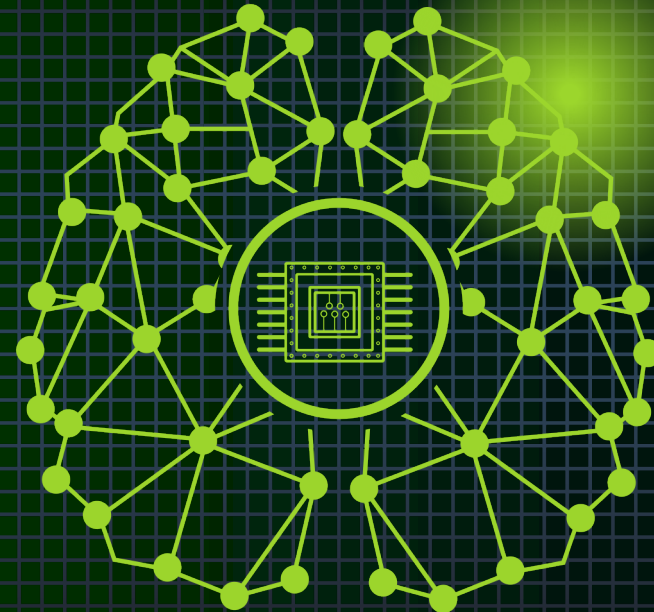


EIE6012

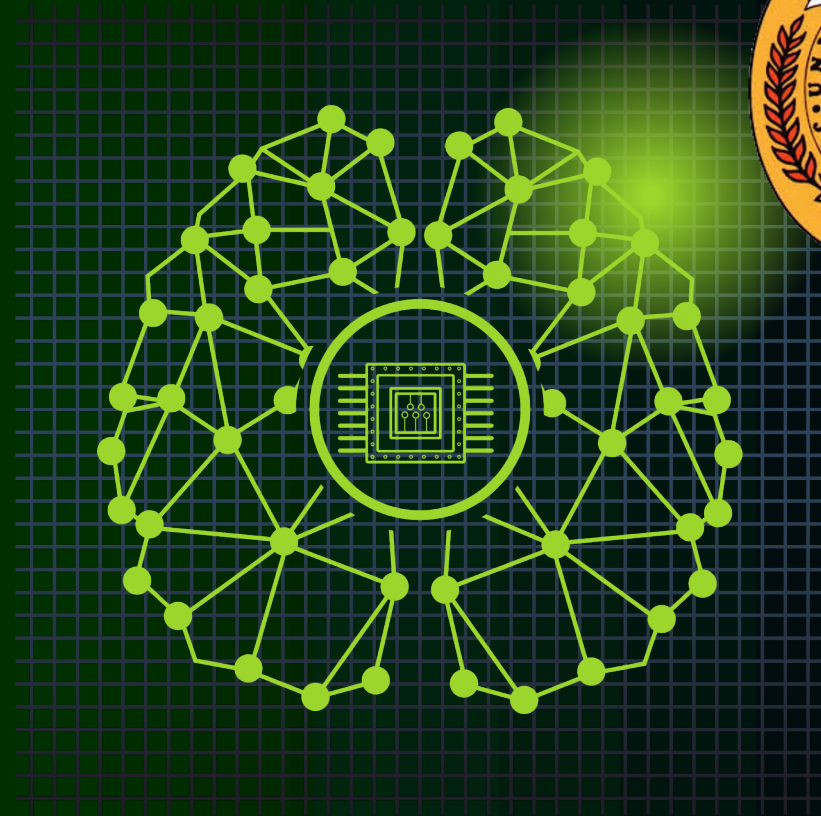
MACHINE LEARNING

ENDANG SRI RAHAYU



5

Regresi Linier



TUJUAN PEMBELAJARAN

Setelah mengikuti kuliah ini, mahasiswa diharapkan mampu:

- Memahami konsep dasar regresi linier dan fungsinya dalam prediksi.
- Menjelaskan komponen model linier (koefisien, intercept, error).
- Membangun dan melatih model regresi linier dengan Python.
- Mengevaluasi performa model menggunakan metrik umum (MAE, MSE, R^2).

Konsep Dasar Regresi Linear

Regresi Linier adalah metode untuk memprediksi nilai kontinu (angka) berdasarkan satu atau lebih variabel input (fitur).

Model ini membentuk garis lurus terbaik (best fit line) yang meminimalkan selisih antara nilai prediksi dan nilai sebenarnya.

Contoh:

Memprediksi harga rumah berdasarkan luas tanah.

Memprediksi gaji berdasarkan lama pengalaman kerja. Regresi linear

Persamaan umum:

$$y = wx + b$$

y : output yang diprediksi

x : input (fitur)

w : bobot (slope/kemiringan garis)

b : bias (intersep)



Analogi sederhana:

"Kita mencari garis lurus yang paling pas di antara kumpulan titik."

TUGAS

Dataset:	Luas Rumah (m ²)	Harga (juta)
	50	500
	70	650
	90	800
	120	1000

- Tampilkan grafik titik-titik data (misal: luas rumah vs harga rumah) dan garis regresi yang mendekati pola data.(Gambarkan grafik scatter plot.)
- Diskusikan: garis siapa yang paling mendekati pola data?
(Kalau Anda menggunakan Python atau Google Colab, bisa ditunjukkan dengan `sklearn.linear_model.LinearRegression` untuk demonstrasi visual.)

Contoh: Regresi Linear

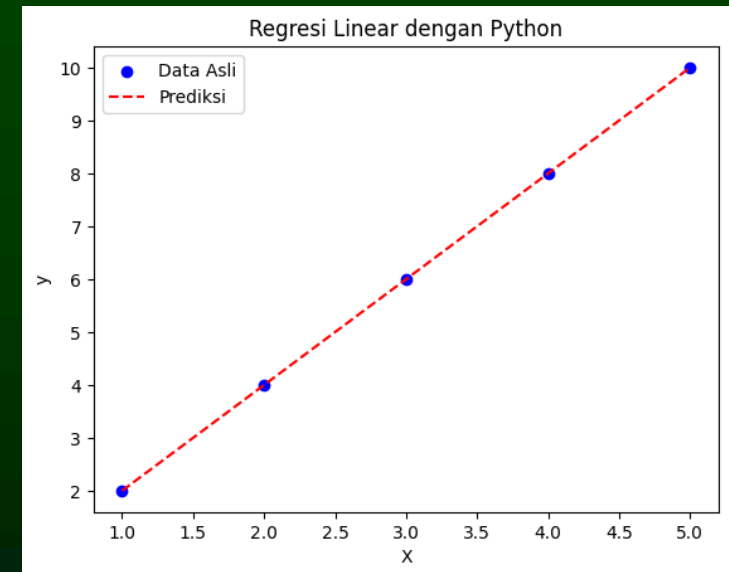
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Data contoh
X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
y = np.array([2, 4, 6, 8, 10])

# Membuat model regresi
model = LinearRegression()
model.fit(X, y)

# Membuat prediksi
y_pred = model.predict(X)

# Visualisasi
plt.scatter(X, y, color='blue', label='Data Asli')
plt.plot(X, y_pred, color='red', linestyle='--', label='Prediksi')
plt.legend()
plt.xlabel('X')
plt.ylabel('y')
plt.title('Regresi Linear dengan Python')
plt.show()
```



Apa maksudnya, jika saya menuliskan
`print(model.predict([[12]]))`

Tujuan Model: Minimasi Error

Untuk menemukan garis terbaik, model menghitung error (selisih) antara prediksi dan nilai sebenarnya.

Gunakan Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Model akan berusaha meminimalkan error ini agar prediksi mendekati data sebenarnya.

Contoh: Regresi Linear

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Data contoh: luas rumah dan harga
data = pd.DataFrame({
    'Luas': [50, 60, 70, 80, 90, 100],
    'Harga': [400, 500, 550, 650, 700, 800]
})

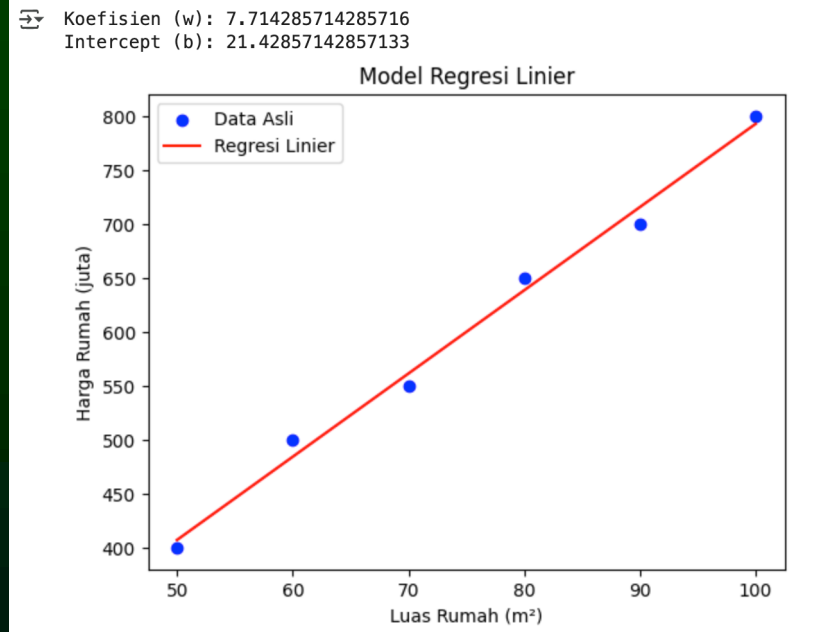
X = data[['Luas']]
y = data['Harga']

# Buat dan latih model
model = LinearRegression()
model.fit(X, y)

# Prediksi
prediksi = model.predict(X)

# Tampilkan hasil
print("Koefisien (w):", model.coef_[0])
print("Intercept (b):", model.intercept_)

# Visualisasi
plt.scatter(X, y, color='blue', label='Data Asli')
plt.plot(X, prediksi, color='red', label='Regresi Linier')
plt.xlabel('Luas Rumah (m²)')
plt.ylabel('Harga Rumah (juta)')
plt.title('Model Regresi Linier')
plt.legend()
plt.show()
```



Evaluasi Model

MSE (Mean Squared Error)

MAE (Mean Absolute Error)

R² (Coefficient of Determination) → menunjukkan seberapa baik garis regresi menjelaskan data.

```
▶ from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

mae = mean_absolute_error(y, prediksi)
mse = mean_squared_error(y, prediksi)
r2 = r2_score(y, prediksi)

print("MAE:", mae)
print("MSE:", mse)
print("R2:", r2)
```

```
⇒ MAE: 11.428571428571408
MSE: 142.85714285714263
R2: 0.9918367346938776
```

DISKUSI



"Mengapa regresi linear tidak selalu cocok untuk semua data?"
"Apa yang terjadi jika hubungan data tidak linier?"



Tugas



1. Carilah dataset sederhana (misalnya hubungan usia mobil dengan harga jual).
2. Buat scatter plot dan gambarkan garis regresi yang Anda perkirakan.
3. Jelaskan hubungan antara fitur dan target.
4. Implementasikan di Python menggunakan scikit-learn.

