Seleksi Asisten Lab AI '23

Supervised Learning: CART

---

1. Cara Kerja Algoritma CART.

Konsep inti: Algoritma CART (Classification and Regression Trees) membangun sebuah *decision tree* dengan melakukan *split* biner secara rekursif. Tujuannya adalah untuk mempartisi data menjadi subset yang semakin homogen dalam hal komposisi kelas.

Pseudocode untuk membangun *Decision Tree*:

Algorithm 1

function GrowTree($X, y, depth$)
    $n_{samples}, n_{features} \leftarrow$ shape of X
    $n_{labels} \leftarrow$ number of unique labels in y

    if $depth \geq max\_depth$ **or** $n_{labels} == 1$ **or** $n_{samples} < min\_samples\_split$ **then**
        $leaf\_value \leftarrow$ most common label in y
        return Node(value=$leaf\_value$)
    end if

    $best\_feat, best\_thresh \leftarrow$ FindBestSplit($X, y, n_{features}$)

    if $best\_feat$ is None **then**
        $leaf\_value \leftarrow$ most common label in y
        return Node(value=$leaf\_value$)
    end if

    $left\_idxs, right\_idxs \leftarrow$ SplitData($X[:, best\_feat], best\_thresh$)
    $left\_child \leftarrow$ GrowTree($X[left\_idxs, :], y[left\_idxs], depth + 1$)
    $right\_child \leftarrow$ GrowTree($X[right\_idxs, :], y[right\_idxs], depth + 1$)

    return Node(feature=$best\_feat$, threshold=$best\_thresh$, left=$left\_child$, right=$right\_child$)
end function

2. Analisis Perbandingan Eksperimen.

Table 1: Perbandingan kinerja model CART

Metric	**From Scratch**	**Scikit-learn**
Accuration	0.7942	0.7942
Precision	0.63	0.63
Time (s)	∼2.08	∼0.02

Hasil kinerja prediktif antara implementasi *from scratch* dan Scikit-learn hampir identik. Waktu eksekusi yang cepat pada model *from scratch* disebabkan oleh pembatasan `max_depth=5`, yang secara efektif mengurangi kompleksitas komputasi.

3. *Improvement* yang bisa dilakukan untuk mencapai hasil yang lebih baik.

Beberapa bagian yang dapat ditingkatkan adalah

- Pruning: Pohon keputusan yang terlalu dalam cenderung *overfitting*. Teknik *pruning* dapat digunakan setelah pohon dibangun untuk memangkas cabang-cabang yang kurang memberikan informasi prediktif, sehingga meningkatkan kemampuan generalisasi model pada data baru.
- Tuning Hyperparameter: Mungkin perlu dilakukan pencarian sistematis untuk kombinasi `max_depth` dan `min_samples_split` yang optimal menggunakan *Grid Search* dengan *cross-validation* akan menghasilkan model yang lebih seimbang antara bias dan varians.