KNN (K-Nearest Neighbor)

Evelyn Yosiana / 13522083

1. Prinsip kerja algoritma KNN yaitu berdasarkan jarak suatu titik ke titik lainnya.
   Proses train: untuk proses train, pada dasarnya class ini (dalam kode saya diberi nama KNN_Selfmade) hanya menyimpan data atribut-atribut (X_train) dan data target (y_train) dalam bentuk array numpy.
   Proses predict: untuk proses ini, tiap data yang akan diprediksi (X_test) dihitung jaraknya dengan tiap data yang ada pada X_train dengan menggunakan metrik yang dipilih. Kemudian dicari sejumlah N data dengan jarak terdekat dengan data yang dites. Kemudian dari N data terdekat tersebut, dicari modus labelnya yang kemudian akan menjadi hasil prediksi dari data yang dites tersebut. Proses tersebut diulangi untuk seluruh data di X_test.
   Parameter:
   - neighbors: jumlah tetangga (jumlah titik data terdekat yang akan dicari modusnya, dalam penjelasan proses fit disebut N). By default = 3.
   - metric: metrik perhitungan yang digunakan untuk mencari jarak antar dua data. Dalam kode saya terdapat 3 metrik yang dapat dipilih yaitu euclidean, manhattan, dan minkowski. By default = euclidean.
   - p: parameter order yang digunakan untuk metrik minkowski. By default = 5.
4. Pada model ini, hasil perbandingan model yang saya buat dan model dari library sama jika menggunakan parameter yang sama.
5. Improvement yang dapat saya lakukan antara lain:
   - **Normalisasi data** karena perhitungan pada KNN (baik dengan euclidean, manhattan, maupun minkowski) sangat bergantung pada jarak tiap data sehingga melakukan normalisasi dapat membantu mencegah fitur dengan skala besar mendominasi perhitungan.
   - **Membuat fitur baru** yang lebih relevan (feature engineering) berdasarkan fitur-fitur yang sudah ada.
   - **Hyperparameter tuning** untuk mendapatkan kombinasi parameter yang optimal (salah satunya dengan menggunakan metode grid search atau library optuna).
   - **Parallel processing** untuk mempercepat pemrosesan, terutama jika data yang digunakan cukup besar.
   - Menggunakan **KD-trees atau ball trees** untuk mempercepat pencarian tetangga terdekat. KD-trees dan ball trees membagi-bagi data menjadi

beberapa section sehingga pencarian tetangga akan lebih cepat jika berada di section yang sama atau section yang berdekatan.

- Menggunakan **confusion metrics** untuk model evaluation.