

# SVM

## 1. Cara kerja algoritma

SVM bertujuan untuk menemukan hyperplane di ruang fitur yang memisahkan kelas data dengan margin terbesar. Hyperplane ini adalah batas keputusan yang memaksimalkan jarak antara titik data dari kelas yang berbeda.

### 1. Membangun Hyperlane

SVM mencari hyperplane yang memisahkan data dari kelas yang berbeda dengan margin terbesar. Margin adalah jarak antara hyperplane dan titik data terdekat dari setiap kelas, yang disebut support vectors.

### 2. Menggunakan fungsi kernel

Untuk data yang tidak dapat dipisahkan secara linear, SVM menggunakan fungsi kernel untuk memetakan data ke ruang berdimensi lebih tinggi di mana data dapat dipisahkan secara linear. Beberapa kernel yang umum digunakan adalah:

- Linear kernel: Digunakan ketika data dapat dipisahkan secara linear.
- Polynomial kernel: Memetakan data ke ruang polinomial.
- Radial Basis Function (RBF) kernel: Memetakan data ke ruang berdimensi lebih tinggi menggunakan fungsi Gaussian.
- Sigmoid kernel: Memetakan data menggunakan fungsi sigmoid.

### 3. Menentukan hyperplane optimal

SVM memecahkan masalah optimasi untuk menemukan hyperplane yang memaksimalkan margin. Ini melibatkan meminimalkan fungsi biaya yang memperhitungkan margin dan kesalahan klasifikasi.

### 4. Klasifikasi data baru

Untuk mengklasifikasikan data baru, SVM menentukan di sisi mana dari hyperplane data tersebut berada. Berdasarkan posisi relatif terhadap hyperplane data tersebut berada, data baru diklasifikasikan ke dalam kelas yang sesuai.

### 5. Mengatasi overfitting

SVM dapat mengatasi overfitting dengan menyesuaikan parameter regulasi  $C$  yang mengontrol trade-off antara margin yang lebih besar dan kesalahan klasifikasi. Nilai  $C$  yang lebih kecil mengizinkan margin yang lebih besar dengan lebih banyak kesalahan, sedangkan nilai  $C$  yang lebih besar mengurangi kesalahan tetapi dengan margin yang lebih kecil.

### 6. Evaluasi

Setelah model dilatih, performanya dievaluasi menggunakan metrik seperti accuracy, precision, recall, dan F1-score pada data uji. Jika hasil tidak memuaskan, dapat dilakukan parameter tuning,

## 2. Perbandingan model buatan sendiri dan library

Berdasarkan hasil evaluasi, model SVM dari library scikit-learn memiliki akurasi 0.83m sedangkan model yang diimplementasikan sendiri memiliki akurasi 0.52. Perbedaan yang signifikan ini menunjukkan bahwa ada beberapa aspek dalam implementasi yang mungkin perlu diperbaiki atau disesuaikan. Berikut beberapa kemungkinan yang dapat menyebabkan perbedaan tersebut:

1. Ada kemungkinan kesalahan dalam pengimplementasian kernel RBF.
2. Nilai regularisasi  $\lambda$  yang tidak sesuai, yang dapat menyebabkan underfitting atau overfitting.
3. Pengimplementasian learning bisa menghambat konvergensi.
4. Pembaruan bias yang masih kurang tepat.

## 3. Improvement yang bisa dilakukan

- Hyperparameter tuning  
Tuning parameter  $\gamma$  untuk RBF kernel dan  $\lambda$  (C dalam terminologi SVM) untuk menemukan kombinasi yang optimal, ini bisa dilakukan dengan menggunakan grid search atau random search.
- Kernel selection  
Eksperimen dengan kernel yang berbeda seperti polynomial atau sigmoid selain linear dan RBF. Setiap kernel dapat menangkap pola yang berbeda dalam data.
- Learning rate schedule  
Implementasikan penjadwalan learning rate yang lebih canggih, seperti exponential decay atau adaptive learning rates, untuk membantu konvergensi yang lebih baik.