

Decission Tree

https://github.com/as-budi/Embedded_AI.git

1. Pengertian Decision Tree

- Decision Tree adalah salah satu algoritma dalam machine learning yang digunakan untuk tugas klasifikasi dan regresi.
- Algoritma ini bekerja seperti diagram pohon keputusan yang terdiri dari simpul (node) dan cabang (branch).
- Decision Tree sangat populer karena mudah dipahami dan diinterpretasikan.

2. Struktur Decision Tree

- **Root Node:** Simpul awal yang mewakili seluruh dataset dan menentukan fitur pertama untuk dipilah.
- **Internal Nodes:** Simpul di tengah yang merepresentasikan keputusan berdasarkan fitur tertentu.
- **Leaf Nodes:** Simpul akhir yang mewakili label kelas (untuk klasifikasi) atau nilai (untuk regresi).
- **Branches:** Jalur yang menghubungkan node satu ke node lainnya berdasarkan kondisi tertentu.

3. Cara Kerja Decision Tree

1. Memilih Fitur Utama

- Decision Tree memilih fitur yang paling berpengaruh untuk membagi data.
- Pemilihan ini didasarkan pada metrik seperti *Gini Index*, *Entropy*, atau *Variance Reduction*.

2. Memecah Data (Splitting)

- Data akan dibagi berdasarkan nilai fitur yang dipilih.
- Proses ini terus berulang hingga mencapai kondisi tertentu, seperti kedalaman maksimal atau jumlah data minimum dalam satu simpul.

3. Membentuk Struktur Pohon

- Setiap cabang dari decision tree merepresentasikan keputusan dari data yang dikelompokkan.

4. Membuat Prediksi

- Untuk prediksi, input data akan mengikuti jalur dari root node hingga leaf node berdasarkan aturan yang telah dibuat.

4. Metrik Pemilihan Fitur

- **Entropy & Information Gain (IG)**

- Entropy mengukur ketidakteraturan data, sedangkan Information Gain mengukur pengurangan ketidakteraturan setelah pemisahan data.

- Rumus Entropy:

$$H(S) = - \sum p_i \log_2 p_i$$

- Information Gain:

$$IG(S, A) = H(S) - \sum \frac{|S_v|}{|S|} H(S_v)$$

- **Gini Index**

- Digunakan dalam algoritma CART (Classification and Regression Tree).
- Rumus Gini Index:
$$Gini(S) = 1 - \sum p_i^2$$
- Nilai Gini yang lebih rendah menunjukkan pembagian yang lebih baik.

5. Algoritma Pembangunan Decision Tree

1. ID3 (Iterative Dichotomiser 3)

- Menggunakan *Information Gain* untuk memilih fitur terbaik.

2. C4.5

- Perbaikan dari ID3 yang menangani atribut numerik dan missing values.

3. CART (Classification and Regression Trees)

- Menggunakan *Gini Index* untuk pemilihan fitur dan bisa digunakan untuk regresi.

6. Kelebihan dan Kekurangan Decision Tree

- **Kelebihan**

- ✓ Mudah dimengerti dan diinterpretasikan.
- ✓ Tidak memerlukan banyak prapemrosesan data.
- ✓ Dapat menangani data kategorikal maupun numerik.
- ✓ Mampu menangani fitur yang tidak relevan dengan baik.

- **Kekurangan**

- ✗ Rentan terhadap overfitting jika tidak diatur dengan baik.
- ✗ Bisa menghasilkan model yang terlalu kompleks.
- ✗ Kurang optimal dalam menangani dataset besar dan high-dimensional.

Contoh Decission Tree ID3 dataset Iris

1. Dataset Iris

Dataset **Iris** memiliki **4 fitur** dan **3 kelas**:

Fitur	Tipe	Keterangan
Sepal Length (cm)	Numerik	Panjang kelopak
Sepal Width (cm)	Numerik	Lebar kelopak
Petal Length (cm)	Numerik	Panjang mahkota
Petal Width (cm)	Numerik	Lebar mahkota
Kelas	Kategorikal	Setosa, Versicolor, Virginica

2. Langkah-langkah ID3

1. **Hitung Entropy dataset awal.**
2. **Tentukan splitting point untuk setiap fitur.**
3. **Hitung Information Gain untuk setiap splitting point.**
4. **Pilih fitur dengan Information Gain tertinggi sebagai root node.**
5. **Pisahkan dataset berdasarkan fitur tersebut dan rekursif hingga semua data diklasifikasikan.**

3. Perhitungan Entropy Awal

- Sebelum memulai splitting, kita menghitung **Entropy** dari dataset awal:

Kelas	Jumlah
Setosa	50
Versicolor	50
Virginica	50
Total	150

$$\begin{aligned} H(S) &= - \sum p_i \log_2 p_i \\ &= - \left(\frac{50}{150} \log_2 \frac{50}{150} + \frac{50}{150} \log_2 \frac{50}{150} + \frac{50}{150} \log_2 \frac{50}{150} \right) \\ &= -(0.333 \times -1.585 + 0.333 \times -1.585 + 0.333 \times -1.585) \\ &= 1.585 \end{aligned}$$

4. Menentukan Splitting Point untuk Setiap Fitur

- Karena semua fitur dalam dataset **Iris** adalah numerik, kita harus mencari **nilai tengah antara dua titik berurutan sebagai kandidat splitting point**.

Splitting Point untuk Petal Length

No	Petal Length (cm)	Kelas
1	1.4	Setosa
2	1.5	Setosa
3	1.6	Setosa
4	1.7	Setosa
5	3.0	Versicolor
6	4.0	Versicolor
7	4.5	Versicolor

No	Petal Length (cm)	Kelas
8	5.1	Versicolor
9	5.8	Virginica
10	6.3	Virginica
11	6.7	Virginica
12	7.1	Virginica

Jumlah nilai unik = 12, maka jumlah kandidat splitting point adalah $(n-1) = 11$.

Menghitung semua kandidat splitting point:

$$1. \frac{1.4+1.5}{2} = 1.45$$

$$2. \frac{1.5+1.6}{2} = 1.55$$

$$3. \frac{1.6+1.7}{2} = 1.65$$

$$4. \frac{1.7+3.0}{2} = 2.35$$

$$5. \frac{3.0+4.0}{2} = 3.50$$

$$6. \frac{4.0+4.5}{2} = 4.25$$

$$7. \frac{4.5+5.1}{2} = 4.80$$

$$8. \frac{5.1+5.8}{2} = 5.45$$

$$9. \frac{5.8+6.3}{2} = 6.05$$

$$10. \frac{6.3+6.7}{2} = 6.50$$

$$11. \frac{6.7+7.1}{2} = 6.90$$

- Jadi, kandidat **splitting point untuk Petal Length** adalah:
- 1.45, 1.55, 1.65, 2.35, 3.50, 4.25, 4.80, 5.45, 6.05, 6.50, 6.90
- **Kandidat splitting point untuk fitur yang lain dihitung dengan cara yang sama**

5. Menghitung Information Gain untuk Setiap Splitting Point

- Misalnya, untuk **Petal Length ≤ 2.35 cm**:

Kelas	Jumlah
Setosa	50
Versicolor	0
Virginica	0
Total	50

- Karena semua dalam satu kelas, entropy **0**.

- Untuk **Petal Length** > 2.35 cm:

Kelas	Jumlah
Setosa	0
Versicolor	50
Virginica	50
Total	100

- $H(S_2) = 1$
- $IG(PetalLength = 2.45) = 1.585 - \left(\frac{50}{150} \times 0 + \frac{100}{150} \times 1 \right)$
- $= 1.585 - 0.667 = 0.918$

- Perhitungan IG dilakukan untuk semua kandidat split dari semua fitur.
- Split point dengan nilai IG tertinggi di masing-masing fitur dipilih sebagai split point di fitur tersebut.

- Setelah melakukan perhitungan IG untuk semua kandidat split point dari semua fitur didapat best split point sebagai berikut:

Fitur	Best Split	Information Gain
Petal Length	2.35 cm	0.918
Petal Width	1.75 cm	0.983
Sepal Length	5.8 cm	0.792
Sepal Width	3.0 cm	0.595

6. Memilih Fitur Root Node

- Karena **Petal Width (1.75 cm)** memiliki **IG tertinggi (0.983)**, maka **Petal Width \leq 1.75 cm** dipilih sebagai **root node**.

7. Root Node yang Sudah Ditetapkan

- Dari perhitungan sebelumnya, kita mendapatkan bahwa fitur dengan **Information Gain (IG) tertinggi** adalah **Petal Width** dengan split **1.75 cm**. Maka, kita pilih sebagai **root node**:

```
      Petal Width ≤ 1.75?  
      /           \  
    Yes          No
```

8. Pembagian Dataset Setelah Root Node

- Setelah membagi dataset berdasarkan **Petal Width**, kita memiliki dua subset:

(A) Subset 1: Petal Width ≤ 1.75 cm

Kelas	Jumlah
Setosa	50
Versicolor	0
Virginica	0

- Karena **semua data dalam subset ini adalah Setosa**, maka kita dapat membuat **leaf node** langsung:

```
      Petal Width ≤ 1.75?  
      /           \  
Yes (Setosa)      No
```

(B) Subset 2: Petal Width > 1.75 cm

Kelas	Jumlah
Versicolor	50
Virginica	50
Total	100

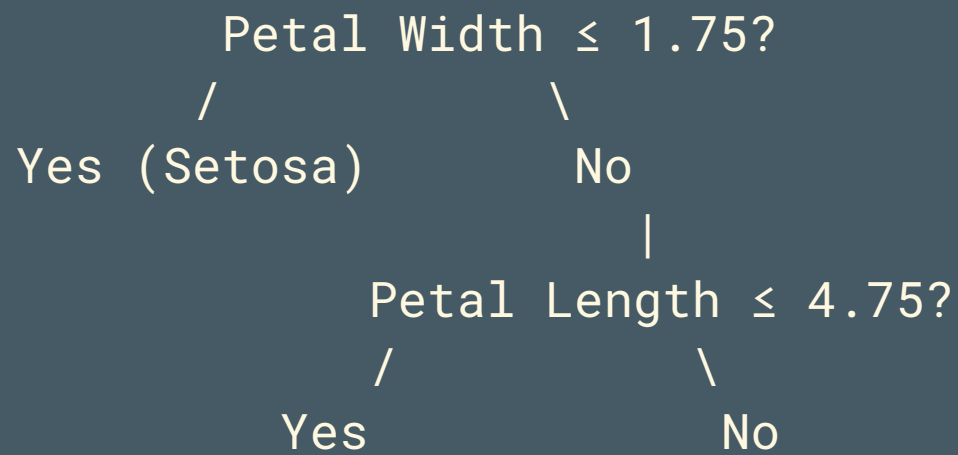
Karena **Versicolor** dan **Virginica** masih **bercampur**, kita perlu memilih **fitur terbaik** untuk **pemisahan lebih lanjut**.

9. Memilih Fitur Terbaik untuk Subset 2

- Sekarang, kita menghitung **Information Gain (IG)** untuk fitur lain dalam **Subset 2**:

Fitur	Best Split	Information Gain
Petal Length	4.75 cm	0.870
Sepal Length	5.8 cm	0.792
Sepal Width	3.0 cm	0.595

- Karena **Petal Length (4.75 cm)** memiliki **IG tertinggi (0.870)**, kita memilih **Petal Length** sebagai fitur selanjutnya:



10. Memisahkan Data Berdasarkan Petal Length ≤ 4.75 cm

- Sekarang kita membagi subset (**Petal Width > 1.75 cm**) berdasarkan **Petal Length ≤ 4.75 cm**.

(A) Subset 1: Petal Length ≤ 4.75 cm

Kelas	Jumlah
Versicolor	50
Virginica	0

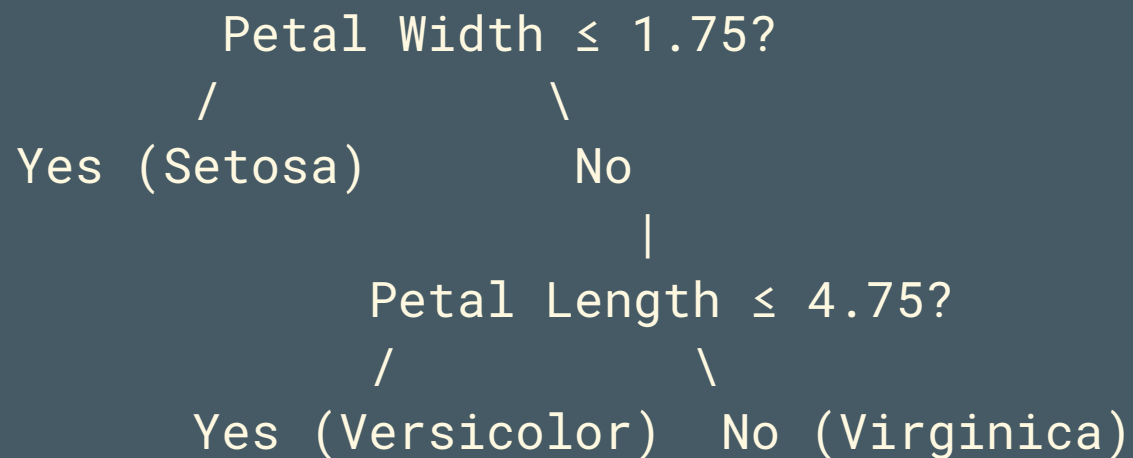
Karena **semua data adalah Versicolor**, kita buat **leaf node**:

```
      Petal Width  $\leq$  1.75?  
      /           \  
Yes (Setosa)      No  
                  |  
      Petal Length  $\leq$  4.75?  
      /           \  
Yes (Versicolor) No
```


(B) Subset 2: Petal Length > 4.75 cm

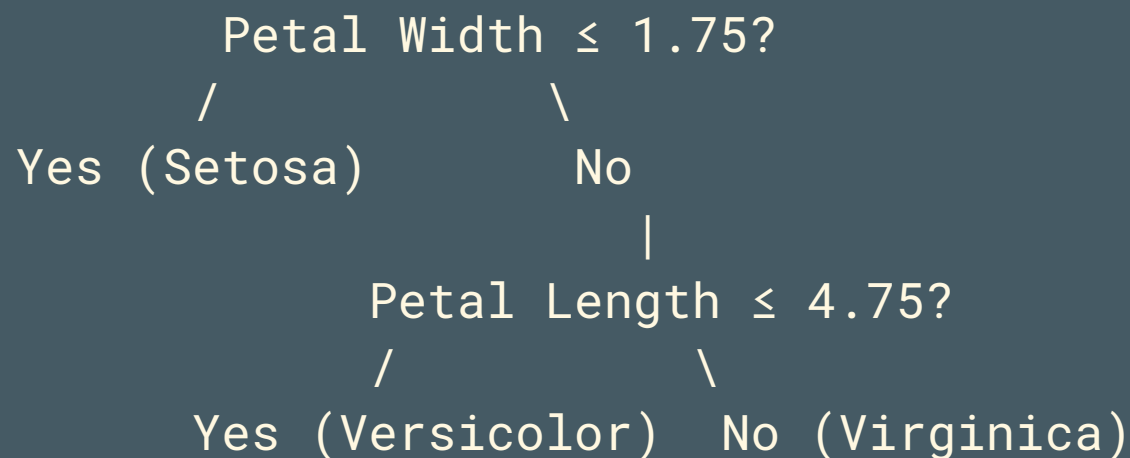
Kelas	Jumlah
Versicolor	0
Virginica	50

- Karena **semua data adalah Virginica**, kita buat **leaf node**:



11. Decision Tree Final

Setelah semua subset hanya mengandung satu kelas, kita memiliki pohon keputusan final:



Metode C4.5

- Perbedaan dengan ID3 adalah penggunaan Gain Ratio (GR) untuk menggantikan IG dalam menentukan Split Point dan fitur pada node.
- **Gain Ratio (GR)** dihitung dengan membagi Information Gain dengan Split Information (SI):

$$GR(A) = \frac{IG(A)}{SI(A)}$$

- di mana **Split Information (SI)** dihitung sebagai:

$$SI(A) = - \sum_{i=1}^k \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

Penjelasan Notasi:

- A = fitur yang sedang diuji sebagai kandidat split.
- S = jumlah total sampel.
- k = jumlah subset setelah pembagian berdasarkan fitur A .
- S_i = jumlah sampel dalam subset i .
- $|S_i|/|S|$ = proporsi sampel dalam subset i .
- **Split Information (SI)** mengukur seberapa banyak data terbagi berdasarkan fitur yang digunakan untuk split.

Contoh Perhitungan Split Information (SI)

- Misalkan kita membagi dataset berdasarkan **Petal Length ≤ 2.43 cm**, yang menghasilkan dua subset:

Subset	Jumlah Sampel	Proporsi
S1 (Petal Length ≤ 2.35 cm)	50	$\frac{50}{150} = 0.333$
S2 (Petal Length > 2.35 cm)	100	$\frac{100}{150} = 0.667$

$$\begin{aligned} SI(PetalLength) &= - \left(\frac{50}{150} \log_2 \frac{50}{150} + \frac{100}{150} \log_2 \frac{100}{150} \right) \\ &= - (0.333 \log_2 0.333 + 0.667 \log_2 0.667) \\ &= - (0.333 \times -1.585 + 0.667 \times -0.585) \\ &= - (-0.528 - 0.390) = 0.918 \end{aligned}$$

Sehingga, **Split Information untuk Petal Length = 0.918.**

CART

- ID3 menggunakan Entropy dan Information Gain untuk memilih fitur terbaik.
- CART menggunakan Gini Index untuk memilih fitur terbaik dan hanya menghasilkan pohon biner.

Rumus Gini Index

- Gini Index untuk suatu himpunan data **S** dihitung sebagai:
 - $Gini(S) = 1 - \sum p_i^2$
 - di mana:
 - p_i adalah **proporsi** dari kelas ke- i dalam dataset.
 - $\sum p_i^2$ adalah jumlah dari kuadrat probabilitas masing-masing kelas.
- Gini Index **bernilai 0 jika dataset sepenuhnya homogen** (hanya satu kelas) dan **bernilai maksimum jika distribusi kelas merata**.

Contoh Perhitungan Gini Index

- Misalkan kita memiliki dataset dengan **150 sampel** dalam tiga kelas:

Kelas	Jumlah	Proporsi (p_i)
Setosa	50	$50/150 = 0.333$
Versicolor	50	$50/150 = 0.333$
Virginica	50	$50/150 = 0.333$

- $Gini(S) = 1 - ((0.333)^2 + (0.333)^2 + (0.333)^2)$
- $= 1 - (0.111 + 0.111 + 0.111)$
- $= 1 - 0.333 = 0.667$
- Jadi, **Gini Index untuk dataset awal adalah 0.667.**

Gini Index Setelah Splitting

- Misalkan kita membagi dataset berdasarkan **Petal Length ≤ 2.35 cm**, yang menghasilkan **dua subset**:
- **Subset 1 (Petal Length ≤ 2.45 cm)**

Kelas	Jumlah	Proporsi (p_i)
Setosa	50	$50/50 = 1.0$
Versicolor	0	$0/50 = 0.0$
Virginica	0	$0/50 = 0.0$

- $Gini(S_1) = 1 - (1.0^2 + 0.0^2 + 0.0^2) = 1 - 1 = 0$

- **Subset 2 (Petal Length > 2.35 cm)**

Kelas	Jumlah	Proporsi (p_i)
Setosa	0	$0/100 = 0.0$
Versicolor	50	$50/100 = 0.5$
Virginica	50	$50/100 = 0.5$

- $Gini(S_2) = 1 - (0.5^2 + 0.5^2) = 1 - (0.25 + 0.25) = 1 - 0.5 = 0.5$

Weighted Gini Index (Setelah Split)

- $Gini_{split} = \left(\frac{50}{150} \times Gini(S_1)\right) + \left(\frac{100}{150} \times Gini(S_2)\right)$
- $= (0.333 \times 0) + (0.667 \times 0.5)$
- $= 0 + 0.333 = 0.333$

1. **Gini Index** digunakan dalam **CART** untuk mengukur impurity.
2. **Semakin kecil Gini Index, semakin homogen** data dalam node tersebut.
3. **CART memilih fitur dengan Gini Gain tertinggi** sebagai fitur terbaik untuk splitting.