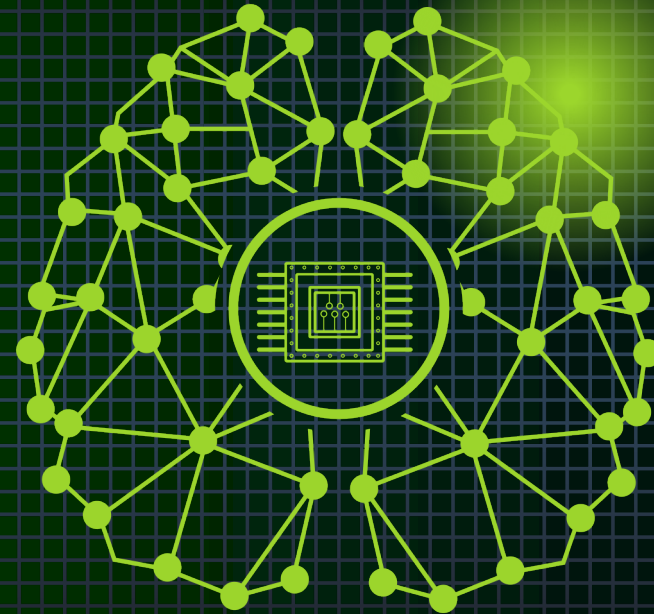


EIE6012

MACHINE LEARNING

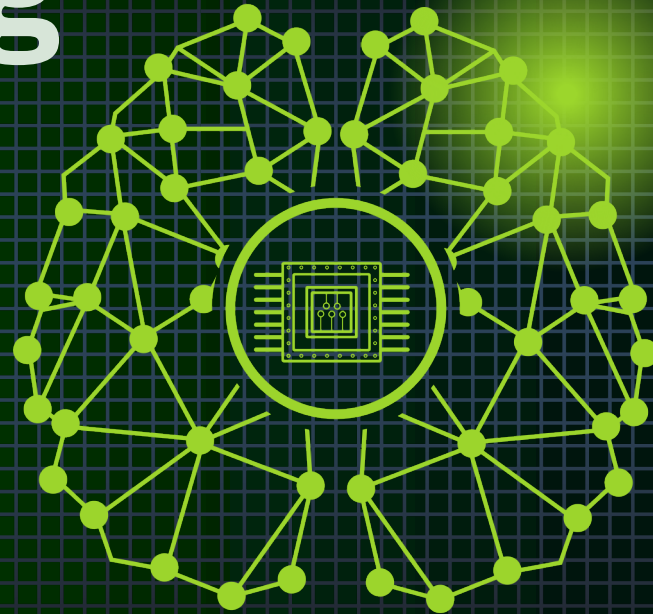
ENDANG SRI RAHAYU



2

Pre-processing

Data cleaning; data transformation



Garbage in, garbage out

Tujuan Pembelajaran

Setelah perkuliahan ini, mahasiswa mampu:

- Menjelaskan pentingnya tahap data preprocessing dalam machine learning.
- Melakukan pembersihan dan transformasi data sebelum training model.

Tahapan Proses dalam Machine Learning

Data Collection – mengumpulkan data dari sumber relevan.

Data Preprocessing – membersihkan data, menangani nilai kosong, normalisasi.

Feature Selection/Extraction – memilih fitur yang paling relevan.

Model Training – melatih model menggunakan algoritma.

Model Evaluation – mengukur kinerja model.

Deployment – menerapkan model ke dunia nyata.

Data sebagai Inti Machine Learning

Konsep utama:

Dataset → kumpulan data yang digunakan untuk melatih dan menguji model.

Fitur (features) → atribut atau variabel yang menjadi input model.

Label (target/output) → nilai keluaran yang ingin diprediksi (pada supervised learning).

Dataset prediksi harga rumah

Luas (m ²)	Jumlah kamar	Lokasi	Harga (Label)
80	3	Bandung	950 juta
60	2	Bekasi	600 juta



Masalah umum pada data mentah:

- Nilai hilang (missing values)
- Data duplikat
- Skala antar fitur berbeda jauh
- Data kategorikal (teks) tidak bisa langsung diolah
- Outlier (data ekstrem) yang merusak pola

Tahapan umum preprocessing:

- Data Cleaning - menghapus duplikat, mengisi nilai hilang
- Data Transformation - scaling, encoding, normalisasi
- Feature Selection / Extraction - memilih fitur paling relevan

Data Cleaning

Menangani Missing Values

- `fillna()` mengisi nilai hilang dengan rata-rata kolom.
- Bisa juga diganti dengan median, modus, atau nilai konstan tergantung konteks.

```
import pandas as pd
import numpy as np

# Contoh dataset
data = pd.DataFrame({
    'Usia': [25, 30, np.nan, 45, 50],
    'Pendapatan': [2000, 2500, 3000, np.nan, 5000],
    'Kelas': ['A', 'B', 'A', 'B', 'A']
})

print("Data Asli:")
print(data)

# Mengisi missing value dengan rata-rata kolom (versi aman)
data['Usia'] = data['Usia'].fillna(data['Usia'].mean())
data['Pendapatan'] = data['Pendapatan'].fillna(data['Pendapatan'].mean())

print("\nData Setelah Imputasi:")
print(data)
```

Feature Scaling

Ketika fitur memiliki rentang nilai berbeda (misal "Pendapatan" bisa 1000-10000 sementara "Umur" hanya 20-60), model cenderung bias terhadap fitur berskala besar.

Dua metode umum:

Normalization (Min-Max Scaling):

ubah nilai ke rentang [0, 1]

Standardization (Z-score): ubah data agar memiliki mean=0 dan std=1

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler
```

```
X = data[['Usia', 'Pendapatan']]
```

```
# Normalisasi
```

```
scaler_minmax = MinMaxScaler()
```

```
X_norm = scaler_minmax.fit_transform(X)
```

```
print("Hasil Normalisasi:\n", X_norm)
```

```
# Standarisasi
```

```
scaler_std = StandardScaler()
```

```
X_std = scaler_std.fit_transform(X)
```

```
print("\nHasil Standarisasi:\n", X_std)
```

Normalisasi → cocok untuk algoritma berbasis jarak (KNN, K-Means)

Standarisasi → cocok untuk model linear (regresi, logistic)

Encoding Data Kategorikal

Model ML hanya menerima angka, jadi fitur berbentuk teks harus dikonversi.

Contoh One-Hot Encoding

```
from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder(sparse_output=False)
encoded = encoder.fit_transform(data[['Kelas']])

print("Hasil One-Hot Encoding:")
print(encoded)
```

Setiap kategori diubah menjadi kolom baru dengan nilai 0 atau 1.

Tugas Rumah / Latihan

1. Carilah contoh dataset publik (misalnya dari Kaggle atau UCI).
2. Deskripsikan fitur dan label dari dataset tersebut.
3. Jelaskan langkah-langkah preprocessing yang mungkin perlu dilakukan.

