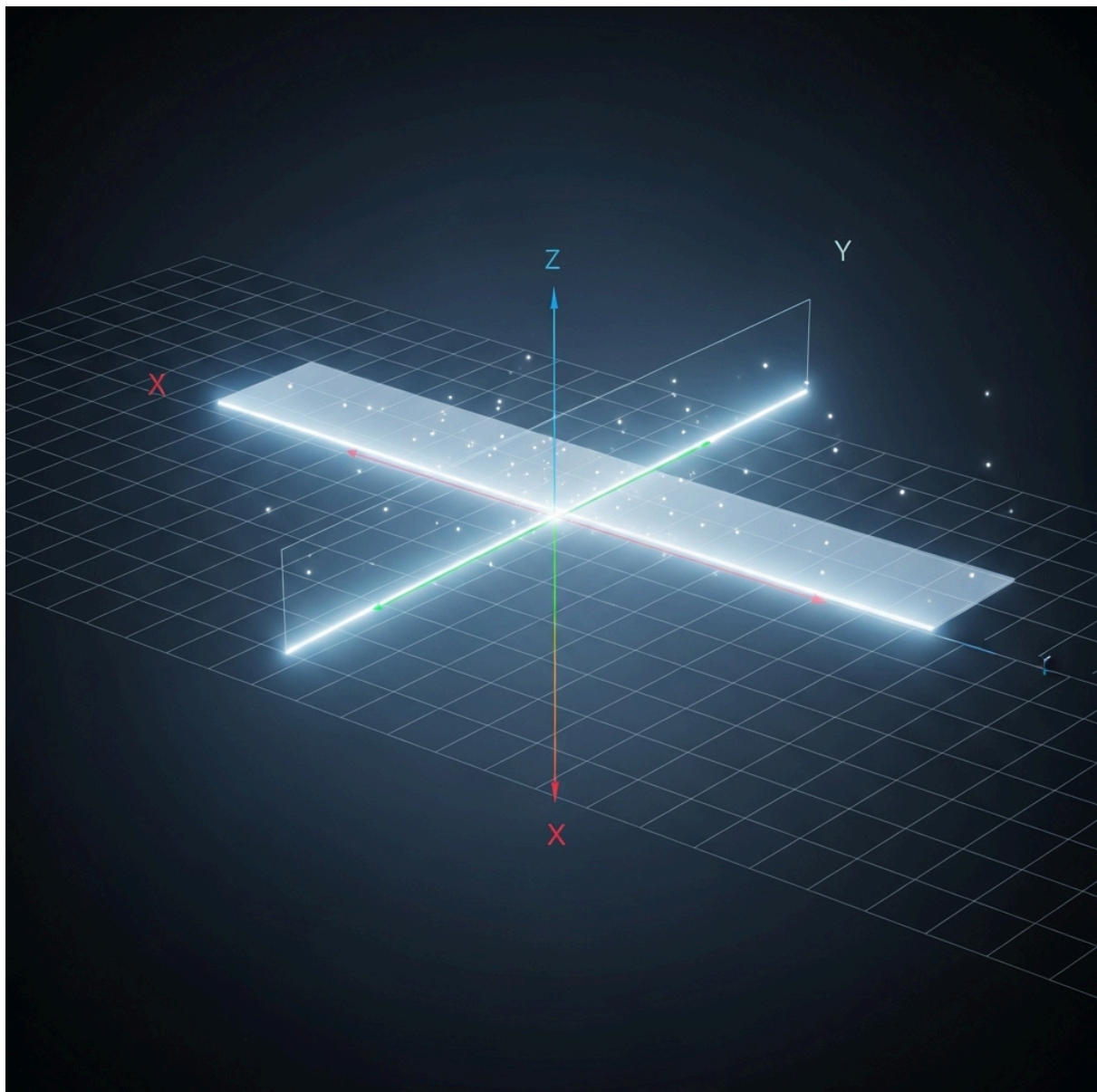
SVM or support vector machine is a Supervised Learning Model

Used to build Classification and Regression models

It classifies input data based on a hyper-plane

Now what is a hyper plane?

(**A hyperplane** is a flat, dividing surface that splits a space into two separate parts.)

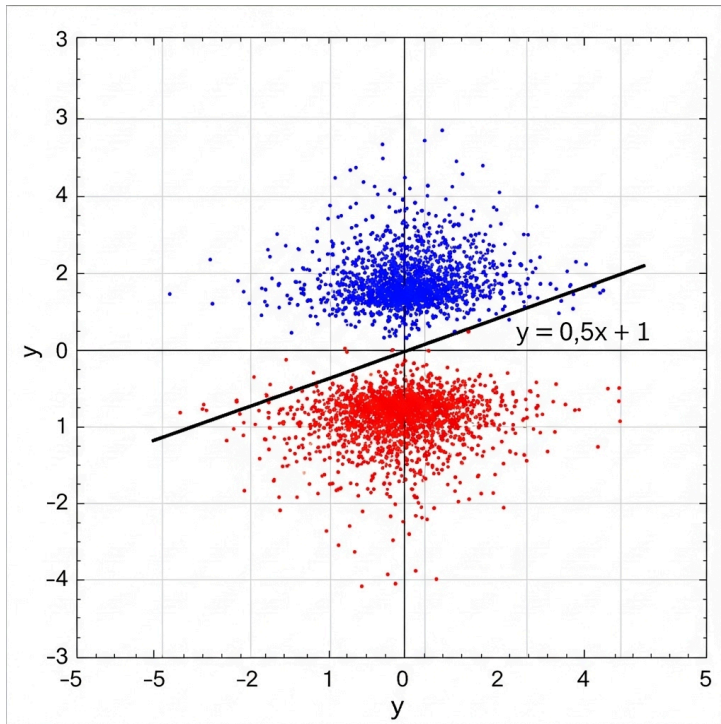


SVM tries to divide the data into 2 classes by finding decision boundary

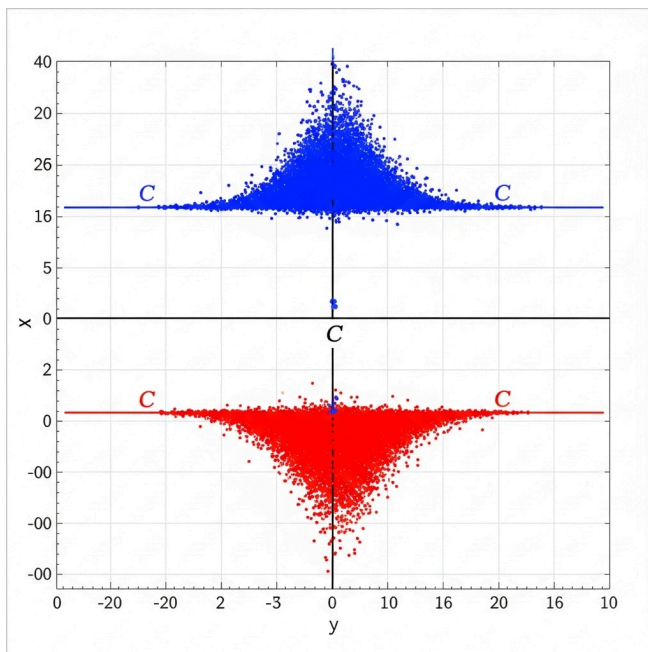
Now what is the decision boundary?

(Think of it as a "fence" the model builds to distinguish one group from another)

Main goal of SVM is to find this margin it segregate a dataset in 2 parts (this is a sample image of hyperplane in 2 dimensional)



How svm works (it draws a softer margin on both sides, Which is represented by smaller c and harder margin C (which is the origin line)



The svm works on this equation on $w \cdot x + b = 1$ for positive c (blue) and $w \cdot x + b = 0$ for origin line and $w \cdot x + b = -1$ negative (red)

Now you guys will be like what is this equation ?

Let's break it down.

If equation returns greater than 0 then its positive(blue) $w \cdot x + b = 1$

If the equation evaluates to a value less than 0, it signifies a negative result, indicating that it falls beneath the origin line (marked in red):

$$w \cdot x + b = -1$$

$$w \cdot x + b = 0$$

- **w** = weight vector
- **x** = input feature vector (your data point)
- **b** = bias

The weight vector **w** is always perpendicular (or "normal") to the hyperplane

Now you guys might be curious about this mathematical equation.

Imagine a pencil lying on a table. The table is the hyperplane. A pen standing straight up on the table represents the weight vector **w**. If you tilt the pen, the table (hyperplane) would have to tilt with it to remain perpendicular.

So your goal is to minimize the **w**

The bias **b** term defines the **position** or **offset** of the hyperplane.

The bias **b** moves the hyperplane away from the origin without changing its orientation.

So the best combination of **w** and **b** to create a hyperplane that not only separates the two classes but also maximizes the margin (the distance between the hyperplane and the closest data points)

Scikit learn offers various functions to classify different data

linear: Creates a simple, straight-line decision boundary for linearly separable data.

poly: Creates a curved decision boundary, useful for data with polynomial relationships.

rbf: The powerful default choice, which creates complex, non-linear boundaries for most datasets.

sigmoid: A niche kernel that mimics the activation function of a neural network.also used in logistic Regression.

You'll learn these features as you go, so don't worry too much

Why use SVM

- It is extremely good in overfitting the data (meaning it memorizes the data)
- It's very effective in higher dimension (you can classify data even when you have multiple features)

Limitation

Slow training on large datasets

Sensitive to noise or overlapping or outliers (Imagine you're learning to identify pictures of cats. Someone gives you a stack of 100 photos to learn from, but two of them are secretly pictures of dogs that have been mislabeled as "cat.")

sensitive to kernel and regularization parameters, it means its performance can change dramatically with even small adjustments to these settings. It's like a race car that's very fast but difficult to drive – you need to tune it perfectly to get a good result, and a small mistake can lead to a poor outcome

Now the spicy part

Where you can use SVM models

- Image classification task (like finding difference between cats and dogs)
- Recognising handwriting to digital characters
- Also good for spam detection (which I have done a project on, you can check that too)
- Speech recognition and noise filtering