CART (Classification and Regression Tree)

Adinda Putri - 13523071

CART merupakan decision tree yang digunakan untuk mempelajari hubungan dan pola dalam dataset dan membuat struktur berdasarkan nilai variabel dalam dataset tersebut. Saat proses konstruksi struktur tree, model memilih variabel dan nilai threshold yang paling baik untuk membagi data dan melakukan classification atau regression. Pada kali ini, akan dibahas hanya CART untuk classification. Kelebihan utama dari CART adalah model ini dapat mengukur pengaruh dan relevansi variabel dalam dataset. Hal ini dapat meningkatkan akurasi model dan bekerja baik untuk dataset nonlinear.

Cara Kerja

1. Buat struktur **tree-like** yang terdiri dari **nodes** dan **branches**. Node merepresentasikan decision point, branch merepresentasikan possible outcomes dari berbagai decision tersebut. Terdapat juga **leaf nodes** yang berisi nilai untuk variabel target.
2. Hitung probabilitas kelas di node dengan rumus berikut

$$p_i = \frac{\text{jumlah data dengan kelas i}}{\text{total data di node}}$$

Dengan yi merupakan label class ke-i

3. Untuk setiap fitur Xj, ambil semua nilai unik dan hitung kandidat thresholdnya (median)

$$\text{thresholds} = \frac{x_k + x_{k+1}}{2}$$

4. HItung Gini impurity untuk tiap split

$$Gini = 1 - \sum_{i=1}^{K} p_i^2$$

dengan K adalah jumlah class unik pada node tersebut.

5. Hitung weighted Gini untuk split menjadi dua subset

$$Gini_{split} = \frac{N_{left}}{N} Gini(D_{left}) + \frac{N_{right}}{N} Gini(D_{right})$$

6. Pilih fitur + thresholdnya dengan **weighted Gini terkecil** sebagai best split

7. Buat node baru untuk menyimpan fitur terbaik, threshold terbaik dan probabilitas kelas di node tersebut

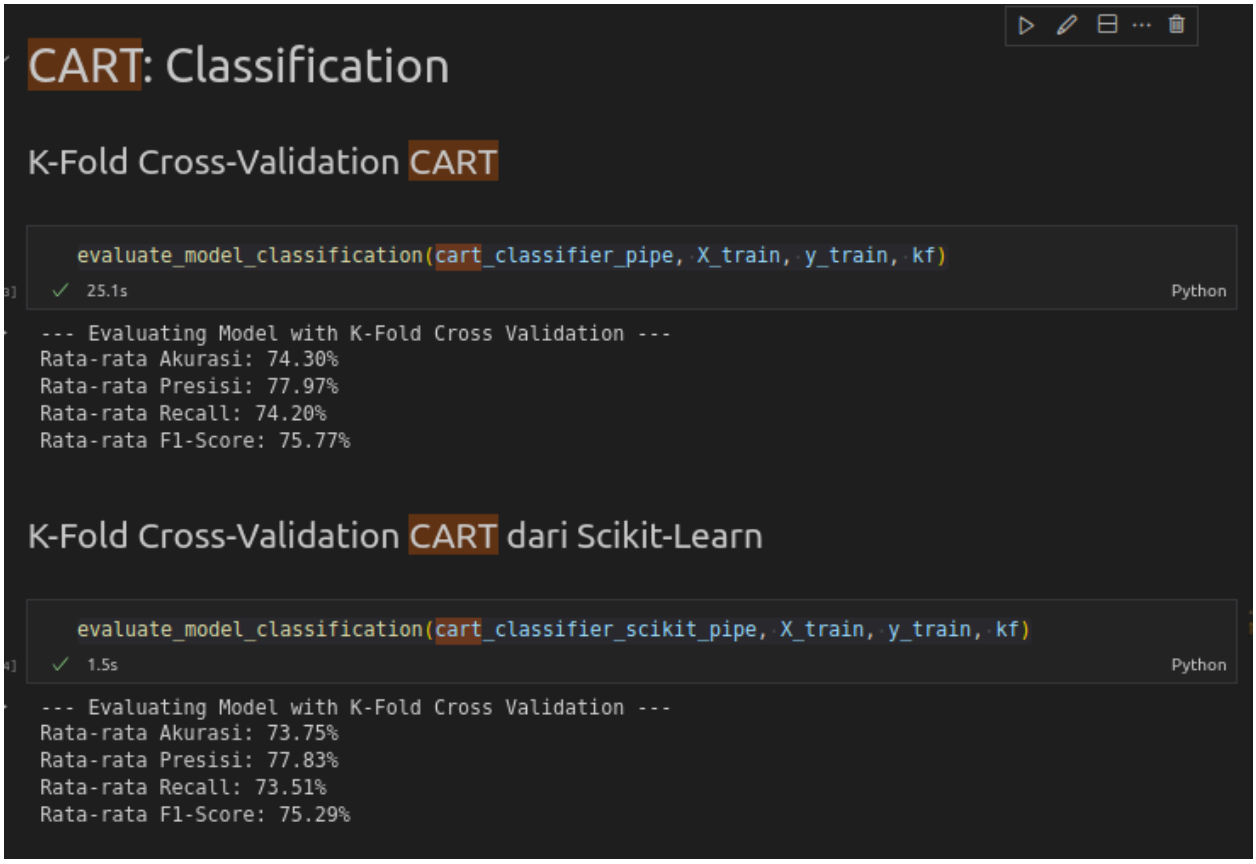
8. Split data menjadi dua subset dengan subset kiri nilai fitur <= threshold dan subset kanan nilai fitur > threshold

9. Lakukan langkah 4-8 secara rekursif hingga max_depth tercapai, min_samples_split bernilai False atau semua node pure (hanya satu kelas)

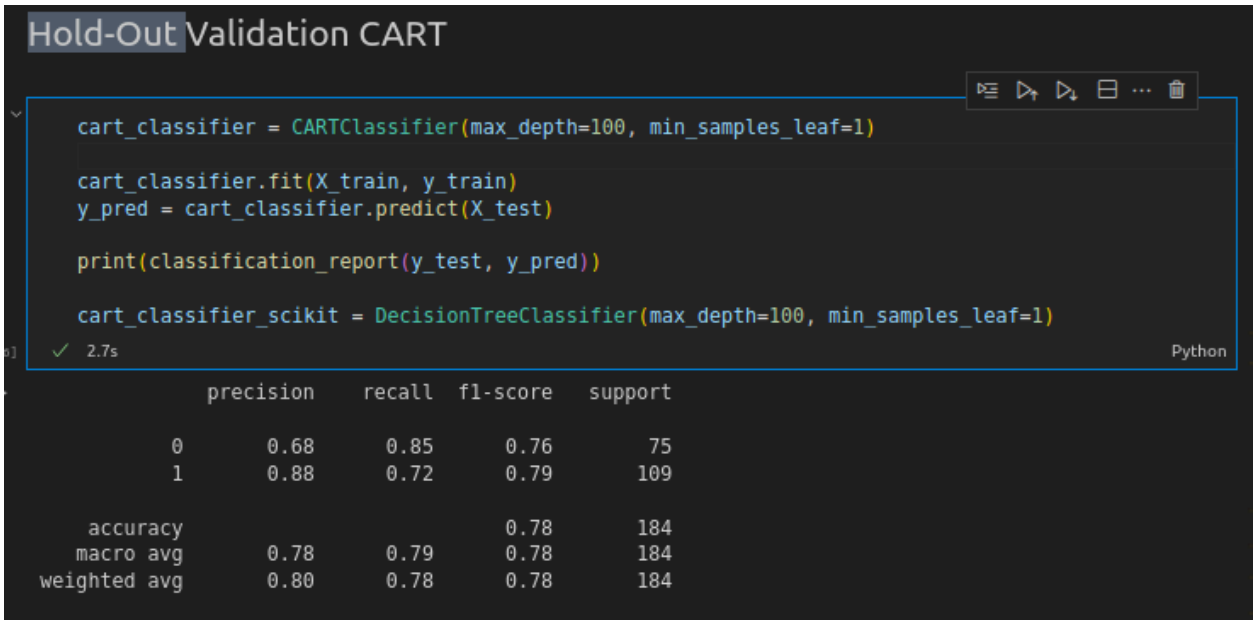
10. Leaf node menyimpan probabilitas kelas dan tidak bisa di-split lagi.

11. Dilakukan prediksi untuk tiap sampel.

Perbandingan model dari scratch dan dari Scikit-Learn

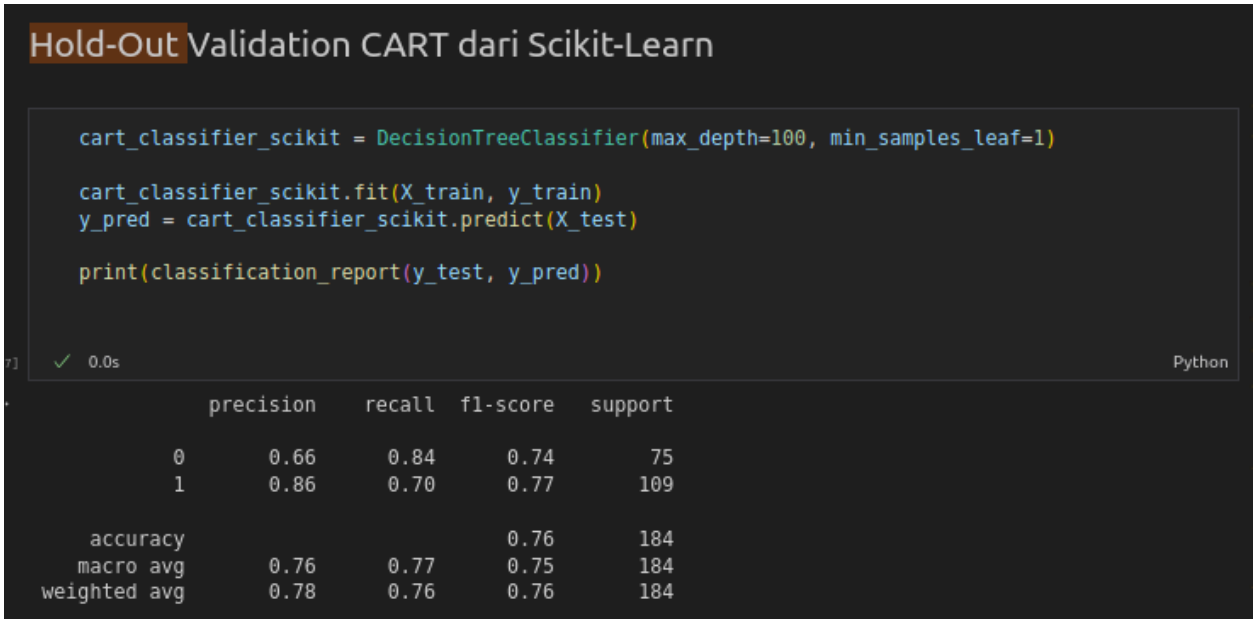


Gambar 1. K-Fold Cross-Validation CART dari Scratch dan CART dari Sklearn
Sumber: Penulis



Gambar 2. Hold-Out Validation CART dari Scratch

Sumber: Penulis



Gambar 3. Hold-Out Validation Cart dari Sklearn
Sumber: Penulis

Terdapat perbedaan hasil pada model yang mungkin terjadi karena implementasi algoritma yang berbeda. Meskipun keduanya tidak menggunakan prunning, dan mendapatkan nilai parameter yang sama. Akan tetapi, implementasi pencarian nilai thresholdnya mungkin berbeda. Selain itu algoritma CART versi Scikit-Learn sudah optimized.

Ruang Improvement:

- Optimisasi pemilihan threshold dengan binary search, quicksort, mergesort dengan optimization.
- Eksperimen dengan exhaustive search untuk melakukan pencarian threshold lebih komprehensif
- Model dari scratch belum menggunakan prunning. Dapat diterapkan cost-complexity prunning untuk menghindari overfitting
- Optimisasi tree operation seperti memaksimalkan vectorized operation dan gunakan in-place operations dan data struktur yang efisien.

Referensi:

[1] *Decision Trees in Machine Learning: CART and Advanced Trees*, Medium oleh Kaan Erden, 16 April 2023. [Daring]. Tersedia:

https://medium.com/@kaanerdenn/decision-trees-in-machine-learning-cart-and-advanced-trees-8b3fe375e9f9. [Diakses: 2 September 2025].

[2] *Decision Tree (CART) from scratch. Full tutorial*, Kaggle Notebook oleh Egazakharenko, 2025. [Daring]. Tersedia: https://www.kaggle.com/code/egazakharenko/decision-tree-cart-from-scratch-full-tutorial. [Diakses: 2 September 2025].