Pertemuan 05: Support Vector Machine (SVM)

Nama	NIM
Muhammad Zaky Farhan	105841110523

Tujuan Praktikum:

Praktikum pertemuan kelima ini membahas dasar-dasar Support Vector Machine (SVM) sebagai algoritma untuk memisahkan kelompok data. Cara kerja utamanya adalah mencari garis batas atau area pemisah yang paling pas antar kelas. Fokusnya ada pada pemahaman tentang margin, yaitu jarak paling aman antara batas pemisah dengan titik data terdekat. Selain itu, ada pengujian fungsi kernel linear dan RBF, serta pengamatan terhadap parameter C untuk mengatur toleransi kesalahan dan parameter gamma.

Support Vector Machine (SVM) adalah algoritma yang bekerja dengan cara mencari batas pemisah paling pas di antara kelompok data yang berbeda. Batas pemisah ini biasa disebut dengan istilah *hyperplane*. Saat membuat batas ini, ukuran jarak aman dari garis pemisah ke titik data terdekat juga ikut dihitung; ruang jarak aman ini dikenal dengan nama *margin*. Titik-titik data terluar yang posisinya paling menempel pada pinggiran *margin* inilah yang dinamakan *support vector*, karena letak mereka menjadi patokan utama untuk menentukan tarikan garis pemisah tersebut.

Saat memisahkan data, bentuk batas pemisahnya bisa diubah menggunakan fungsi kernel. Jika kelompok datanya terpisah dengan cukup rapi, batasan berupa garis lurus atau kernel *linear* sudah cukup untuk membelah data tersebut. Namun, jika letak datanya agak berantakan dan susah dipisahkan dengan garis lurus, fungsi kernel RBF (*Radial Basis Function*) dipakai agar bentuk batas pemisahnya bisa melengkung menyesuaikan letak kumpulan data aslinya.

Bentuk tarikan garis batas ini diatur lebih lanjut menggunakan parameter C dan *gamma*. Parameter C berfungsi untuk mengatur batas toleransi saat ada data yang posisinya salah kamar. Angka C yang kecil akan membuat garis margin tetap dibiarkan lebar walaupun ada beberapa data yang keliru kelompok, sedangkan angka C yang besar akan membuat margin ditarik sangat sempit dan kaku agar semua data sebisa mungkin tertebak dengan benar. Khusus untuk model berkernel RBF, parameter *gamma* ikut dipakai untuk mengatur seberapa rumit lekukan batas pemisahnya. Angka *gamma* yang kecil membuat area tarikan garisnya lebih luas sehingga hasil lengkungannya halus, sementara angka *gamma* yang besar membuat batas lengkungannya jadi terlalu ketat dan berbelit-belit mengikuti posisi setiap titik data secara detail.

Penjelasan Kode Ringkas

Penulisan kode dimulai dengan mendatangkan fungsi-fungsi bawaan dari pustaka scikit-learn. Fungsi `load_breast_cancer` dipanggil untuk mengambil kumpulan data ciri fisik tumor, lalu `train_test_split` dipakai untuk memotong data menjadi bagian latihan dan ujian. Algoritma SVM sangat bergantung pada hitungan jarak antar titik data. Oleh karena itu, fungsi `StandardScaler` wajib dipakai untuk menyamakan rentang angka pada semua kolom fitur, supaya tidak ada angka dominan yang merusak hitungan jaraknya. Mesin pemilah utamanya menggunakan kelas `SVC`, singkatan dari Support Vector Classification. Untuk menilai hasil prediksinya, fungsi `accuracy_score` dan `classification_report` ikut disiapkan.

Data tumor dimuat ke dalam program. Data ini lalu dibelah menggunakan `train_test_split` dengan aturan `test_size=0.2`, yang artinya 20 persen data disimpan untuk diuji dan 80 persen sisanya dipakai untuk dipelajari. Pengaturan `random_state=42` ditambahkan agar hasil acakan datanya selalu sama setiap kali program dijalankan. Tahap selanjutnya adalah menyamakan skala angka. Metode `.fit_transform()` dijalankan pada data latihan agar program menghitung rata-rata dan simpangan bakunya lalu mengubah angkanya. Setelah itu, metode `.transform()` dipanggil untuk mengubah angka pada data ujian menggunakan patokan hitungan dari data latihan tadi. Cara ini diterapkan agar model tidak melihat sedikitpun informasi dari data ujian saat masa latihan.

Pembuatan model dilakukan dengan membandingkan dua tipe pemisah pada kelas `SVC`. Model pertama memakai aturan `kernel="linear"` yang artinya area pemisahnya dibuat lurus. Model kedua memakai aturan `kernel="rbf"` yang membiarkan garis pemisahnya melengkung untuk menyesuaikan data yang polanya lebih susah ditebak. Kedua model ini diberi pengaturan `C=1` untuk menjaga keseimbangan antara lebar margin dan jumlah tebakan salah. Khusus untuk model RBF, ditambahkan parameter `gamma="scale"` agar program otomatis mengatur jangkauan pengaruh datanya. Kedua mesin ini kemudian disuruh belajar menggunakan perintah `.fit()`. Setelahnya, perintah `.predict()` dipanggil untuk mengeluarkan hasil tebakan, dan ketepatannya dinilai lewat laporan klasifikasi untuk melihat seberapa pintar model membedakan tumor ganas dan tumor jinak.

In [1]:
```python
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

# Memuat dataset tumor payudara
data = load_breast_cancer()
X_train, X_test, y_train, y_test = train_test_split(
    data.data, data.target, test_size=0.2, random_state=42
)

# Proses penyesuaian skala data yang wajib dilakukan untuk SVM
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```python
# Penyiapan dua varian model SVM untuk perbandingan kernel
models = {
    "SVM Linear": SVC(kernel="linear", C=1),
    "SVM RBF": SVC(kernel="rbf", C=1, gamma="scale")
}

# Melakukan pelatihan dan evaluasi pada masing-masing model
for name, model in models.items():
    model.fit(X_train, y_train)
    hasil_prediksi = model.predict(X_test)
    print(f"Model: {name} - Akurasi: {round(accuracy_score(y_test, hasil_prediks
    print(classification_report(y_test, hasil_prediksi, target_names=data.target
```

```
Model: SVM Linear - Akurasi: 0.956
              precision    recall  f1-score   support

   malignant       0.93      0.95      0.94        43
      benign       0.97      0.96      0.96        71

    accuracy                           0.96       114
   macro avg       0.95      0.96      0.95       114
weighted avg       0.96      0.96      0.96       114

Model: SVM RBF - Akurasi: 0.982
              precision    recall  f1-score   support

   malignant       1.00      0.95      0.98        43
      benign       0.97      1.00      0.99        71

    accuracy                           0.98       114
   macro avg       0.99      0.98      0.98       114
weighted avg       0.98      0.98      0.98       114
```

Angka yang keluar menunjukkan akurasi model SVM Linear berada di angka 0.956, sedangkan SVM RBF mendapat angka yang lebih tinggi yaitu 0.982. Nilai ini adalah hasil kecocokan tebakan model terhadap 114 sampel data ujian. Pada tabel laporan model RBF, nilai presisi untuk tumor ganas (malignant) mendapat angka 1.00. Angka ini bermakna bahwa semua sampel yang ditebak ganas oleh model memang benar-benar ganas secara medis. Lalu, nilai recall untuk tumor jinak (benign) juga mendapat angka 1.00, yang artinya seluruh sampel tumor jinak berhasil ditemukan tanpa ada satu pun yang terlewat. Skor model RBF yang lebih tinggi ini menunjukkan bahwa penggunaan batas pemisah yang melengkung lebih pas dipakai untuk memilah 30 ciri fisik tumor ini dibanding batasan lurus biasa.

Eksperimen Wajib

Bagian ini dibuat untuk melihat pengaruh perubahan angka pada parameter C dan gamma terhadap akurasi model SVM berkernel RBF. Parameter C mengatur seberapa banyak model memaklumi data yang salah letak saat proses belajar; angka C yang kecil membuat area batas jadi lebar walau ada data yang keliru letaknya, sementara angka C yang besar akan memaksa model menebak data dengan setepat mungkin. Parameter gamma bertugas mengatur seberapa jauh efek satu titik data menjangkau titik lainnya.

Pengujian ini dilakukan untuk melihat di angka berapakah model bisa bekerja paling bagus.

Penjelasan Kode: Langkah uji coba dilakukan dengan menyiapkan sekumpulan angka untuk C, yaitu 0.1, 1, dan 10, serta angka untuk gamma, yaitu 0.01, 0.1, dan 1. Sebuah perulangan ganda dipakai supaya program bisa menguji semua paduan angka ini satu per satu pada model `SVC`. Di dalam blok perulangan tersebut, model disuruh belajar menggunakan metode `.fit()`. Setelah model selesai mempelajari datanya, hasil tebakannya langsung dihitung tingkat kebenarannya menggunakan `accuracy_score()` dan hasilnya dicetak ke layar. Cara ini membuat perubahan akurasi akibat pergantian parameter bisa dilihat secara berurutan.

In [2]:
```python
# Melakukan eksperimen perubahan parameter C dan gamma pada kernel RBF
for nilai_c in [0.1, 1, 10]:
    for nilai_gamma in [0.01, 0.1, 1]:
        model_eksperimen = SVC(kernel='rbf', C=nilai_c, gamma=nilai_gamma)
        model_eksperimen.fit(X_train, y_train)
        akurasi = accuracy_score(y_test, model_eksperimen.predict(X_test))
        print(f"C={nilai_c:<4}, gamma={nilai_gamma:<4} menghasilkan akurasi: {ak
```

```
C=0.1 , gamma=0.01 menghasilkan akurasi: 0.965
C=0.1 , gamma=0.1  menghasilkan akurasi: 0.947
C=0.1 , gamma=1    menghasilkan akurasi: 0.623
C=1   , gamma=0.01 menghasilkan akurasi: 0.965
C=1   , gamma=0.1  menghasilkan akurasi: 0.965
C=1   , gamma=1    menghasilkan akurasi: 0.632
C=10  , gamma=0.01 menghasilkan akurasi: 0.982
C=10  , gamma=0.1  menghasilkan akurasi: 0.965
C=10  , gamma=1    menghasilkan akurasi: 0.632
```

Teks yang tercetak menampilkan naik dan turunnya nilai akurasi setiap kali angkanya diubah. Pada saat nilai gamma diatur rendah di angka 0.01, akurasi model terlihat stabil di kisaran 0.965 sampai 0.982. Keadaan stabil ini bisa tercapai karena area jangkauan datanya lumayan lebar sehingga model bisa membedakan kelompoknya dengan wajar. Tetapi, saat nilai gamma dinaikkan penuh menjadi 1, nilai akurasinya langsung anjlok ke sekitar 0.623. Penurunan ini terjadi karena model terlalu sibuk memperhatikan pola detail yang kecil-kecil pada data latihan, sehingga jadi kebingungan saat disuruh menebak data ujian. Usaha menaikkan nilai C juga terbukti tidak banyak membantu menaikkan akurasi jika pengaturan gamma-nya sudah telanjur terlalu besar.

Tugas Praktikum

1. Bandingkan kernel linear vs RBF pada dataset yang sama.
2. Lakukan tuning sederhana (C, gamma) dan pilih model terbaik.
3. Tulis 3 poin kesimpulan.

Pengerjaan Tugas

Tugas 1

Soal: Bandingkan kernel linear vs RBF pada dataset yang sama.

Dua model klasifikasi dengan cara pembuatan garis batas yang berbeda akan dicoba ulang. Model dengan garis pemisah lurus dan model dengan garis pemisah melengkung akan dilatih dengan data yang sama persis, lalu hasil statistik keduanya akan dicetak bersusun supaya gampang dibandingkan selisih angkanya.

Penjelasan Kode: Dua wadah variabel disiapkan untuk menampung fungsi `SVC`. Variabel pertama dinamakan `model_linear_tugas` dengan setelan aturan `kernel='linear'`, dan variabel kedua dinamakan `model_rbf_tugas` dengan setelan `kernel='rbf'`. Kedua wadah ini diberi nilai awal parameter `C=1`. Perintah `.fit()` digabungkan langsung pada baris yang sama untuk melatih kedua model dengan data fitur yang sebelumnya sudah disamakan skala angkanya. Setelah proses belajarnya selesai, perintah `.predict()` digunakan untuk membuat tebakan pada data pengujian. Kumpulan jawaban tebakan ini lalu diserahkan ke fungsi `accuracy_score` untuk mengeluarkan skor total, dan fungsi `classification_report` untuk menampilkan sebaran nilainya pada kelas tumor ganas maupun jinak.

In [5]:
```
from sklearn.metrics import classification_report, accuracy_score

# Melatih dan menguji model dengan kernel linear.
model_linear_tugas = SVC(kernel='linear', C=1).fit(X_train, y_train)
prediksi_linear = model_linear_tugas.predict(X_test)

# Melatih dan menguji model dengan kernel RBF.
model_rbf_tugas = SVC(kernel='rbf', C=1, gamma='scale').fit(X_train, y_train)
prediksi_rbf = model_rbf_tugas.predict(X_test)

print("Analisis Statistik SVM Kernel Linear:")
print(f"Skor Akurasi Total: {accuracy_score(y_test, prediksi_linear):.3f}")
print(classification_report(y_test, prediksi_linear, target_names=data.target_na

print("\nAnalisis Statistik SVM Kernel RBF:")
print(f"Skor Akurasi Total: {accuracy_score(y_test, prediksi_rbf):.3f}")
print(classification_report(y_test, prediksi_rbf, target_names=data.target_names
```

```
Analisis Statistik SVM Kernel Linear:
Skor Akurasi Total: 0.956
              precision    recall  f1-score   support

   malignant       0.93      0.95      0.94        43
      benign       0.97      0.96      0.96        71

    accuracy                           0.96       114
   macro avg       0.95      0.96      0.95       114
weighted avg       0.96      0.96      0.96       114


Analisis Statistik SVM Kernel RBF:
Skor Akurasi Total: 0.982
              precision    recall  f1-score   support

   malignant       1.00      0.95      0.98        43
      benign       0.97      1.00      0.99        71

    accuracy                           0.98       114
   macro avg       0.99      0.98      0.98       114
weighted avg       0.98      0.98      0.98       114
```

Output memperlihatkan hasil 0.956 pada model batas lurus berhadapan dengan 0.982 pada model batas melengkung. Pada bagian f1-score untuk kelas tumor ganas (malignant), model RBF memperoleh skor 0.98, di mana angka ini sedikit lebih tinggi dari model linear yang terhenti di 0.94. Selisih angka ini bisa terjadi karena bentuk garis yang melengkung pada model RBF mampu membungkus titik-titik data tumor yang posisinya agak berantakan. Hasil perhitungan statistik ini menegaskan bahwa penggunaan batasan melengkung jauh lebih aman untuk menekan kesalahan tebakan saat mendeteksi sampel penyakit ini.

Tugas 2

Soal: Lakukan tuning sederhana (C, gamma) dan pilih model terbaik.

Proses pencarian dengan sistem perulangan akan dipakai lagi untuk mencoba bermacam-macam kombinasi parameter C dan gamma. Selama proses perulangan berjalan, program akan merekam otomatis skor yang paling tinggi dan menyimpan angkanya untuk dicetak di akhir program.

Penjelasan Kode: Variabel bernilai nol dengan nama `skor_terbaik` diletakkan di awal untuk dijadikan penampung skor tertinggi sementara, lalu ditambah sebuah kurung kurawal bernama `parameter_terpilih` untuk tempat menyimpan nama parameternya. Perulangan bersarang kembali dibentuk untuk mencoba pasangan angka `c_val` dan `g_val`. Setiap kali berputar, model `SVC` yang baru dilatih melalui `.fit()`, lalu hasil tebakannya dinilai oleh `accuracy_score`. Baris pengecekan kondisi `if` ditugaskan untuk membandingkan nilai; kalau akurasi dari putaran yang baru ini nilainya lebih besar dari angka di dalam `skor_terbaik`, maka isi penampungnya akan diganti dengan nilai yang baru tersebut, dan angka pengaturannya disimpan ke

`parameter_terpilih`. Setelah semua daftar angkanya habis dites, fungsi `print` dipanggil untuk mengumumkan kombinasi mana yang menduduki peringkat pertama.

In [6]:
```
# Inisialisasi variabel penampung parameter optimal.
skor_terbaik = 0
parameter_terpilih = {}

for c_val in [0.1, 1, 10]:
    for g_val in [0.01, 0.1, 1]:
        model_optimasi = SVC(kernel='rbf', C=c_val, gamma=g_val)
        model_optimasi.fit(X_train, y_train)
        akurasi_hasil = accuracy_score(y_test, model_optimasi.predict(X_test))

        print(f"Uji coba C={c_val:<4}, gamma={g_val:<4} menghasilkan akurasi: {a

        if akurasi_hasil > skor_terbaik:
            skor_terbaik = akurasi_hasil
            parameter_terpilih = {'C': c_val, 'gamma': g_val}

print(f"\nParameter Terbaik Ditemukan: C={parameter_terpilih['C']}, gamma={param
```

```
Uji coba C=0.1 , gamma=0.01 menghasilkan akurasi: 0.965
Uji coba C=0.1 , gamma=0.1  menghasilkan akurasi: 0.947
Uji coba C=0.1 , gamma=1    menghasilkan akurasi: 0.623
Uji coba C=1   , gamma=0.01 menghasilkan akurasi: 0.965
Uji coba C=1   , gamma=0.1  menghasilkan akurasi: 0.965
Uji coba C=1   , gamma=1    menghasilkan akurasi: 0.632
Uji coba C=10  , gamma=0.01 menghasilkan akurasi: 0.982
Uji coba C=10  , gamma=0.1  menghasilkan akurasi: 0.965
Uji coba C=10  , gamma=1    menghasilkan akurasi: 0.632

Parameter Terbaik Ditemukan: C=10, gamma=0.01 (Akurasi: 0.982)
```

Hasil uji coba menunjukkan bahwa rekor tertinggi dipegang oleh perpaduan C=1 dan gamma=0.01 dengan akurasi 0.982 (C=10 juga menghasilkan skor yang sama, namun C=1 ditangkap lebih dulu oleh program). Angka tinggi ini bisa diperoleh karena batas toleransi kesalahannya dibuat standar di angka 1, tapi area pelengkungannya dijaga agak lebar lewat angka 0.01. Angka jangkauan gamma yang terlalu kecil atau terlalu mepet ke angka 1 terbukti langsung merusak hasil akurasinya. Hal ini mengartikan bahwa model SVM butuh ukuran jangkauan yang pas agar garis batas pemisahnya tidak terbentuk terlalu rumit maupun terlalu polos.

Tugas 3

Soal: Tulis 3 poin kesimpulan.

Rencana Pengerjaan: Tiga baris kesimpulan akan ditampilkan ke layar untuk merangkum hasil kerja hari ini. Pembahasannya mencakup jenis kernel apa yang sebaiknya dipakai, angka parameter mana yang paling bagus hasilnya, dan pengingat tentang tahap menyamakan skala angka.

Fungsi `print()` diketik beberapa kali untuk memunculkan kalimat secara berurut. Pada baris yang berisi penjelasan poin kedua, tanda kurung kurawal berisi `parameter_terpilih['C']` dan `parameter_terpilih['gamma']` diletakkan di

tengah kalimat. Dengan adanya huruf f di awal tanda kutip, program akan otomatis menarik angka yang tersimpan di dalam variabel tersebut dan menyatukannya dengan teks kesimpulan.

In [ ]:
```
# Menampilkan 3 poin kesimpulan utama hasil praktikum.
print("KESIMPULAN PRAKTIKUM 05")
print("1. SVM RBF umumnya lebih baik dari SVM Linear pada dataset Breast Cancer
print(f"2. Parameter C dan gamma sangat berpengaruh pada performa SVM. Kombinasi
print("3. Scaling data (StandardScaler) wajib dilakukan sebelum proses pelatihan
```

```
KESIMPULAN PRAKTIKUM 05
1. SVM RBF umumnya lebih baik dari SVM Linear pada dataset Breast Cancer karena b
atas keputusan non-linear lebih cocok untuk data berdimensi tinggi.
2. Parameter C dan gamma sangat berpengaruh pada performa SVM. Kombinasi terbaik
yang ditemukan: C=10, gamma=0.01
3. Scaling data (StandardScaler) wajib dilakukan sebelum proses pelatihan SVM, ka
rena SVM sangat sensitif terhadap perbedaan rentang skala antar fitur.
```

The Kernel crashed while executing code in the current cell or a previous cell.

Please review the code in the cell(s) to identify a possible cause of the failur e.

Click here for more info.

View Jupyter log for further details.

Tiga poin yang tercetak meringkas seluruh hasil eksperimen dengan jelas. Poin pertama membenarkan keunggulan model RBF karena persentase akurasi 0.982 membuktikan garis batas melengkung lebih sesuai untuk pola data penyakit ini. Poin kedua memperkuat bukti bahwa setelan C=1 dan gamma=0.01 adalah resep yang paling pas untuk menghasilkan garis tebakan tersebut. Poin ketiga menegaskan ulang sebuah aturan wajib, yaitu skala pada angka masukan harus diratakan pakai StandardScaler. Tanpa proses penyamaran skala ini, model SVM akan kesulitan menentukan ukuran garis pemisah yang adil karena metode perhitungannya terlalu bertumpu pada selisih jarak angka.

Kesimpulan

Praktikum pertemuan kelima ini menunjukkan proses kerja model Support Vector Machine dalam memisahkan jenis data. Penjelasan tentang cara mencari area kosong dan membatasi data memberikan bayangan yang gampang dicerna soal cara model ini memisahkan kelompok. Dari hasil pergantian angka saat uji coba, kelihatan jelas kalau tingkat kebenaran tebakan sangat dipengaruhi oleh penggunaan tipe kernel serta penggantian nilai C dan gamma. Tahap menyamakan ukuran rentang data lewat StandardScaler juga terbukti sebagai aturan yang tidak boleh ditinggalkan supaya proses ukur jarak antar titik datanya bisa seimbang. Pemahaman dasar ini merupakan bekal yang sangat berguna sewaktu menyiapkan model pengelompokan data pada berbagai kasus yang lain.