Chapter 3 : Classification

Bab ini memperkenalkan konsep klasifikasi, yaitu jenis supervised learning di mana output berupa label diskret (bukan angka kontinu seperti regresi).

A. Proyek dalam Bab Ini: Mendeteksi Angka Tulisan Tangan (Digit Classification)

Dataset: MNIST

Dataset terkenal yang berisi gambar digit tulisan tangan (angka 0 sampai 9).

Setiap gambar: 28x28 piksel, grayscale → 784 fitur (28×28 = 784).

Total data: 70.000 gambar (60.000 training, 10.000 test)

Tujuan proyek: Melatih model machine learning untuk mengklasifikasikan gambar ke dalam kelas 0 sampai 9.

B. Langkah-Langkah dan Topik yang Dipelajari

1. Load Dataset

Menggunakan fetch_openml("mnist_784") dari Scikit-Learn

Dataset dimuat ke dalam Pandas DataFrame dan NumPy array

2. Visualisasi Data

Menampilkan gambar pertama dengan matplotlib.imshow()

Memastikan label cocok dengan gambar

3. Membuat Model Klasifikasi

Model pertama: Stochastic Gradient Descent (SGD) Classifier

Diterapkan pada seluruh dataset (60.000 data latih)

4. Evaluasi Model

Cross-validation dengan cross_val_score

Metrik evaluasi awal: akurasi

5. Confusion Matrix

Matrix yang menunjukkan prediksi benar/salah tiap kelas

Membantu mengidentifikasi kesalahan spesifik

6. Precision, Recall, F1 Score

Precision: berapa banyak prediksi positif yang benar?

Recall: berapa banyak data positif yang berhasil ditemukan?

F1 Score: harmonic mean dari precision dan recall

Gunakan precision_score, recall_score, dan f1_score dari sklearn.metrics

7. Thresholding dan Skor Probabilitas

Beberapa model (seperti SGD) dapat menghasilkan skor, bukan hanya label

Dapat menyetel threshold untuk mengontrol precision vs recall

8. Multiclass Classification

MNIST adalah 10-class classification

Scikit-Learn menangani multiclass dengan One-vs-All (OvA) secara otomatis

9. Error Analysis

Menampilkan gambar-gambar yang diklasifikasikan salah

Menganalisis jenis kesalahan → misalnya 5 vs 3 atau 8 vs 9

10. Multilabel Classification

Contoh: deteksi apakah angka itu “besar dari 7” dan “ganjil”

Model bisa memiliki beberapa label per instance

Digunakan KNeighborsClassifier untuk mendukung multilabel

11. Multioutput Classification

Contoh: hapus noise dari gambar

Input = gambar berisik, Output = gambar asli

Ini seperti autoencoder sederhana

C. Pelajaran dari chapter ini

Memahami:

- Cara kerja klasifikasi dasar menggunakan dataset gambar (MNIST)
- Evaluasi model klasifikasi menggunakan metrik yang tepat
- Bagaimana menangani klasifikasi biner, multiclass, multilabel, dan multioutput
- Teknik thresholding untuk mengatur trade-off antara precision dan recall
- Pentingnya error analysis visual untuk perbaikan model

D. Algoritma yang Diperkenalkan

SGDClassifier (klasifikasi linear berbasis gradient descent)

KNeighborsClassifier (untuk multilabel dan multioutput)

E. Insight Kunci

Akurasi tinggi tidak berarti model bagus, terutama jika datanya tidak seimbang.

Evaluasi dengan confusion matrix dan F1 Score sering lebih informatif.

Visualisasi prediksi salah sangat penting untuk memahami model secara intuitif.

Threshold kustomisasi memberikan fleksibilitas berdasarkan kebutuhan bisnis (misalnya deteksi kanker → recall tinggi lebih penting).