

Analisis Chapter 5 – Support Vector Machines (SVM)

Ringkasan Materi

Chapter ini membahas konsep dan implementasi **Support Vector Machines (SVM)**, salah satu algoritma klasifikasi paling kuat dan fleksibel. SVM bekerja dengan mencari hyperplane terbaik yang memisahkan data dua kelas dengan margin terbesar. Materi utama dalam bab ini meliputi:

- Linear SVM (Hard Margin & Soft Margin)
 - Pentingnya Feature Scaling
 - Polynomial Feature Mapping
 - Kernel Trick (Polynomial & RBF)
 - Optimasi hyperparameter dengan GridSearchCV
-

1. Linear SVM (Virginica Classifier)

Eksperimen pertama dilakukan pada dataset **Iris** untuk klasifikasi biner: **apakah spesies adalah Virginica atau bukan**. Model yang digunakan adalah LinearSVC dari Scikit-Learn dengan regularisasi $C=1.0$. Fitur yang digunakan adalah panjang dan lebar petal.

Hasil:

- Tanpa standardisasi, model memberikan decision boundary yang kurang akurat.
- Dengan **StandardScaler**, margin menjadi simetris dan pemisahan antar kelas menjadi lebih baik.

Perbandingan antara $C = 1$ dan $C = 100$ menunjukkan bahwa nilai C memengaruhi margin:

- C besar \rightarrow margin sempit dan model lebih ketat
 - C kecil \rightarrow margin lebih lebar dan toleran terhadap pelanggaran
-

2. Pengaruh Feature Scaling

Untuk menunjukkan pentingnya standardisasi, digunakan dataset mainan dengan perbedaan skala fitur yang besar. Hasil menunjukkan bahwa:

- Tanpa scaling: decision boundary menjadi sangat miring dan tidak optimal.
- Dengan StandardScaler: margin menjadi bersih, proporsional, dan akurat.

Kesimpulan:

SVM sangat sensitif terhadap skala fitur. Selalu lakukan standardisasi sebelum melatih model SVM.

3. Polynomial Feature Mapping (Tanpa Kernel)

Untuk menangani data non-linear, dilakukan eksperimen pada dataset bulan sabit (make_moons). Dataset ini tidak bisa dipisahkan secara linear di ruang fitur asli.

Solusi:

- Tambahkan fitur polinomial (PolynomialFeatures(degree=3))
- Gunakan LinearSVC setelah transformasi dan scaling

Hasil:

- Model linear berhasil membuat boundary non-linear yang mengikuti bentuk bulan sabit.

Keuntungan:

- Tetap menggunakan model linear
 - Tidak perlu kernel trick
-

4. SVM dengan Kernel Polynomial (SVC)

Selanjutnya digunakan SVC dengan kernel "poly" agar tidak perlu transformasi fitur secara eksplisit. Parameter degree=3, coef0=1, dan C=5.

Hasil:

- Decision boundary secara otomatis menjadi non-linear
 - Hasil sangat mirip dengan metode polynomial manual, tapi lebih efisien
-

5. SVM dengan Kernel RBF (Gaussian)

Kernel RBF adalah yang paling umum digunakan karena sangat fleksibel.

Konfigurasi:

- kernel="rbf"
- gamma=0.5 → kontrol radius pengaruh tiap titik
- C=5 → kontrol margin

Hasil:

- Boundary sangat kompleks dan mengikuti bentuk data
 - Akurasi tinggi tanpa overfitting
-

6. Optimasi Hyperparameter dengan GridSearchCV

Digunakan GridSearchCV untuk mencari kombinasi C dan gamma terbaik pada kernel RBF.

Parameter grid:

- C: [0.1, 1, 10, 100]
- gamma: [0.01, 0.1, 0.5, 1]

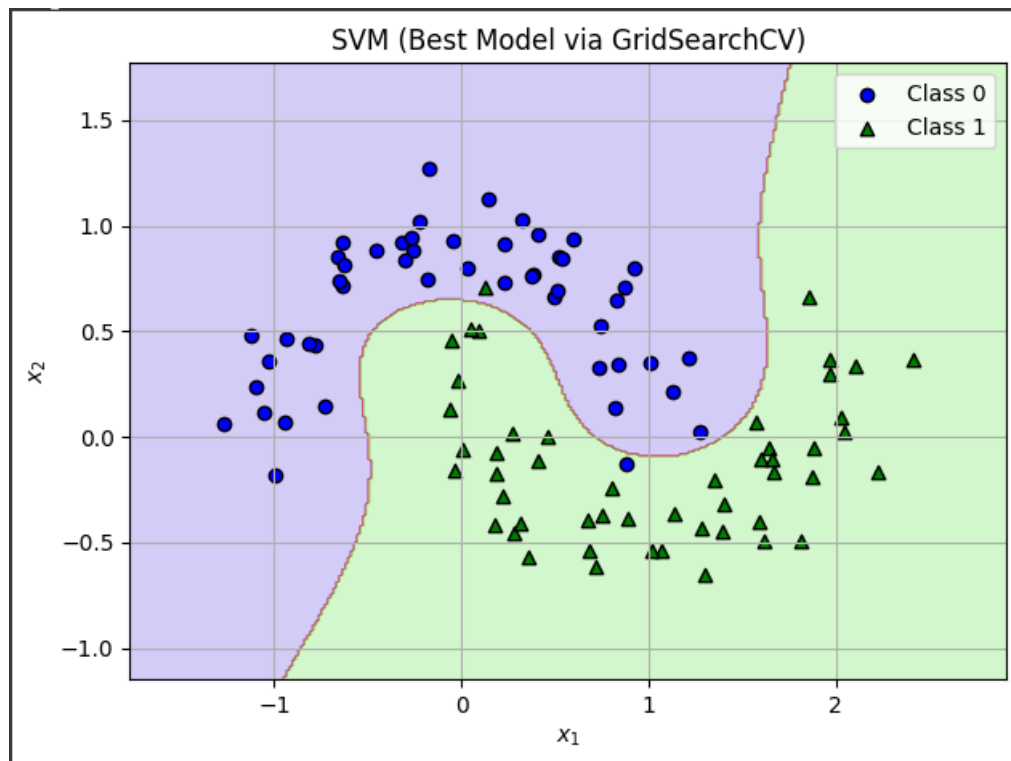
Hasil GridSearchCV:

- **Best parameters:** C = 10, gamma = 0.5
- **Best CV accuracy: 97%**

Model terbaik memberikan decision boundary yang sangat tepat membelah dua kelas dalam data `make_moons`.

Visualisasi Akhir (Model Terbaik)

Visualisasi model terbaik dari GridSearchCV menunjukkan pemisahan dua kelas secara sempurna dengan decision boundary yang halus dan presisi:



Kesimpulan Akhir

Aspek Utama	Insight
Linear SVM	Cocok untuk data yang separable secara linear
Feature Scaling	Wajib sebelum training SVM, terutama untuk LinearSVC atau SVC
Polynomial Mapping	Bisa atasi non-linear dengan transformasi fitur
Kernel Polynomial	Efisien, tanpa transformasi manual
Kernel RBF	Powerful untuk boundary kompleks, cocok untuk bentuk tidak umum
GridSearchCV	Menemukan kombinasi hyperparameter optimal secara sistematis

Bab ini menekankan bahwa SVM adalah metode yang sangat kuat untuk klasifikasi dan mampu menangani berbagai bentuk data. Kombinasi antara feature engineering, kernel trick, dan tuning hyperparameter menjadikan SVM sangat fleksibel untuk masalah klasifikasi nyata.