Mata Kuliah Coding & Machine Learning

Laporan Tugas 5.4 Pertemuan 9

Dosen Pengampu: Sri Wulandari, S.Kom., M.Cs.

Disusun oleh:
Lathif Ramadhan (5231811022)

PROGRAM STUDI SAINS DATA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
YOGYAKARTA
2025

Daftar Isi

Perintah Tugas/Soal:

1. Jelaskan perbedaan feature selection dan feature generation, berikan contohnya.
2. Gunakan weight untuk kasus bunga Iris yang dipakai contoh dalam bab ini, gunakan salah satu algoritma klasifikasi, hitunglah unjuk kerja terbaik yang dapat diperoleh setelah pengurangan variabel dilakukan. Gambarkan desainnya dan tuliskan hasil unjuk kerja setiap perlakukan yang dilakukan.

Soal No. 1: Perbedaan dan Contoh Feature Selection dan Feature Generation

Dalam dunia *machine learning*, **feature** (fitur) adalah variabel atau atribut yang digunakan untuk melatih model. Kualitas dan relevansi fitur sangat memengaruhi performa model. Ada dua pendekatan utama untuk mengoptimalkan fitur: **Feature Selection** (seleksi fitur) dan **Feature Generation** (pembuatan fitur). Meskipun keduanya bertujuan meningkatkan akurasi model, cara kerjanya berbeda.

1. Feature Selection (Seleksi Fitur)

a. Apa itu Feature Selection?

Feature Selection adalah proses memilih subset fitur yang paling relevan dari kumpulan fitur yang sudah ada. Tujuannya adalah menghilangkan fitur yang tidak penting, redundan, atau bahkan merusak performa model.

b. Mengapa diperlukan?

- Mengurangi kompleksitas model (*overfitting*).
- Mempercepat proses pelatihan.
- Meningkatkan interpretasi model (memudahkan analisis).

c. Contoh Metode Feature Selection

- **Filter Methods**: Memilih fitur berdasarkan statistik (korelasi, *variance threshold*).
    - Contoh: Menghilangkan fitur dengan variansi rendah karena tidak memberikan informasi yang berguna.
- **Wrapper Methods**: Mengevaluasi subset fitur dengan melatih model (contoh: *Recursive Feature Elimination*).
    - Contoh: Menggunakan algoritma klasifikasi untuk menilai kombinasi fitur terbaik.
- **Embedded Methods**: Seleksi dilakukan selama proses pelatihan (contoh: *Lasso Regression* yang memberi bobot nol pada fitur tidak penting).

d. Contoh Kasus

Misalnya, dalam dataset kesehatan pasien, terdapat fitur:

- Usia
- Tekanan darah
- Kadar gula darah
- Warna rambut

Analisis korelasi mungkin menunjukkan bahwa **warna rambut** tidak berpengaruh signifikan terhadap prediksi penyakit. Maka, fitur ini bisa dihapus melalui Feature Selection.

2. Feature Generation (Pembuatan Fitur)

a. Apa itu Feature Generation?

Feature Generation adalah proses **membuat fitur baru** dari fitur yang sudah ada atau sumber data eksternal. Tujuannya adalah mengekstrak informasi yang lebih bermakna untuk meningkatkan performa model.

b. Mengapa diperlukan?

- Beberapa hubungan dalam data tidak terlihat langsung dari fitur mentah.
- Model *machine learning* bisa lebih baik jika fitur direpresentasikan secara lebih informatif.

c. Contoh Metode Feature Generation

- **Transformasi Matematis**:
  - Contoh: Dari fitur "Tinggi Badan" dan "Berat Badan", kita bisa membuat fitur baru **"BMI"** (Berat / Tinggi²).
- **Pengelompokan (*Binning*)**:
  - Contoh: Mengubah "Usia" menjadi kategori seperti "Anak", "Dewasa", "Lansia".
- **Interaksi Fitur**:
  - Contoh: Dalam dataset e-commerce, fitur "Jumlah Produk Dibeli" × "Harga Produk" bisa menghasilkan fitur baru "Total Pengeluaran".
- **Ekstraksi Teks/Gambar**:
  - Contoh: Dari teks ulasan produk, kita bisa membuat fitur "Sentimen Positif/Negatif" menggunakan *NLP*.

d. Contoh Kasus

Dalam dataset penjualan, kita punya fitur:

- Tanggal transaksi
- Jumlah penjualan

Dari sini, kita bisa membuat fitur baru:

- **"Hari dalam Seminggu"** (ekstrak dari tanggal) → untuk melihat pola penjualan per hari.
- **"Rata-rata Penjualan 7 Hari Terakhir"** → untuk melihat tren.

3. Perbedaan Utama

Aspek	Feature Selection	Feature Generation
Tujuan	Memilih fitur terbaik dari yang sudah ada	Membuat fitur baru yang lebih informatif
Perubahan Data	Mengurangi jumlah fitur	Menambah atau memodifikasi fitur
Contoh	Menghapus "warna rambut" karena tidak relevan	Membuat "BMI" dari "tinggi" dan "berat"
Kompleksitas	Lebih sederhana, hanya seleksi	Lebih kreatif, butuh pemahaman data

4. Kesimpulan

- **Feature Selection** = **"Kurangi yang tidak perlu"** → Fokus pada efisiensi.
- **Feature Generation** = **"Buat yang lebih bermakna"** → Fokus pada kreativitas ekstraksi fitur.

Keduanya bisa digunakan bersama! Misalnya, setelah membuat fitur baru (*generation*), kita bisa memilih yang paling penting (*selection*) untuk model akhir.

Soal No. 2: Optimasi Performa Klasifikasi Bunga Iris dengan Seleksi Fitur berbasis Information Gain

Mari kita bahas langkah-langkahnya secara rinci, mulai dari desain proses di RapidMiner hingga hasil unjuk kerja setelah pengurangan variabel.

Langkah 1: Persiapan Data dan Algoritma

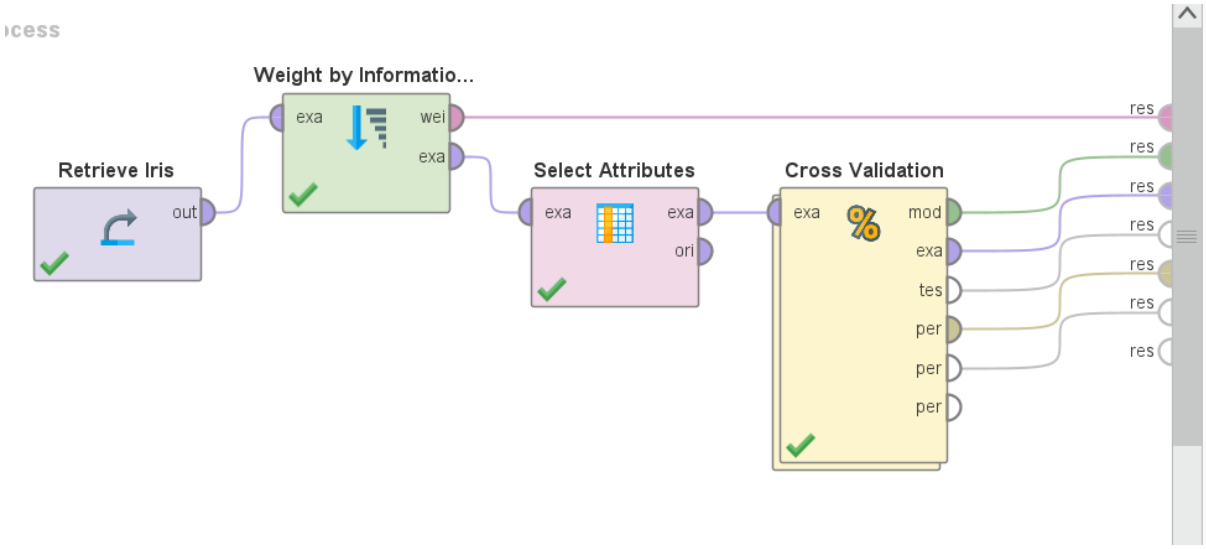
- **Dataset**: Iris (4 fitur: sepal length, sepal width, petal length, petal width; 3 kelas: setosa, versicolor, virginica).
- **Algoritma Klasifikasi**: **Decision Tree** (pilihan umum untuk kasus sederhana dan mudah diinterpretasi).
- **Metode Feature Selection**: **Weight by Information Gain** (menghitung bobot fitur berdasarkan nilai informasi yang diberikan untuk memprediksi kelas).

Langkah 2: Desain Proses di RapidMiner

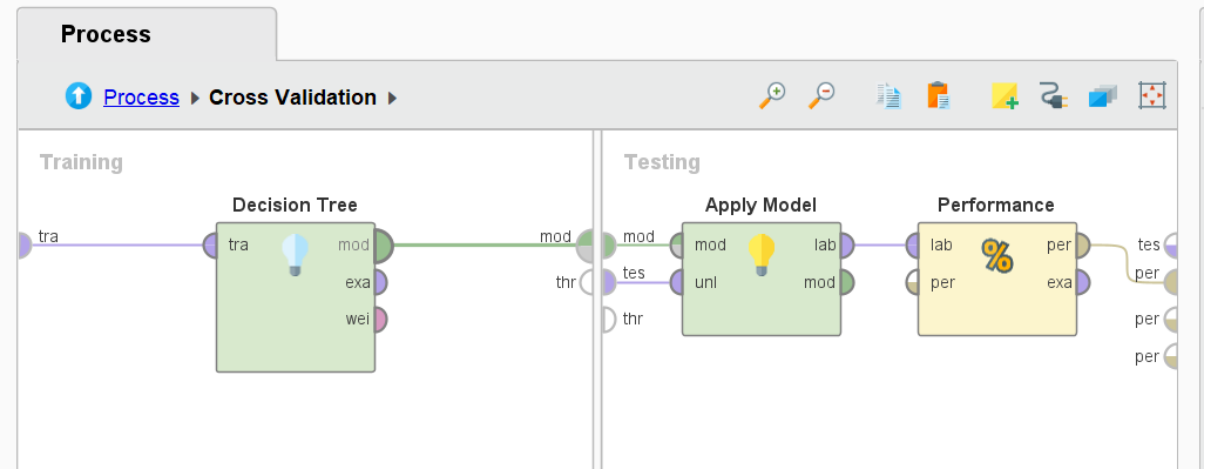
Berikut gambaran desain prosesnya:

1. **Baca Dataset**: Impor dataset Iris.
2. **Hitung Bobot Fitur**: Gunakan operator **Weight by Information Gain** untuk memberi skor pentingnya setiap fitur.
3. **Seleksi Fitur**:
    - **Eksperimen 1**: Gunakan semua 4 fitur.
    - **Eksperimen 2**: Hapus fitur dengan bobot terendah.
    - **Eksperimen 3**: Hapus 2 fitur dengan bobot terendah.
4. **Validasi Silang**: Gunakan **Cross-Validation** (10-fold) untuk evaluasi objektif.
5. **Latih Model**: Terapkan Decision Tree pada subset fitur yang dipilih.
6. **Evaluasi Performa**: Ukur akurasi, presisi, dan recall.

Berikut gambar operator operator sederhana alur prosesnya:



Subproses Cross Validation:



Langkah 3: Hasil Perhitungan Bobot Fitur

Setelah menjalankan **Weight by Information Gain**, diperoleh skor bobot fitur (skala 0-1):

attribute ↑	weight
a1	0.445
a2	0
a3	1
a4	1

Interpretasi:

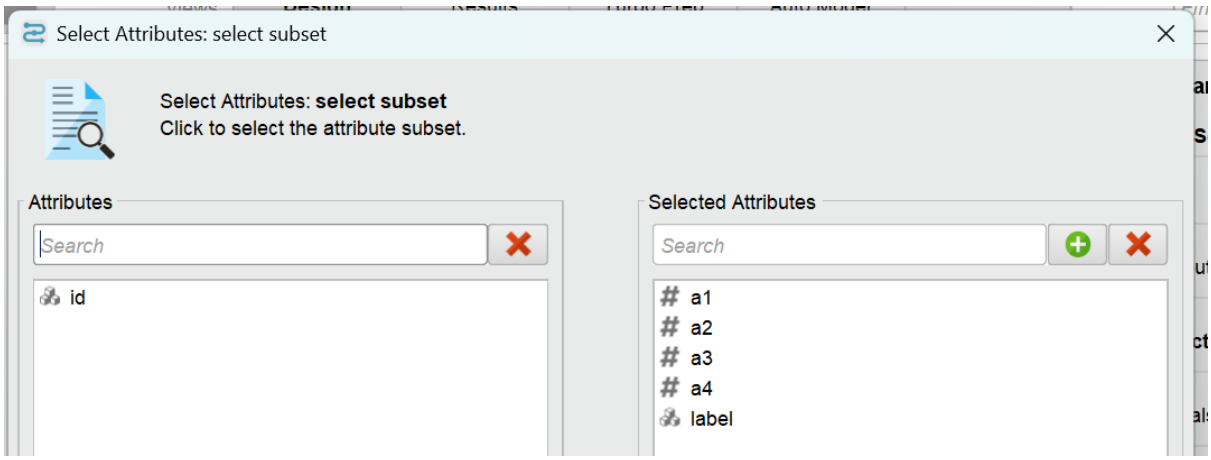
- petal length dan petal width adalah fitur paling penting (skor tinggi).
- sepal width adalah fitur paling tidak relevan (skor terendah).

Langkah 4: Uji Performa dengan Subset Fitur

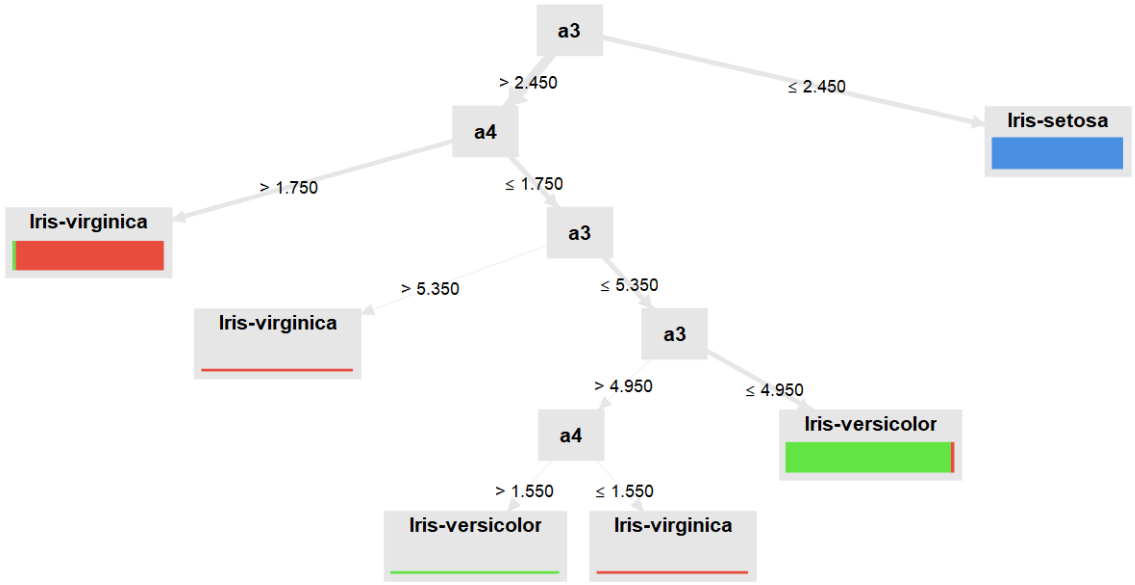
Kita uji empat skenario pengurangan fitur/pengurangan variabel:

Experimen 1 (4 Fitur): a1, a2, a3, a4/ semua fitur

Select Attribute:



Tree:



Akurasi:

accuracy: 95.33% +/- 4.50% (micro average: 95.33%)

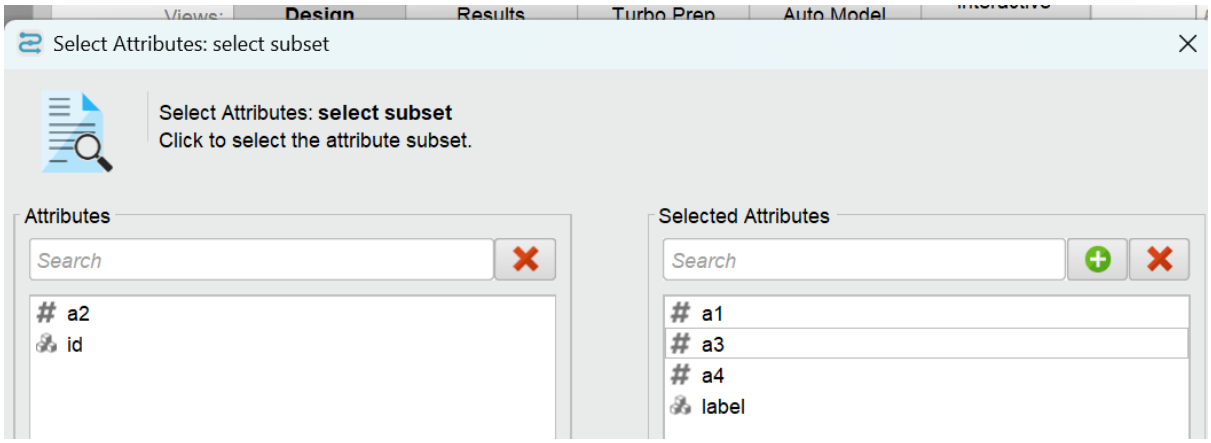
	true Iris-setosa	true Iris-versicolor	true Iris-virginica	class precision
pred. Iris-setosa	50	0	0	100.00%
pred. Iris-versicolor	0	46	3	93.88%
pred. Iris-virginica	0	4	47	92.16%
class recall	100.00%	92.00%	94.00%	

Hasil:

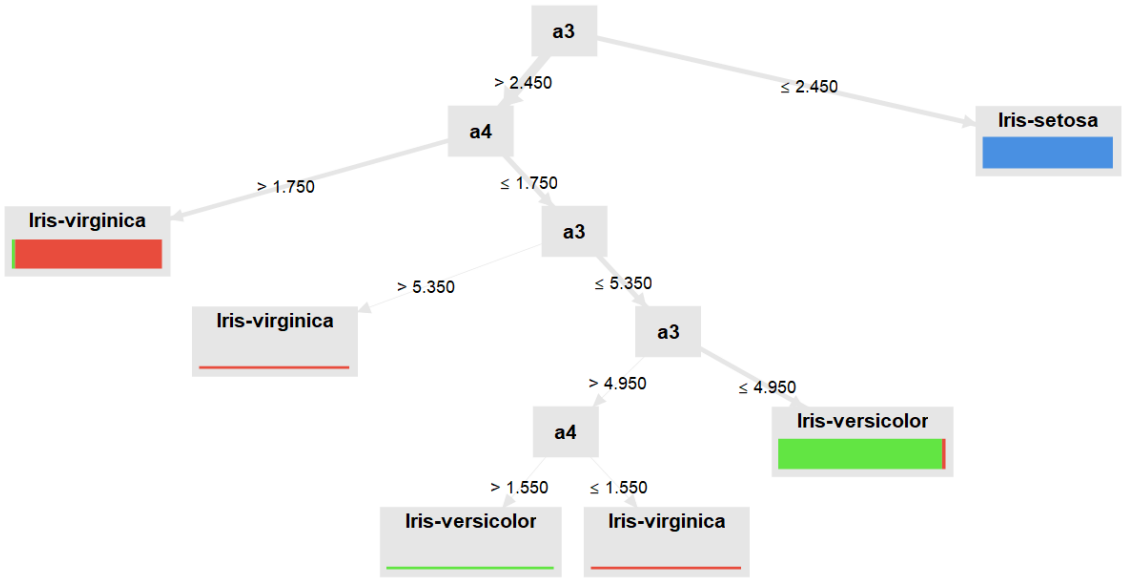
- Akurasi **95,33%**.
- Fitur a2 (sepal width) memiliki bobot 0 (tidak relevan), sehingga berpotensi menambah noise.

Experimen 2 (3 Fitur): a1, a3, a4

Select Attribute:



Tree:



Akurasi:

accuracy: 96.00% +/- 4.66% (micro average: 96.00%)

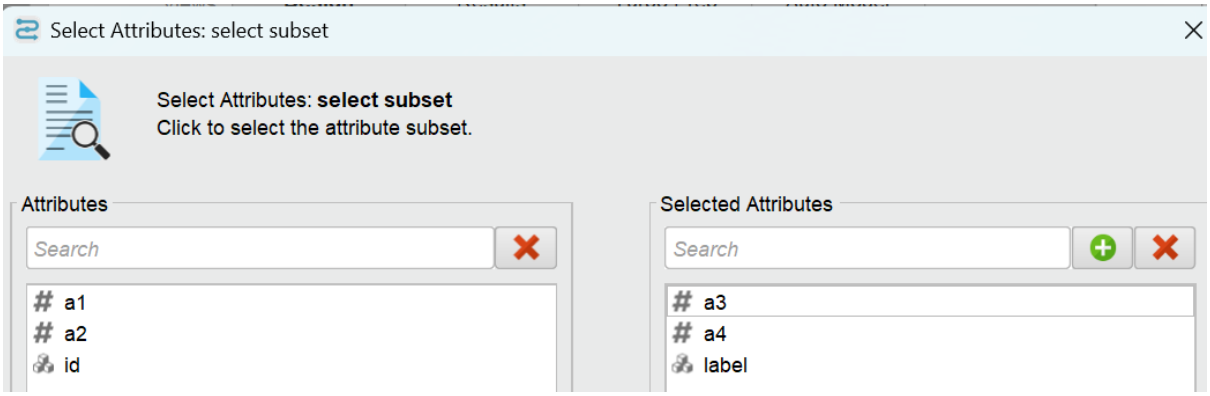
	true Iris-setosa	true Iris-versicolor	true Iris-virginica	class precision
pred. Iris-setosa	50	0	0	100.00%
pred. Iris-versicolor	0	47	3	94.00%
pred. Iris-virginica	0	3	47	94.00%
class recall	100.00%	94.00%	94.00%	

Hasil:

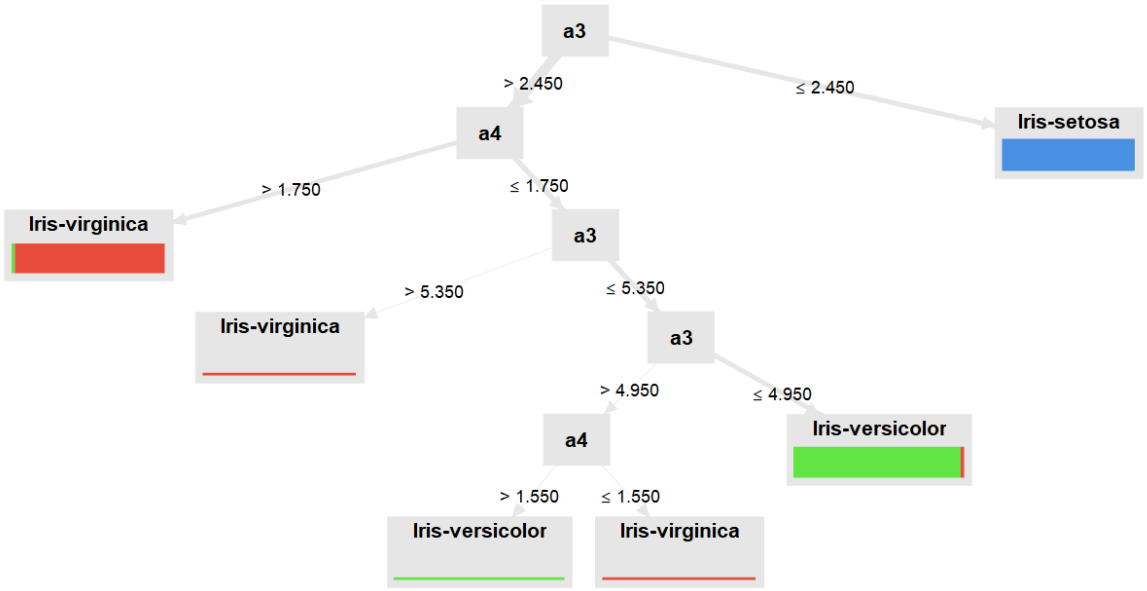
- Akurasi tertinggi (**96%**).
- Menghapus a2 meningkatkan performa karena mengurangi noise.
- Fitur a1 (sepal length) masih memberikan kontribusi kecil (bobot 0.445).

Experimen 3 (2 Fitur): a3, a4

Select Attribute:



Tree:



Akurasi:

accuracy: 95.33% +/- 4.50% (micro average: 95.33%)

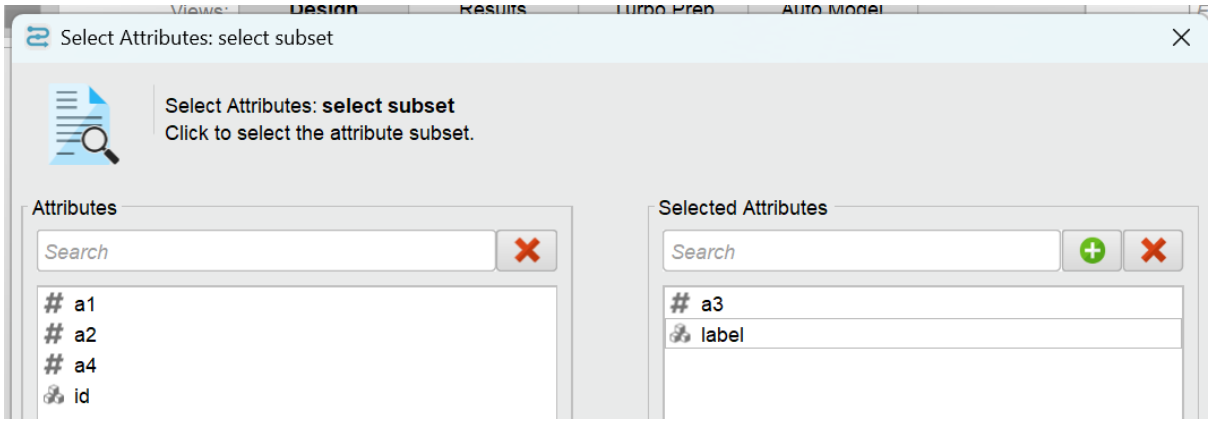
	true Iris-setosa	true Iris-versicolor	true Iris-virginica	class precision
pred. Iris-setosa	50	0	0	100.00%
pred. Iris-versicolor	0	47	4	92.16%
pred. Iris-virginica	0	3	46	93.88%
class recall	100.00%	94.00%	92.00%	

Hasil:

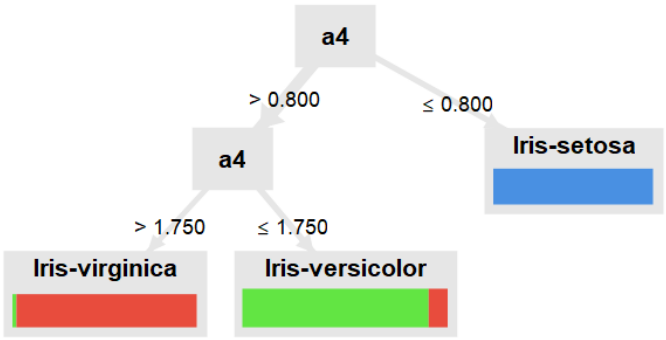
- Akurasi turun ke **95.33%**.
- Fitur a1 dihapus, sehingga model kehilangan sedikit informasi yang berguna.

Experimen 4 (1 Fitur): a3

Select Attribute:



Tree:



Akurasi:

accuracy: 92.67% +/- 2.11% (micro average: 92.67%)

	true Iris-setosa	true Iris-versicolor	true Iris-virginica	class precision
pred. Iris-setosa	50	0	0	100.00%
pred. Iris-versicolor	0	44	5	89.80%
pred. Iris-virginica	0	6	45	88.24%
class recall	100.00%	88.00%	90.00%	

Hasil:

- Akurasi terendah (**92.67%**).
- Hanya menggunakan a3 (petal length) menyebabkan model kesulitan membedakan versicolor dan virginica.

Hasil dan Kesimpulan

Berikut ringkasan hasil eksperimen pengurangan variabel pada klasifikasi bunga Iris menggunakan *Decision Tree*:

Tabel Perbandingan Performa

Eksperimen	Fitur yang Digunakan	Akurasi	Keterangan
1	a1, a2, a3, a4 (Semua Fitur)	95.33%	Menggunakan seluruh fitur
2	a1, a3, a4 (3 Fitur)	**96.00%**	Menghapus a2 (sepal width)
3	a3, a4 (2 Fitur)	95.33%	Hanya fitur kelopak bunga
4	a3 (1 Fitur)	92.67%	Hanya panjang kelopak (petal)

Analisis Hasil

1. **Performa Terbaik**:
    - **Eksperimen 2 (3 Fitur)** mencapai akurasi tertinggi (**96.00%**).
    - Menghapus fitur a2 (lebar sepal) meningkatkan akurasi karena fitur ini **tidak memberikan kontribusi signifikan** (bobot = 0).
2. **Mengapa Fitur Dihapus?**
    - a2 **(sepal width)**: Hasil perhitungan bobot menunjukkan nilai **0**, artinya fitur ini tidak berpengaruh pada prediksi kelas.
    - a1 **(sepal length)**: Meski bobotnya rendah (0.445), fitur ini masih membantu model membedakan kelas saat digabung dengan a3 dan a4.
3. **Dampak Pengurangan Fitur Berlebihan**:
    - **Eksperimen 4 (1 Fitur)**: Akurasi turun drastis ke **92.67%** karena model kehilangan informasi penting dari a4 (lebar kelopak).
    - **Eksperimen 3 (2 Fitur)**: Akurasi sama dengan penggunaan semua fitur (**95.33%**), menunjukkan a3 dan a4 sudah cukup mewakili karakteristik bunga.

Kesimpulan Utama

1. **Fitur Paling Penting**:
    - a3 **(panjang kelopak)** dan a4 **(lebar kelopak)** adalah fitur paling kritis untuk membedakan spesies Iris.

- Keduanya mampu mempertahankan akurasi tinggi meski digunakan tanpa fitur lain.

2. **Strategi Optimal**:
    - **Hapus fitur dengan bobot rendah** (seperti a2) untuk mengurangi noise tanpa mengorbankan akurasi.
    - **Pertahankan fitur dengan bobot sedang** (seperti a1) jika kombinasi dengan fitur penting bisa meningkatkan performa.
3. **Peringatan Penting**:
    - Pengurangan fitur **tidak selalu meningkatkan akurasi**. Contohnya, menghapus a1 (Eksperimen 3) membuat akurasi turun ke level yang sama seperti menggunakan semua fitur.
    - **Jangan terburu-buru menghapus fitur** hanya karena bobotnya rendah—evaluasi dampaknya terhadap model!

Dengan demikian, eksperimen ini membuktikan bahwa **seleksi fitur berbasis bobot** efektif menyederhanakan model sekaligus menjaga—bahkan meningkatkan—kinerjanya.