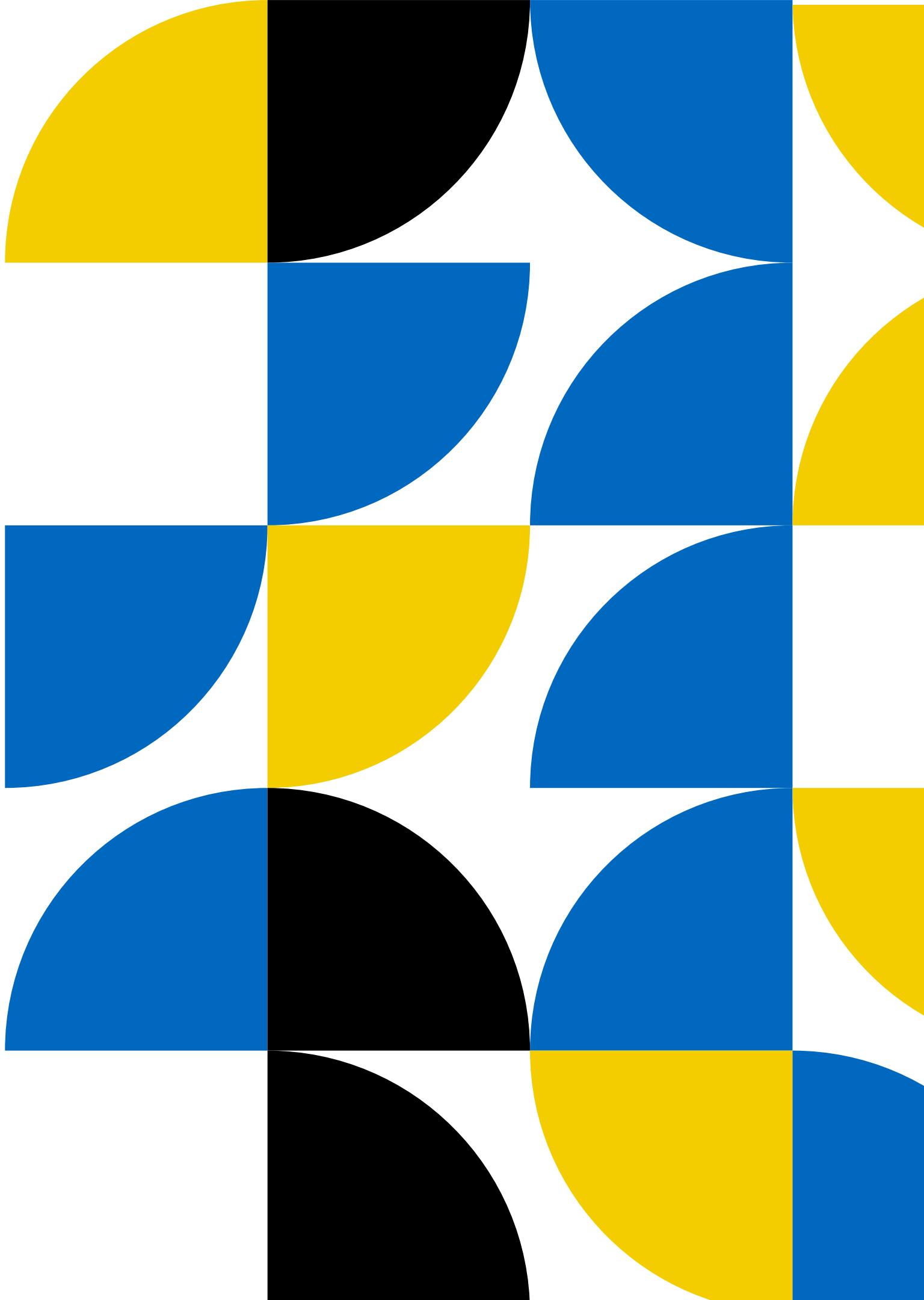




Support Vector Machine

Data Science - TRPL 5A

Oleh Kelompok 3



LANJUT





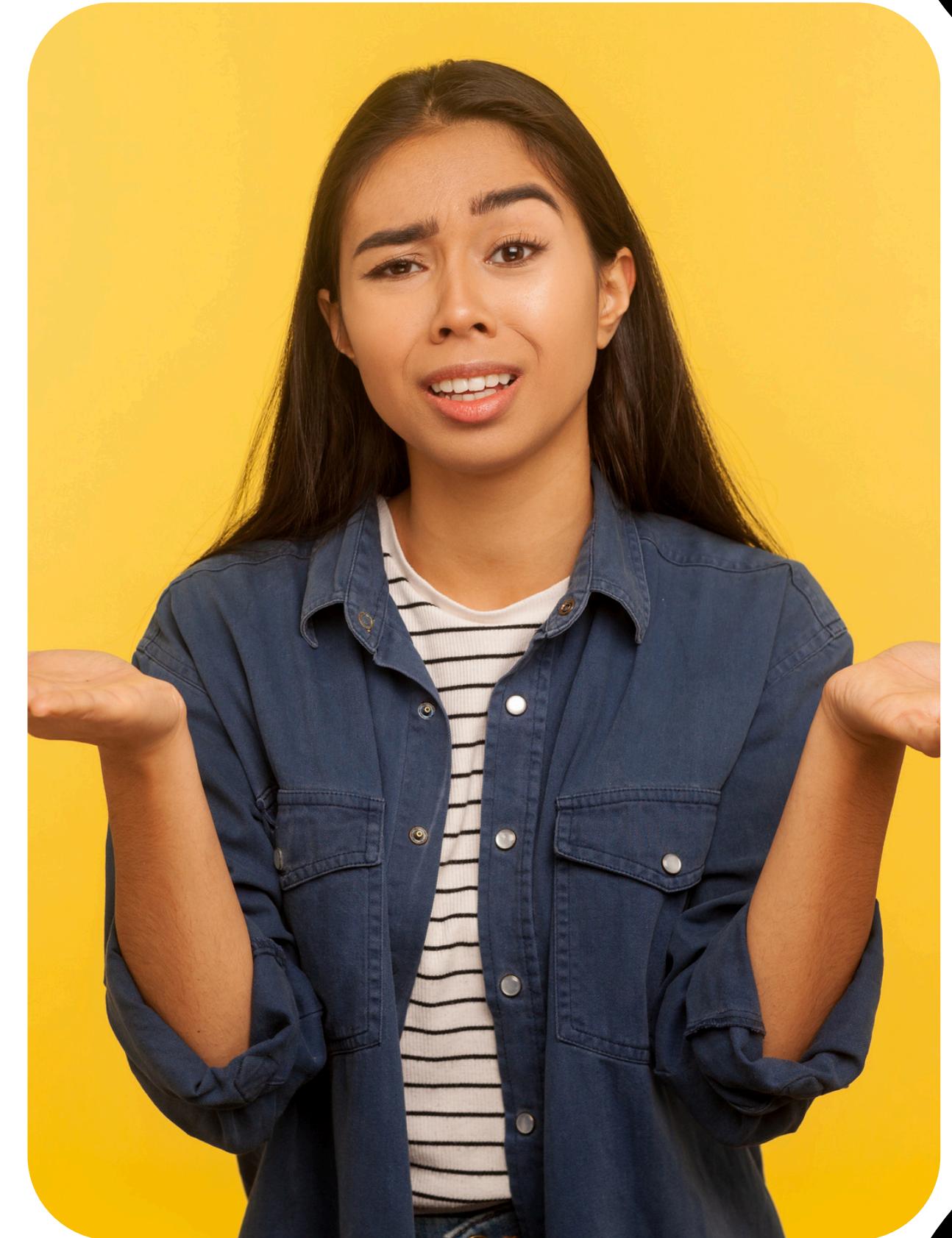
Anggota

- 01 Arya Yudha Prasetya - 234311007
- 02 Richo Novian Saputra - 234311024
- 03 Shaffa Dwiaji Feryansyah Putra - 234311028

Apa Itu SVM?

Support Vector Machine (SVM) adalah algoritma Supervised Learning yang digunakan untuk klasifikasi dan regresi, berlandaskan pada prinsip structural risk minimization (SRM).

SVM bekerja dengan mengolah data untuk menghasilkan hyperplane dengan memisahkan ruang input menjadi dua kelas, dimulai dari pengelompokan kasus linier yang dipisahkan oleh hyperplane.



Tujuan SVM

Berikut adalah tujuan dari Support Vector Machine :

- 1. Memisahkan data ke dalam kelas yang berbeda seakurat mungkin** (mencari pembatas (hyperplane) yang mampu memisahkan kelas-kelas data).
- 2. Memaksimalkan margin** (semakin besar margin maka model makin stabil dan tidak mudah salah (generalisasi lebih baik)).
- 3. Menangani data yang tidak linear** (SVM dapat memetakan data ke ruang berdimensi lebih tinggi sehingga bisa dipisahkan).
- 4. Mencari solusi optimal secara matematis.**
- 5. Menghasilkan model yang robust** (SVM tahan terhadap outlier dan bekerja baik pada data dengan dimensi tinggi).

Kelebihan SVM

1. Efektif pada dataset berdimensi tinggi
2. Bekerja optimal meski jumlah sampel sedikit
3. Hasil akurasi tinggi pada banyak kasus
4. Memiliki generalisasi yang baik

Kekurangan SVM

1. Lambat pada dataset berjumlah besar
2. Pemilihan kernel dan parameter cukup sulit (perlu tuning)
3. Kurang cocok untuk dataset dengan banyak noise
4. Interpretasi model kurang intuitif

Komponen Utama SVM

1. Hyperplane

4. Kernel Function

2. Margin

5. Parameter C

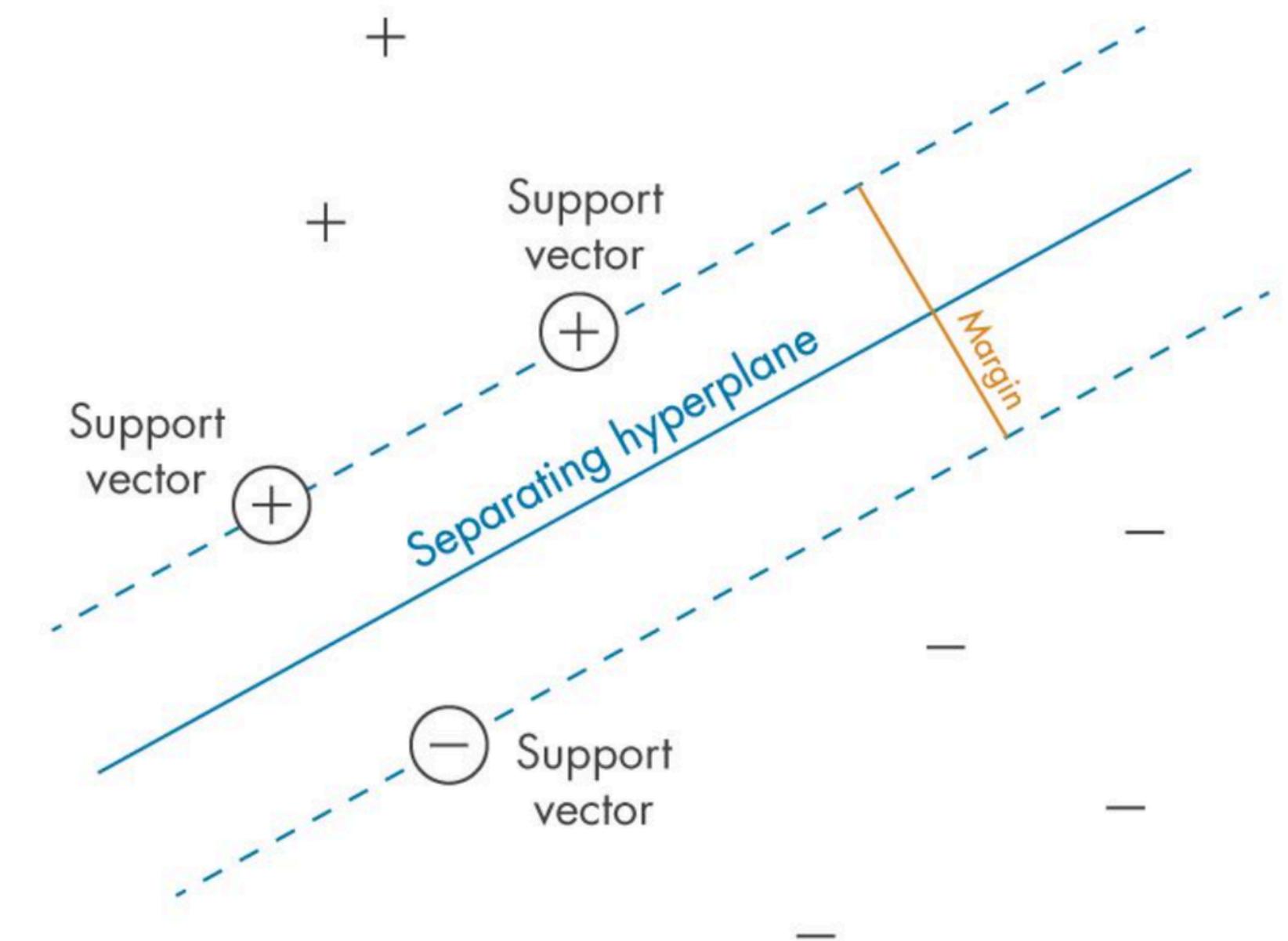
3. Support Vectors

6. Hinge Loss

HYPERPLANE

Hyperplane

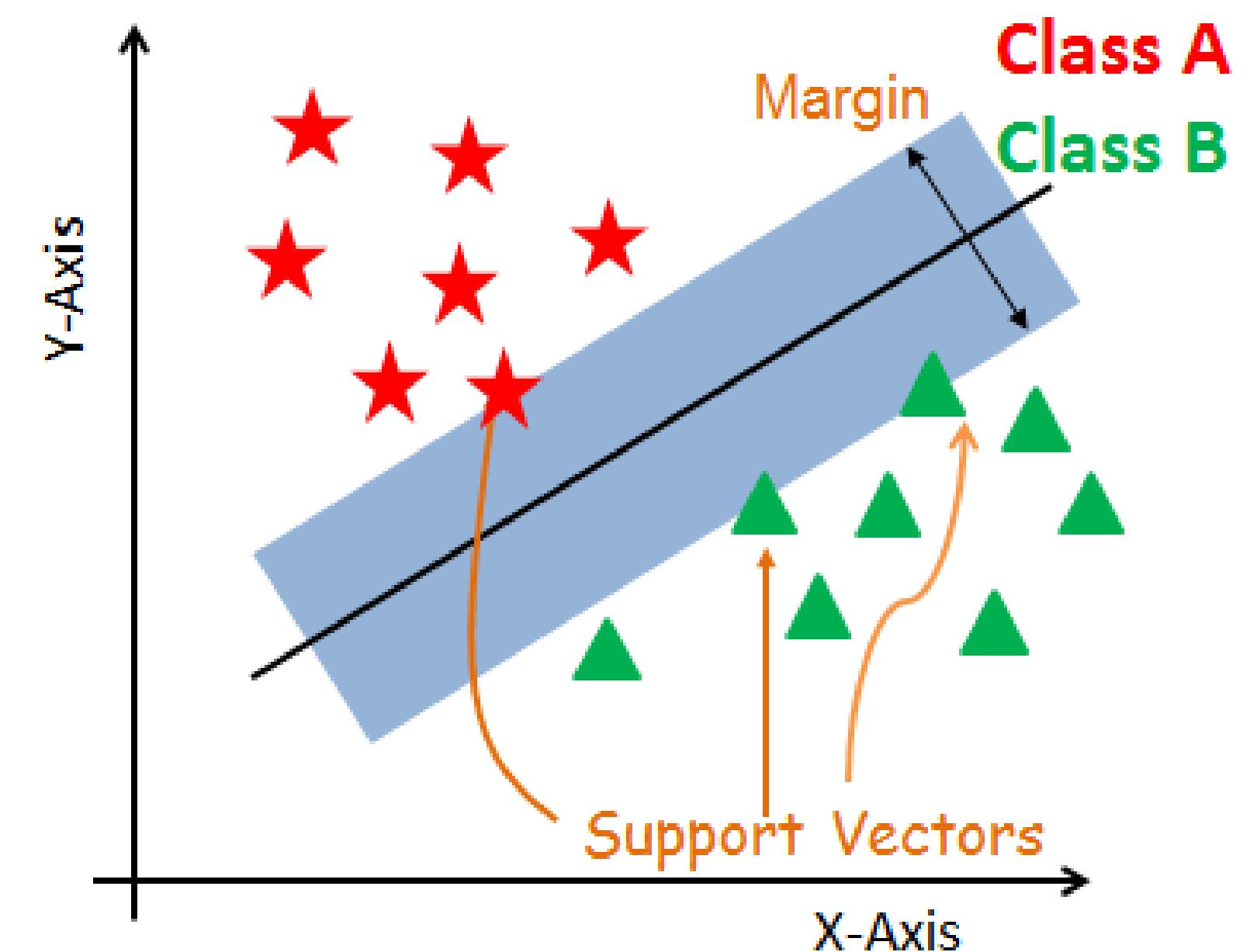
Garis (2D), bidang (3D), atau permukaan berdimensi tinggi yang digunakan untuk memisahkan dua kelas.



MARGIN

Margin

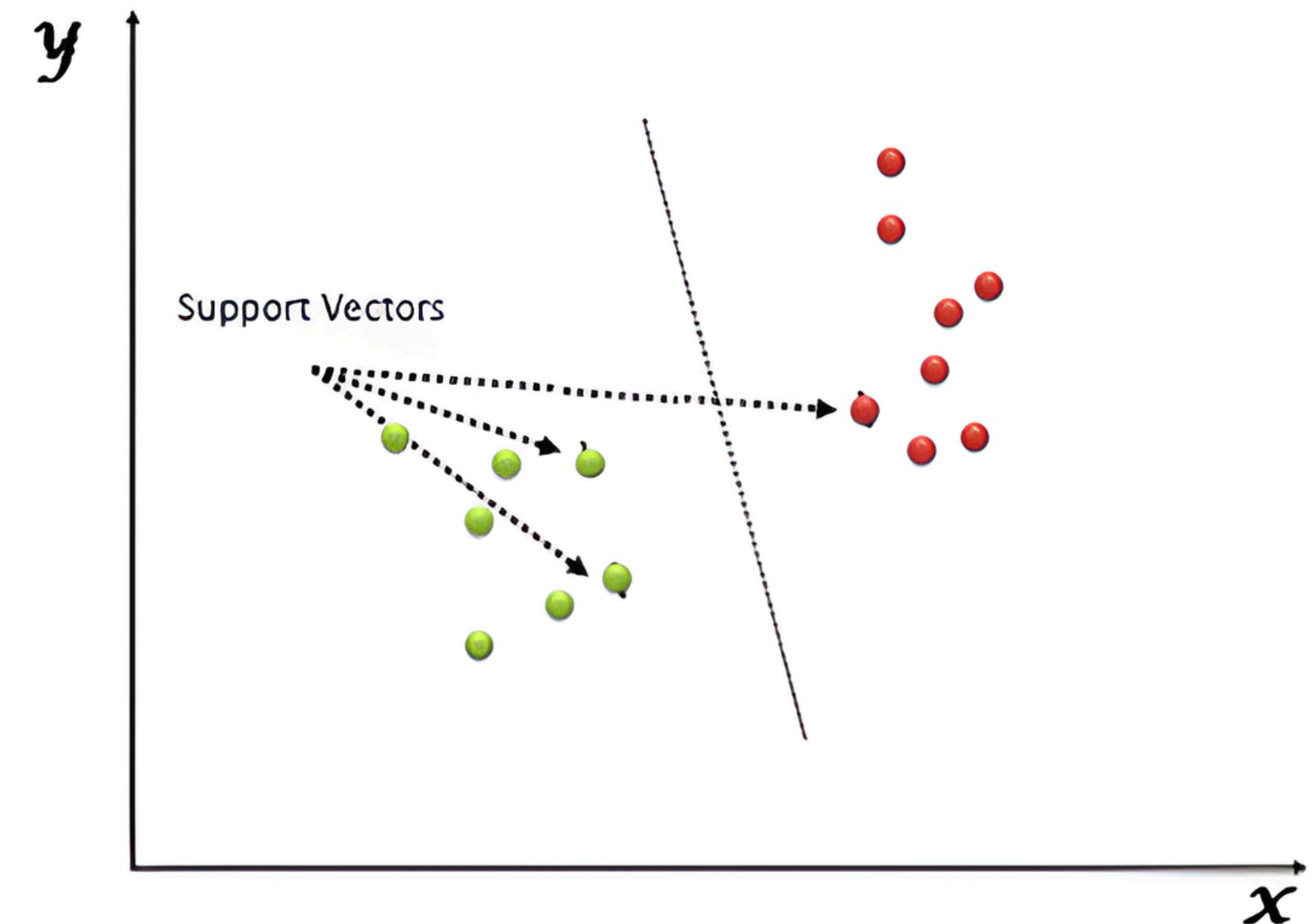
Jarak antara hyperplane dan support vectors terdekat dari masing-masing kelas.



SUPPORT VECTORS

Support Vectors

Titik data terdekat ke hyperplane yang menentukan posisi dan arah hyperplane.

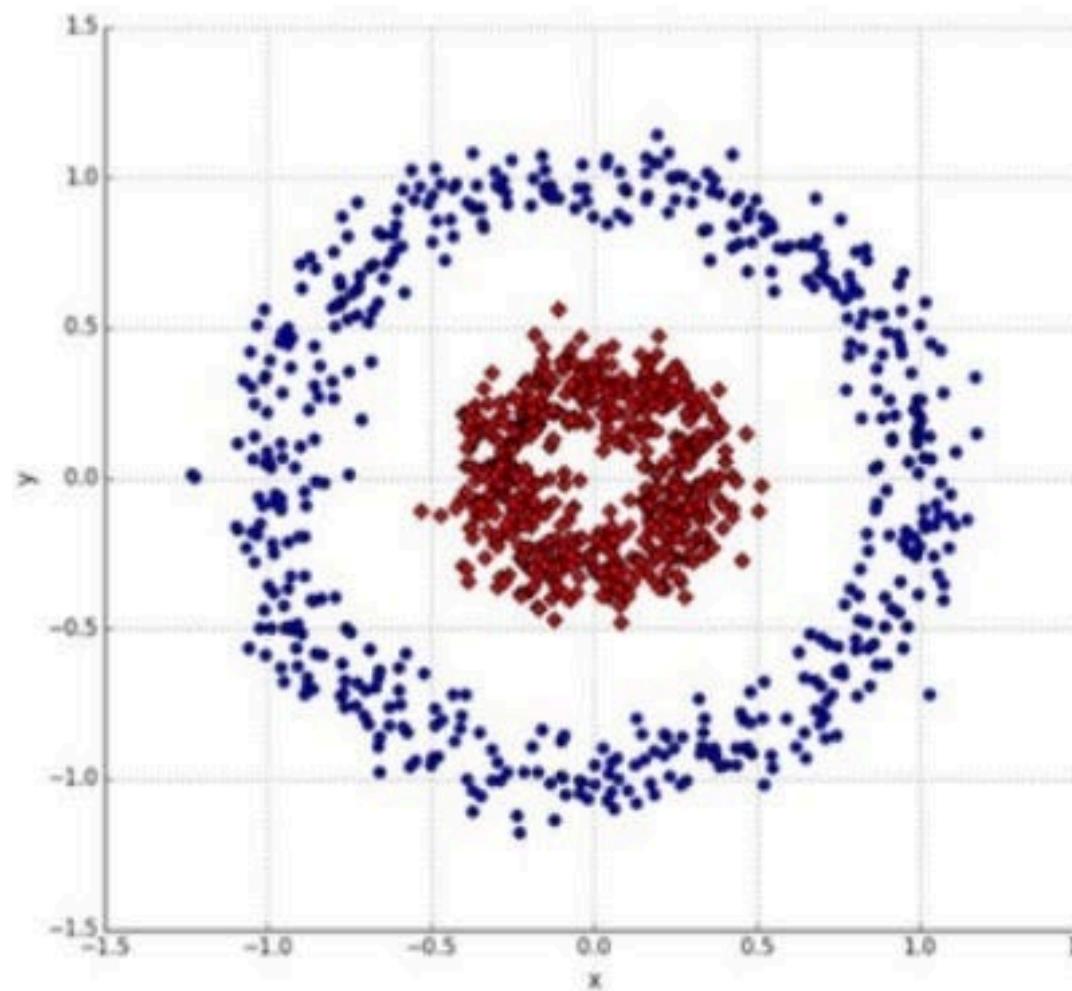


KERNEL FUNCTION

Kernel Function

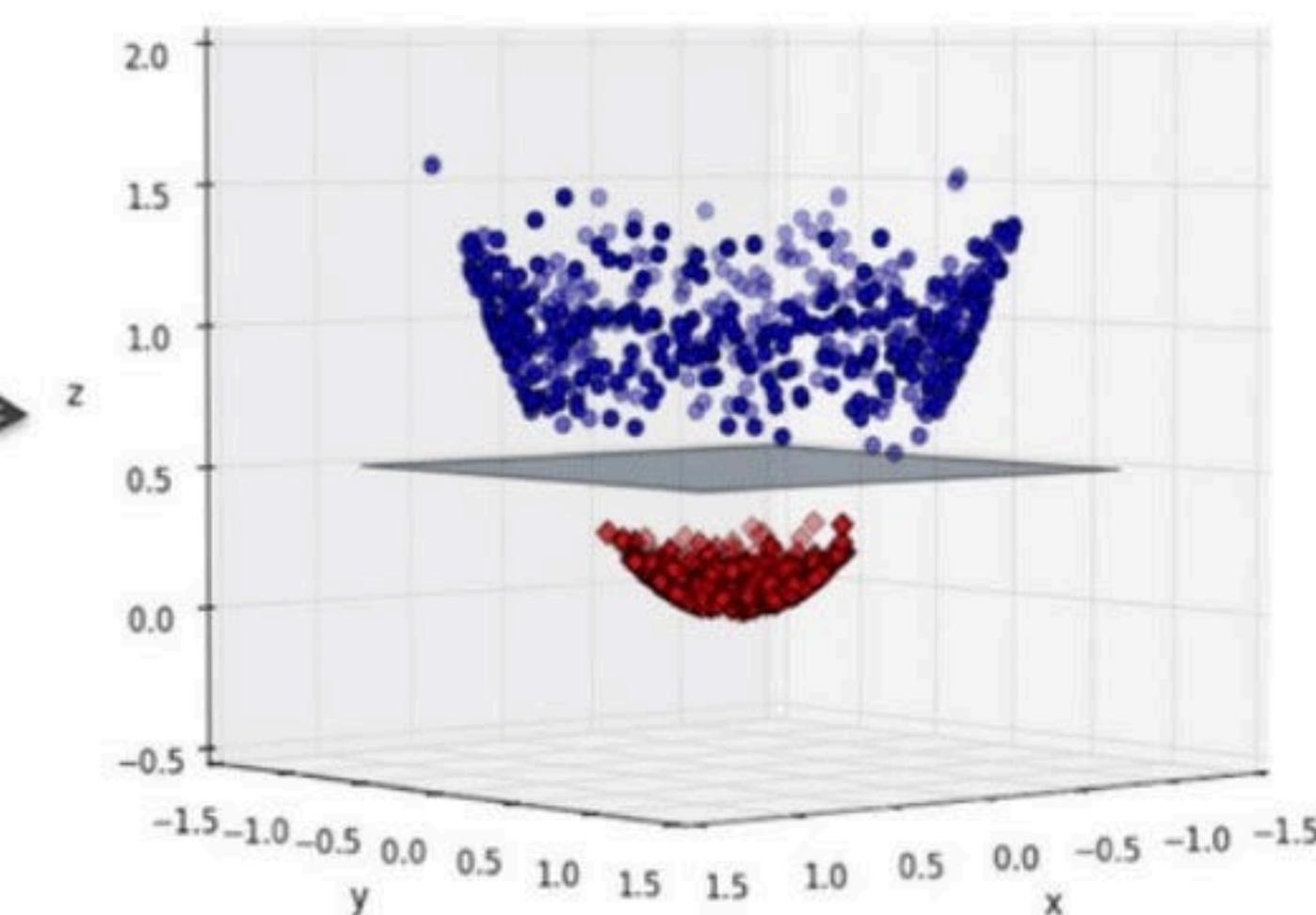
Memetakan data ke ruang fitur berdimensi lebih tinggi untuk membuat data lebih mudah dipisahkan secara linear.

2D



Kernel

3D

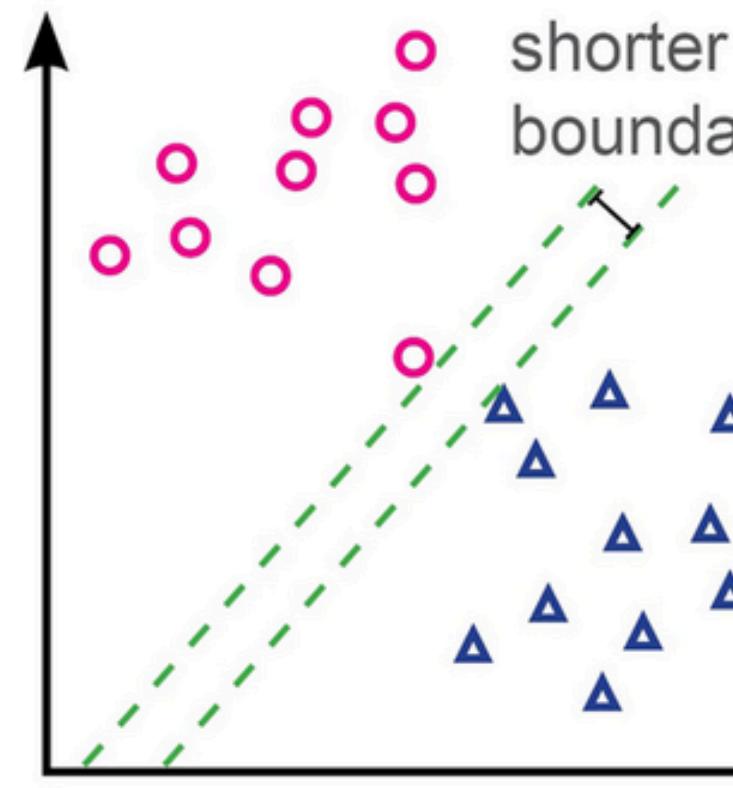


PARAMETER C

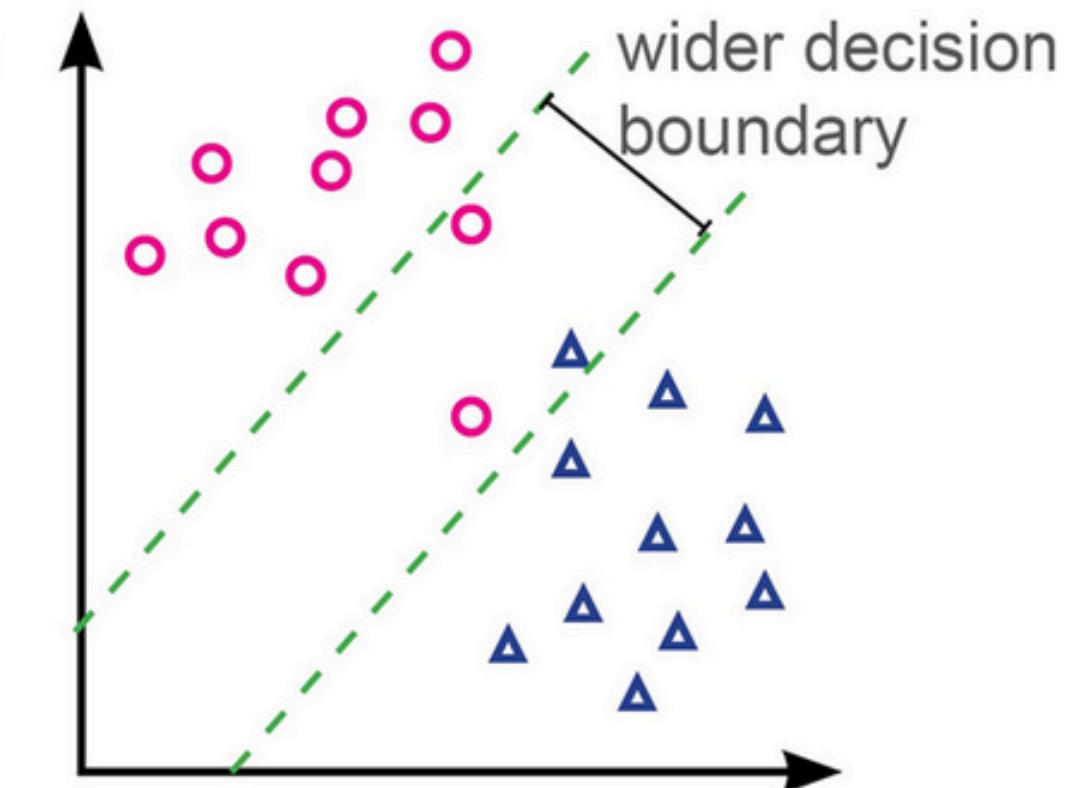
Parameter C

Pengatur tingkat toleransi terhadap kesalahan yang memengaruhi seberapa ketat margin SVM.

$C = \text{large value}$



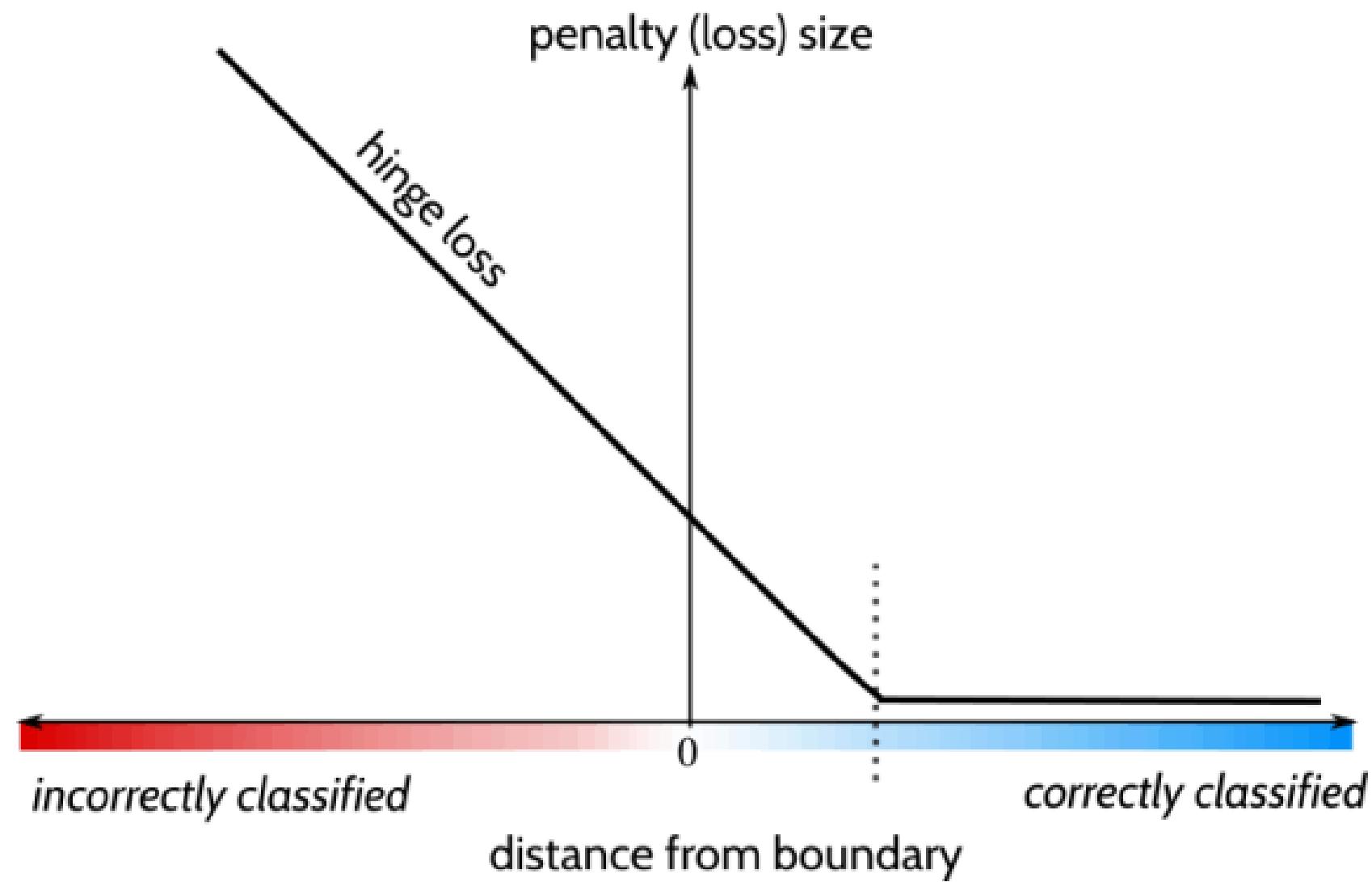
$C = \text{small value}$



HINGE LOSS

Hinge Loss

Menghitung penalti ketika titik data berada di sisi hyperplane yang salah atau terlalu dekat.



Klasifikasi SVM

1. Linear SVM

SVM linear digunakan dengan data yang dapat dipisahkan secara linier, ini berarti bahwa data tidak perlu menjalani transformasi apa pun untuk memisahkan data ke dalam berbagai kelas yang berbeda.

2. Non-Linear SVM

SVM non-linear digunakan ketika data tidak dapat dipisahkan secara linier, sehingga diperlukan transformasi ruang fitur menggunakan kernel trick untuk memetakan data ke dimensi yang lebih tinggi agar dapat dipisahkan oleh hyperplane.

Jenis kernel SVM

1. Linear Kernel

Rumus:

$$K(x, x') = \mathbf{x}^T \mathbf{x}'$$

Linear kernel digunakan ketika data dapat dipisahkan oleh garis lurus (atau hiperplane). Kernel ini tidak melakukan transformasi apa pun ke ruang fitur lain, sehingga perhitungannya sangat cepat dan ringan.

2. Polynomial Kernel

Rumus:

$$K(x, x') = (\mathbf{x}^T \mathbf{x}' + c)^d$$

Kernel polynomial melakukan pemetaan non-linear menggunakan polinomial derajat tertentu (d). Artinya, data yang tidak linear pada ruang asli bisa menjadi linear pada ruang polinomial. Semakin tinggi derajat polinomialnya, semakin kompleks bentuk decision boundary-nya.

Jenis kernel SVM

3. RBF Kernel (Gaussian Kernel)

Rumus:

$$K(x, x') = e^{-\gamma \|x - x'\|^2}$$

RBF Kernel dapat memetakan data ke ruang fitur berdimensi sangat tinggi secara otomatis. Kernel ini sangat fleksibel dan dapat membentuk boundary yang sangat kompleks.

4. Mencari solusi optimal secara matematis.

Rumus:

$$K(x, x') = \tanh(\alpha x^T x' + c)$$

Sigmoid kernel meniru fungsi aktivasinya neural network perceptron. Kernel ini bisa bekerja dengan baik untuk beberapa dataset tertentu, tetapi tidak selalu stabil seperti RBF atau polynomial.

Hipotesa Function SVM

Hipotesa Function pada Support Vector Machine adalah fungsi yang digunakan untuk menentukan keputusan klasifikasi, yaitu memutuskan apakah sebuah data akan masuk ke kelas tertentu atau tidak.

Dalam SVM, proses pengambilan keputusan dilakukan dengan melihat posisi sebuah titik data terhadap hyperplane (garis pemisah). Hipotesa Function inilah yang menghitung jarak dan menentukan sisi mana suatu data berada.

Untuk SVM linear, hipotesa function-nya adalah:

$$h(x) = w^T x + b$$

di mana:

- w : vektor bobot (weight)
- x : vektor fitur input
- b : bias
- w^T : transpos dari vektor bobot w
- Tanda $h(x)$ menentukan kelas:
 - Jika $h(x) \geq 0 \rightarrow$ kelas +1
 - Jika $h(x) < 0 \rightarrow$ kelas -1

Cost Function SVM

Cost Function pada Support Vector Machine adalah ukuran yang digunakan untuk menilai seberapa baik sebuah hyperplane memisahkan data. Fungsi ini digunakan selama proses pelatihan untuk membantu SVM menemukan garis pemisah yang paling optimal.

Dalam SVM, tujuan utamanya adalah mencari hyperplane dengan margin terbesar, yaitu jarak terluas antara garis pemisah dan titik data terdekat. Namun di saat yang sama, model juga harus meminimalkan kesalahan klasifikasi, terutama ketika data memiliki noise atau tidak terpisah secara sempurna.

Terdapat tiga bentuk dari Cost Function SVM:

1. Cost Function Hard Margin
2. Cost Function Soft Margin
3. Cost Function dengan Hinge Loss

Cost Function Hard Margin

Cost function ini dipakai ketika data benar-benar bisa dipisahkan secara sempurna tanpa ada data yang salah posisi. Dalam kondisi ideal seperti ini, SVM hanya perlu mencari garis pemisah yang margin-nya paling lebar, tanpa perlu mempertimbangkan adanya kesalahan.

$$\text{Cost} = \frac{1}{2} \|w\|^2$$

Karena tidak ada error, cost function-nya hanya fokus pada memperkecil nilai $\|w\|^2$, yang secara langsung berarti memperbesar margin.

Cost Function Soft Margin

Pada dunia nyata, data tidak selalu sempurna seperti seringnya noise, outlier, atau data yang sulit dipisahkan. Untuk itu, digunakan konsep Soft Margin, yang memungkinkan sedikit kesalahan demi mendapatkan model yang lebih fleksibel dan realistik.

Soft Margin Cost Function menyeimbangkan dua hal, yaitu memperbesar margin, dan mengurangi kesalahan klasifikasi yang diukur menggunakan variabel tambahan yang disebut slack variable. Parameter C mengatur seberapa besar penalti untuk kesalahan tersebut.

$$\text{Cost} = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

Keterangan:

- $\|w\|^2$: menjaga margin tetap besar.
- ξ_i (ksi-i): penalti untuk kesalahan klasifikasi.
- C menentukan trade-off:
 - C besar -> penalti kesalahan besar -> model lebih ketat (risiko overfitting).
 - C kecil -> lebih toleran kesalahan -> margin lebih besar (risiko underfitting).

Cost Function Hinge Loss

Hinge Loss adalah bentuk lain dari Soft Margin SVM, tetapi ditulis dengan menggunakan konsep Hinge Loss. Meskipun bentuk formulanya berbeda, konsep dan tujuan akhirnya sama, yaitu memperbesar margin dan mengurangi kesalahan.

Hinge Loss memberikan penalti ketika:

- Prediksi salah
- Prediksi benar tetapi terlalu dekat dengan margin

Namun jika prediksi benar dan jauh dari margin, maka loss-nya nol.

Single Sample:

$$\text{Cost}(x, y) = \max(0, 1 - y(w^T x + b))$$

atau

Total Function

$$J(w, b) = \frac{1}{2} \|w\|^2 + C \sum \max(0, 1 - y_i f(x_i))$$

Kesimpulan

Support Vector Machine (SVM) adalah algoritma klasifikasi yang bekerja dengan mencari hyperplane terbaik yang memisahkan data dua kelas dengan margin terbesar, menggunakan konsep support vector serta cost function yang mengontrol keseimbangan antara margin dan kesalahan, SVM dapat bekerja pada data linear maupun non-linear melalui penggunaan kernel, sehingga mampu memberikan performa yang kuat dan stabil pada berbagai bentuk data.



Terima Kasih