

# SVM

## 1. Cara kerja algoritma K-Means

Support Vector Classification (SVC) adalah metode klasifikasi yang bertujuan untuk menemukan *hyperplane optimal* yang memisahkan data dalam ruang fitur dengan margin terbesar. *Hyperplane* ini adalah batas keputusan yang membagi ruang fitur menjadi dua kelas. Model ini menggunakan *support vectors*, yaitu titik-titik data yang berada di dekat *hyperplane*, untuk menentukan margin terbaik. Langkah-langkah utama dalam algoritma SVC adalah sebagai berikut.

### Inisialisasi

- Set parameter C
- Set kernel menjadi 'linear'
- Set optimizer menjadi 'quadratic'

### Fit

- Jika optimizer adalah 'quadratic' dan kernel adalah 'linear':
  - Hitung ukuran data  $m$  (jumlah sampel) dan  $n$  (jumlah fitur)
  - Hitung matriks Gram  $K$  sebagai hasil dari dot product antara  $X$  dan  $X$  transpose
  - Definisikan matriks  $P$  sebagai hasil perkalian outer produk label  $y$  dengan  $K$
  - Definisikan vektor  $q$  sebagai  $-1$
  - Definisikan matriks  $G$  dan vektor  $h$  untuk batasan
  - Definisikan matriks  $A$  dan vektor  $b$  untuk batasan kesamaan
  - Pecahkan masalah kuadratik dengan menggunakan solver untuk mendapatkan vektor  $\alpha$
  - Tentukan support vectors berdasarkan  $\alpha$  yang lebih besar dari ambang batas
  - Hitung bobot  $w$  sebagai hasil dari dot product antara  $\alpha$ , label support vectors, dan support vectors
  - Hitung bias  $b$  sebagai rata-rata selisih antara label support vectors dan hasil dot product antara support vectors dan bobot  $w$
- Jika tidak, tampilkan pesan error (tidak diimplementasikan)

### Prediksi

- Hitung hasil prediksi dengan menggunakan produk titik antara  $X$  dan bobot  $w$
- Tambahkan bias  $b$  untuk mendapatkan hasil akhir prediksi
- Kembalikan hasil prediksi

#### 4. Hasil Evaluasi Model

From Scratch	From Scikit-Learn
<pre>Accuracy: 0.8260869565217391       precision    recall  f1-score   support       0       0.83      1.00      0.90      399      1       0.00      0.00      0.00       84   accuracy_f1_score      0.83      483  macro avg_f1_score      0.45      483  weighted avg_f1_score    0.75      483</pre>	<pre>Accuracy: 0.8302277432712215       precision    recall  f1-score   support       0       0.84      0.97      0.90      399      1       0.55      0.14      0.23       84   accuracy_f1_score      0.83      483  macro avg_f1_score      0.57      483  weighted avg_f1_score    0.79      483</pre>

Model from scratch menunjukkan akurasi mencapai 82.6%, dengan precision, recall, dan f1-score untuk kelas positif (1) semuanya sangat rendah (0.00, 0.00, dan 0.00 masing-masing), yang menunjukkan bahwa model tidak mampu mengidentifikasi kelas positif dengan baik, yang tercermin dalam nilai recall yang nol. Sementara model Scikit-Learn memiliki akurasi sedikit lebih tinggi, yaitu 83.0%. Meskipun precision untuk kelas positif masih rendah (0.55), recall dan f1-score untuk kelas positif lebih baik dibandingkan dengan model dari scratch (0.14 dan 0.23 masing-masing).

Hal ini terjadi karena model dari scratch mungkin menghadapi masalah dalam perhitungan atau parameterisasi yang mempengaruhi performa. Misalnya, pemilihan kernel, perhitungan matriks kernel, atau penyelesaian program kuadratik dapat menjadi sumber kesalahan. Model scikit-learn biasanya telah melalui proses pengujian dan optimasi yang lebih menyeluruh, menghasilkan implementasi yang lebih stabil dan efisien. Model scikit-learn juga mungkin menggunakan metode optimasi dan tuning yang lebih baik untuk menemukan hyperparameter yang optimal secara otomatis, sedangkan model dari scratch mungkin menggunakan parameter tetap tanpa penyesuaian lebih lanjut. Parameter seperti C, dapat mempengaruhi kinerja secara signifikan, dan scikit-learn mungkin lebih baik dalam menentukan nilai yang optimal.

#### 5. Improvement

Untuk meningkatkan hasil model Linear SVC, Ketidakseimbangan kelas seringkali mempengaruhi performa model dengan membuatnya cenderung memprediksi kelas mayoritas secara dominan. Oleh karena itu, penting untuk menerapkan teknik penanganan ketidakseimbangan, seperti oversampling kelas minoritas menggunakan SMOTE (Synthetic Minority Over-sampling Technique) atau undersampling kelas mayoritas. Hal tersebut dapat membantu model untuk lebih sensitif terhadap kelas minoritas dan meningkatkan kinerja pada kelas yang kurang terwakili.

Selain itu, meskipun kernel linear digunakan dalam implementasi ini, eksplorasi kernel lainnya seperti RBF (Radial Basis Function) atau polinomial bisa

sangat bermanfaat, terutama jika data memiliki pola non-linear yang tidak dapat diakomodasi oleh kernel linear. Penyesuaian parameter kernel dan kernel yang sesuai dapat membantu model dalam menangkap hubungan yang lebih kompleks dalam data, sehingga meningkatkan hasil klasifikasi. Dengan mengoptimalkan teknik penanganan ketidakseimbangan dan melakukan penyetelan kernel, performa model dapat ditingkatkan secara signifikan.