

Nama : Namira Nurfaliani

NIM : 21120122140135

Kelas : Metode Numerik/C

## **Aplikasi Regresi untuk Mencari Hubungan Faktor yang Mempengaruhi Durasi Waktu Belajar dan Nilai Ujian Siswa Menggunakan Metode Linear dan Pangkat Sederhana**

[https://github.com/iranamira/metnum\\_pertemuan11\\_Namira-Nurfaliani](https://github.com/iranamira/metnum_pertemuan11_Namira-Nurfaliani)

Dalam analisis data dan pemodelan statistik, regresi adalah salah satu teknik yang paling penting dan sering digunakan untuk memahami hubungan antara variabel. Regresi digunakan untuk memprediksi nilai suatu variabel berdasarkan nilai variabel lain. Dalam konteks pendidikan, misalnya, kita mungkin ingin memahami bagaimana durasi waktu belajar (TB) mempengaruhi nilai ujian siswa (NT). Analisis ini bertujuan untuk membandingkan dua metode regresi yaitu, regresi linear dan regresi pangkat sederhana dalam memodelkan hubungan antara durasi waktu belajar dan nilai ujian. Dengan menggunakan dataset `Student_Performance.csv`, saya akan menerapkan kedua metode ini dengan menggunakan Bahasa pemrograman python untuk mengevaluasi kinerjanya berdasarkan galat RMS (Root Mean Squared Error), dan menganalisis hasilnya.

### **Regresi Linear**

$$\hat{y} = \theta_0 + \theta_1 x_1$$

Bentuk Umum Persamaan Regresi Linear

Regresi linear dapat diartikan sebagai penggunaan fungsi garis lurus sebagai acuan untuk memprediksi data. Pada persamaan di atas,  $y_{\text{topi}}$  merupakan variabel dependen dari nilai yang akan diprediksi, sementara  $x_1$  merupakan variabel independen atau variabel bebas atau prediktornya. Untuk menghasilkan garis yang tepat dengan error seminimal mungkin, harus menentukan nilai  $\theta_0$  dan  $\theta_1$  yang digunakan sebagai parameter.  $\theta_0$  merupakan sebuah intersep (intercept), sedangkan  $\theta_1$  merupakan gradien atau kemiringan garis.  $\theta_0$  dan  $\theta_1$  dapat disebut juga koefisien persamaan linear.

## Regresi Pangkat Sederhana

$$y = a \cdot x^b$$

Bentuk Umum Persamaan Regresi Pangkat Linear Sederhana

Regresi pangkat sederhana adalah teknik pemodelan statistik yang digunakan untuk menggambarkan hubungan non-linear antara dua variabel. Model ini mengasumsikan bahwa hubungan antara variabel independen  $x$  dan variabel dependen  $y$ . Dimana  $y$  adalah variable dependen (respon) dan  $x$  adalah variable independen (predictor).

## Implementasi Kode

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

# Impor data dari file CSV
file_path = 'Student_Performance.csv'
data = pd.read_csv(file_path)

# Extract relevant columns
TB = data['Hours Studied'].values
NT = data['Performance Index'].values

# Reshape TB for sklearn
TB_reshaped = TB.reshape(-1, 1)

# Model Linear (Metode 1)
linear_model = LinearRegression()
linear_model.fit(TB_reshaped, NT)
NT_pred_linear = linear_model.predict(TB_reshaped)

# Model Pangkat Sederhana (Metode 2)
#  $y = a \cdot x^b \Rightarrow \log(y) = \log(a) + b \cdot \log(x)$ 
log_TB = np.log(TB)
```

```

log_NT = np.log(NT)
pangkat_model = LinearRegression()
pangkat_model.fit(log_TB.reshape(-1, 1), log_NT)
a = np.exp(pangkat_model.intercept_)
b = pangkat_model.coef_[0]
NT_pred_pangkat = a * (TB ** b)

# Plot data dan hasil regresi
plt.figure(figsize=(14, 6))

# Plot Model Linear
plt.subplot(1, 2, 1)
plt.scatter(TB, NT, color='blue', label='Data Asli')
plt.plot(TB, NT_pred_linear, color='red', label='Regresi Linear')
plt.xlabel('Durasi Waktu Belajar (TB)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Regresi Linear')
plt.legend()

# Plot Model Pangkat Sederhana
plt.subplot(1, 2, 2)
plt.scatter(TB, NT, color='blue', label='Data Asli')
plt.plot(TB, NT_pred_pangkat, color='red', label='Regresi Pangkat Sederhana')
plt.xlabel('Durasi Waktu Belajar (TB)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Regresi Pangkat Sederhana')
plt.legend()

plt.tight_layout()
plt.show()

# Hitung galat RMS
rms_linear = np.sqrt(mean_squared_error(NT, NT_pred_linear))
rms_pangkat = np.sqrt(mean_squared_error(NT, NT_pred_pangkat))

print(f"RMS galat - Regresi Linear: {rms_linear}")
print(f"RMS galat - Model Pangkat Sederhana: {rms_pangkat}")

```

## Penjabaran Kode Program

### 1. Import Modul yang Dibutuhkan

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

- `import numpy` : untuk operasi numerik.
- `import pandas` : untuk manipulasi data.
- `import matplotlib.pyplot` : untuk plotting.
- `import LinearRegression` dari scikit-learn untuk membangun model regresi.
- `import mean_squared_error` dari scikit-learn untuk evaluasi model.

### 2. Memuat Dataset dari File CSV

```
file_path = 'Student_Performance.csv'
data = pd.read_csv(file_path)
```

- `file_path` : `'Student_Performance.csv'` yaitu ini mendefinisikan sebuah variabel `file_path` yang menyimpan nama atau jalur dari file CSV yang akan dibaca.
- `data` : menggunakan fungsi `read_csv` dari Pandas untuk membaca file CSV yang jalurnya disimpan di variabel `file_path`.

### 3. Ekstraksi Kolom Relevan

```
TB = data['Hours Studied'].values
NT = data['Performance Index'].values
```

- `TB` dan `NT` : Baris ini mengekstrak kolom `'Hours Studied'` (Durasi Waktu Belajar) dan `'Performance Index'` (Indeks Kinerja) dari `DataFrame` dan menyimpannya masing-masing dalam array NumPy `TB` dan `NT`.

### 4. Reshape TB untuk Sklearn

```
TB_reshaped = TB.reshape(-1, 1)
```

- `TB_reshaped` : Karena Scikit-learn mengharapkan input data berbentuk 2D (`n_samples, n_features`), array TB diubah bentuknya menjadi 2D menggunakan `reshape`.

## 5. Menggunakan Metode Linear

```
linear_model = LinearRegression()  
linear_model.fit(TB_reshaped, NT)  
NT_pred_linear = linear_model.predict(TB_reshaped)
```

Baris ini membuat model regresi linear, melatihnya menggunakan data `TB_reshaped` dan `NT`, dan kemudian memprediksi nilai `NT` berdasarkan model yang dilatih.

## 6. Membuat dan Melatih Model Regresi Pangkat Sederhana

```
log_TB = np.log(TB)  
log_NT = np.log(NT)  
pangkat_model = LinearRegression()  
pangkat_model.fit(log_TB.reshape(-1, 1), log_NT)  
a = np.exp(pangkat_model.intercept_)  
b = pangkat_model.coef_[0]  
NT_pred_pangkat = a * (TB ** b)
```

- Menghitung logaritma alami dari array `TB` (durasi waktu belajar) dan `NT` (nilai ujian).
- `pangkat_model.fit(log_TB.reshape(-1, 1), log_NT)` : Melatih model regresi linear menggunakan data yang telah ditransformasikan (`log_TB` dan `log_NT`). Data `log_TB` diubah menjadi bentuk kolom vektor (`n x 1`) untuk kompatibilitas dengan `sklearn`.
- `a` : dihitung dengan mengambil eksponensial dari intercept model regresi linear.
- `b` : koefisien model regresi linear (`coef_`).
- `NT_pred_pangkat` : Menghitung nilai prediksi `NT` menggunakan model pangkat

## 7. Plot Data dan Hasil Regresi

```
# Plot Model Linear
plt.subplot(1, 2, 1)
plt.scatter(TB, NT, color='blue', label='Data Asli')
plt.plot(TB, NT_pred_linear, color='red', label='Regresi Linear')
plt.xlabel('Durasi Waktu Belajar (TB)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Regresi Linear')
plt.legend()

# Plot Model Pangkat Sederhana
plt.subplot(1, 2, 2)
plt.scatter(TB, NT, color='blue', label='Data Asli')
plt.plot(TB, NT_pred_pangkat, color='red', label='Regresi Pangkat Sederhana')
plt.xlabel('Durasi Waktu Belajar (TB)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Regresi Pangkat Sederhana')
plt.legend()

plt.tight_layout()
plt.show()
```

- Subplot model linear : menunjukkan hasil dari model regresi linear. Data asli digambarkan sebagai titik biru, dan garis regresi linear digambarkan dalam warna merah.
- Subplot pangkat sederhana : menunjukkan hasil dari model regresi pangkat sederhana. Data asli digambarkan sebagai titik biru, dan kurva regresi pangkat sederhana digambarkan dalam warna merah.

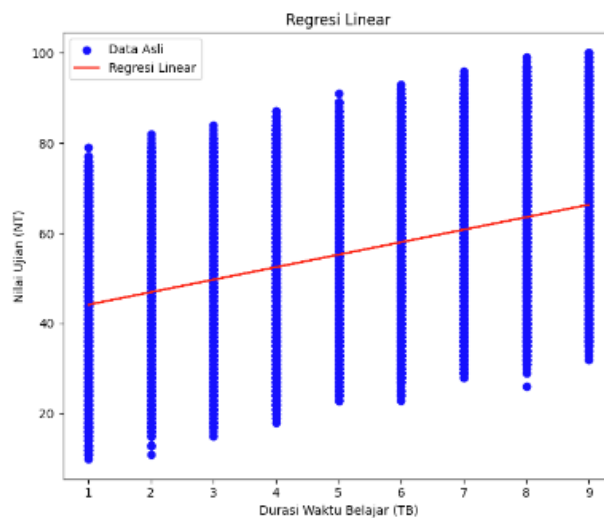
## 8. Hitung dan Cetak Ralat RMS

```
# Hitung galat RMS
rms_linear = np.sqrt(mean_squared_error(NT, NT_pred_linear))
rms_pangkat = np.sqrt(mean_squared_error(NT, NT_pred_pangkat))
```

```
print(f"RMS galat - Regresi Linear: {rms_linear}")
print(f"RMS galat - Model Pangkat Sederhana: {rms_pangkat}")
```

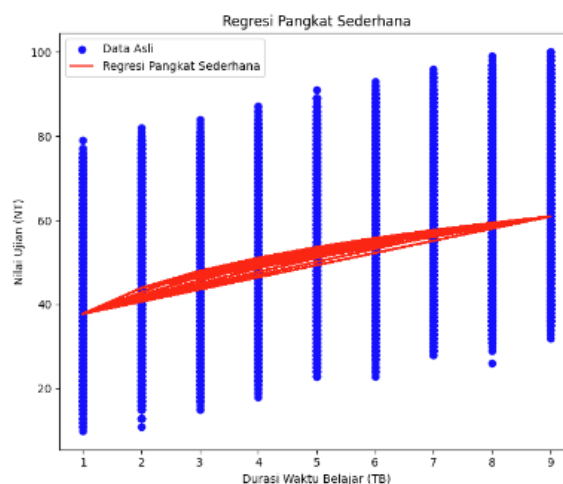
- Galat RMS (Root Mean Squared Error) : dihitung untuk kedua model untuk mengevaluasi performa mereka. Nilai galat RMS memberikan ukuran seberapa baik model memprediksi nilai ujian. Nilai galat RMS yang lebih rendah menunjukkan model yang lebih baik.

## Hasil Pengujian dengan Menggunakan Dua Metode



RMS galat - Regresi Linear: 17.819474832547773

Hasil RMS Galat Metode Regresi Linear



RMS galat - Model Pangkat Sederhana: 18.175837301582977

Hasil RMS Galat Metode Regresi Pangkat Sederhana

## Analisis Hasil Pengujian

### 1. Regresi Linear

- Menghasilkan garis lurus yang mencoba menyesuaikan pola data dengan cara paling sederhana. Garis ini tampaknya tidak sepenuhnya menangkap variasi dalam data, terutama jika terdapat hubungan non-linear antara durasi belajar dan nilai ujian.
- Memiliki RMS galat sebesar 17.819474832547773. Nilai ini menunjukkan tingkat kesalahan yang dihasilkan oleh model ketika memprediksi nilai ujian berdasarkan durasi belajar.

### 2. Regresi Pangkat Sederhana

- Menghasilkan kurva yang lebih baik dalam menangkap variasi dalam data yang mungkin memiliki hubungan non-linear. Ini terlihat dari bentuk kurva yang lebih fleksibel mengikuti distribusi titik-titik data.
- Memiliki RMS galat sedikit lebih tinggi, sebesar 18.175837301582977. Meskipun model ini lebih kompleks dan dapat menangkap pola non-linear, RMS galatnya tidak menunjukkan perbaikan signifikan dibandingkan regresi linear.

## Kesimpulan

1. Akurasi Prediksi: Regresi linear dan regresi pangkat sederhana menghasilkan nilai RMS galat yang relatif dekat, menunjukkan bahwa untuk dataset, model linear sederhana sudah cukup baik dalam melakukan prediksi.
2. Kompleksitas Model: Regresi pangkat sederhana lebih kompleks dan mampu menangkap pola data yang non-linear. Namun, peningkatan kompleksitas ini tidak menghasilkan penurunan RMS galat yang signifikan dibandingkan dengan regresi linear.
3. Pemilihan Model: Dalam konteks dataset, regresi linear mungkin lebih disarankan karena kesederhanaannya dan kemampuannya yang hampir sebanding dengan model pangkat dalam hal akurasi prediksi.