Nama	: Adhimas Aryo Bimo
NIM	: 13523052
Model	: *DecisionTreeRegressor*

1. Singkatnya DecisionTreeRegressor merupakan model berbasis Tree untuk melakukan prediksi. Selayaknya tree, model ini memiliki node dan melakukan split pada node. Nah, inti dari DecisionTreeRegressor ini adalah cara untuk melakukan split tersebut.
   a. Menghitung Error Awal
      - Error awal dihitung dengan menghitung nilai target tiap fitur dengan rata-rata y dari seluruh data. Pada hal ini, saya menggunakan MSE untuk menghitung error awal
   b. Cari split terbaik
      - Untuk setiap fitur j, urutkan nilai $X[:, j]$
      - Coba setiap threshold yang memisahkan data menjadi kiri dan kanan
      - Lalu, hidung MSE kiri dan MSe kanan dan gabungkan jadi weighted MSE dengan formula sebagai berikut:
      $$\text{Weighted MSE} = \frac{n_{\text{left}} \cdot \text{MSE}_{\text{left}} + n_{\text{right}} \cdot \text{MSE}_{\text{right}}}{n}$$
      - Nah lalu hitung **gain,** ini penting untuk menentukkan spit yang bagus. Hitungnya dengan gain = MSE_parent – Weighted_MSE.
   c. Cek stopping criteria.
      - Di sini untuk berhenti membelah menjadi leaf, tujuannya agar tidak overfit tentunya
      - Ada beberapa kriteria untuk berhenti:
        - Kedalaman sudah mencapai max_depth, atau
        - Jumlah sample < min_samples_split, atau
        - Tidak ada split dengan gain cukup besar (gain < min_impurity_decrease), atau
        - Split menghasilkan child dengan sample < min_samples_leaf

        Ketika berhenti, leaf node diset dengan rata-rata nilai target di node itu:
        $$\text{value} = \frac{1}{n}\sum y_i$$
   d. Rekursif dari kiri ke kanan
      - Jika split valid, buat left node dengan data $X[left], y[left]$
      - Buat right node dengan data $X[right], y[right]$
      - Panggil fungsi build ulang untuk masing-masing node sampai semua memenuhi kondisi berhenti.

e. Prediksi (inference)
    - Mulai dari root node.
    - Cek apakah sampel x lebih kecil atau sama dengan threshold di node saat ini.
    - Jika ya → turun ke left child, jika tidak → turun ke right child.
    - Ulangi sampai ketemu leaf, lalu output = nilai leaf (rata-rata target di leaf itu).

2. Ada beberapa hal yang bisa membuat hasil punya saya berbeda dengan sklearn. Secara umum, sklearn pakai "variance reduction" alias impurity decrease. Saya menerapkan MSE reduction yang sebenarnya mirip. Tapi implementasi detailnya beda jadi hasil split beda dan tree beda dan konsekuensinya skor berbeda.

3. Improvement yang bisa dilakuan mungkin bisa diperbaiki feature engineeringnya untuk mempermudah split dari model ini. Dari segi parameter, bisa ditambah max_depth nya agar splitnya bisa menjangkau pola lebih dalam. Tapi hati-hati karena bisa overfit. Ultimatenya, bisa dibuat model ensambel dengan dasar pohon dari model ini.