Fikri Aliansyah Juana

05TPLE017

# Pertemuan 4: Data Preparation

## Langkah

Langkah 1: Import Library

pandas: Untuk manipulasi dan analisis data

numpy: Untuk operasi numerik

matplotlib & seaborn: Untuk visualisasi data

sklearn.model\_selection: Untuk membagi dataset (meski belum digunakan)

1.1 Load Data

df = pd.read\_csv('kelulusan\_mahasiswa.csv')

Membaca dataset dari file CSV ke dalam DataFrame pandas.

Langkah 2: Pemeriksaan Data Dasar

2.1 Check Missing Values

print("Missing values:")

print(df.isnull().sum())

Mengecek apakah ada data yang hilang (null/NaN)

Output menunjukkan semua kolom memiliki 0 missing values

2.2 Check Duplicates

print(f"\nDuplicates: {df.duplicated().sum()}")

Mengecek data duplikat

Output: 0 duplikat ditemukan

2.3 Check Data Info

print("\nData info:")

print(df.info())

Menampilkan informasi struktur data:

10 entri (baris data)

4 kolom: IPK (float64), Jumlah\_Absensi (int64), Waktu\_Belajar\_Jam (int64), Lulus (int64)

Memory usage: 448.0 bytes

3. Exploratory Data Analysis (EDA)

3.1 Visualisasi Distribusi Variabel

Membuat 4 subplot histogram:

IPK: Distribusi nilai IPK mahasiswa

Jumlah Absensi: Frekuensi absensi

Waktu Belajar: Jam belajar per minggu

Kelulusan: Status lulus (0=Tidak, 1=Ya)

3.2 Analisis Korelasi

correlation\_matrix = df.corr()

sns.heatmap(correlation\_matrix, annot=True)

Heatmap untuk melihat hubungan antar variabel

Menampilkan korelasi setiap fitur dengan target 'Lulus'

4. Feature Engineering

Membuat 2 fitur baru:

4.1 Rasio Efisiensi Belajar

df['Rasio\_Efisiensi\_Belajar'] = df['IPK'] / df['Waktu\_Belajar\_Jam']

Mengukur seberapa efisien waktu belajar menghasilkan IPK

Semakin tinggi nilai, semakin efisien proses belajar

4.2 Skor Performa Komprehensif

df['Skor\_Performa\_Komprehensif'] = df['IPK'] \* (1 - df['Jumlah\_Absensi']/15)

Menggabungkan IPK dan tingkat kehadiran

Asumsi: maksimal 15 absensi yang wajar

Semakin tinggi skor, semakin baik performa akademik

5. Data Splitting untuk Machine Learning

# Split 7/2/1: Training/Validation/Test

X\_temp, X\_test, y\_temp, y\_test = train\_test\_split(X, y, test\_size=1)

X\_train, X\_val, y\_train, y\_val = train\_test\_split(X\_temp, y\_temp, test\_size=2)

Split Strategy:

Training: 7 samples (70%)

Validation: 2 samples (20%)

Test: 1 sample (10%)

## Kesimpulan

1. Kualitas Data:

Data bersih: Tidak ada missing values atau duplikat

Struktur baik: 10 observasi, 4 kolom awal + 2 fitur engineering

Dataset sangat kecil: Hanya 10 samples (limited untuk model kompleks)

2. Karakteristik Dataset:

Target: Variabel 'Lulus' (klasifikasi biner)

Fitur original: IPK, Jumlah\_Absensi, Waktu\_Belajar\_Jam

Fitur engineering: 2 fitur baru yang meaningful

3. Insight dari EDA:

Distribusi variabel dapat dilihat melalui histogram

Korelasi antar fitur dapat diidentifikasi melalui heatmap

Fitur baru menambahkan perspektif analisis yang lebih komprehensif

4. Persiapan Machine Learning:

Split ratio 7:2:1 appropriate untuk dataset kecil

Stratification: Tidak digunakan karena dataset terlalu kecil

Data sudah siap untuk modeling dengan fitur yang diperkaya

5. Rekomendasi Next Steps:

Model sederhana: Decision Tree, Logistic Regression (cocok untuk data kecil)

Cross-validation: Penting karena dataset kecil

Feature importance: Analisis kontribusi fitur baru

Data collection: Perlukan lebih banyak samples untuk model yang robust

6. Potensi Issue:

Overfitting risk: Dataset sangat kecil (10 samples)

Generalization: Model mungkin kesulitan belajar pattern yang kompleks

Statistical significance: Hasil mungkin kurang signifikan secara statistik