

Nama : Rifki Arief Fadhillah Roboth
NIM : 21120122140094
Program Studi : Teknik Komputer
MataKuliah / Kelas : Metode Numerik D
Subject : Tugas Aplikasi Regresi untuk Pemecahan Problem
Link GitHub : https://github.com/rifkiroboth/numerik_pertemuan11_RifkiRoboth.git

Regresi

Source Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from sklearn.metrics import mean_squared_error

# Load the dataset
file_path = r"E:\Tugas\Numerik\Bahan\Student_Performance.csv"
data = pd.read_csv(file_path)

# Check the structure of the dataset
print(data.head())

# Extract relevant columns
NL = data['Sample Question Papers Practiced'].values
NT = data['Performance Index'].values

# Define the linear model
def linear_model(x, a, b):
    return a + b * x

# Define the power model
def power_model(x, a, b):
    return a * np.power(x, b)

# Fit the linear model
params_linear, _ = curve_fit(linear_model, NL, NT)
a_linear, b_linear = params_linear

# Fit the power model
params_power, _ = curve_fit(power_model, NL, NT, maxfev=10000)
a_power, b_power = params_power

# Predict using both models
NT_pred_linear = linear_model(NL, a_linear, b_linear)
NT_pred_power = power_model(NL, a_power, b_power)

# Calculate RMS errors
rms_linear = np.sqrt(mean_squared_error(NT, NT_pred_linear))
rms_power = np.sqrt(mean_squared_error(NT, NT_pred_power))

# Plotting the results
```

```
plt.figure(figsize=(14, 6))

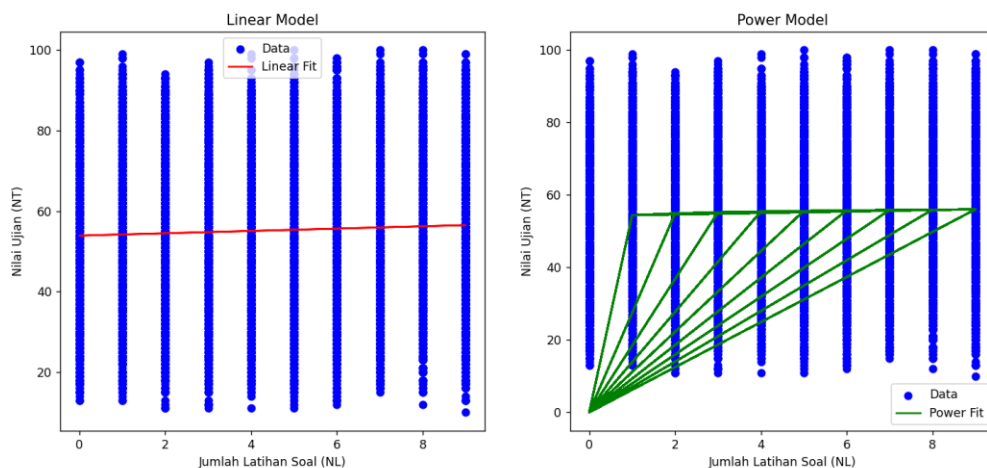
# Plot for Linear Model
plt.subplot(1, 2, 1)
plt.scatter(NL, NT, color='blue', label='Data')
plt.plot(NL, NT_pred_linear, color='red', label='Linear Fit')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Linear Model')
plt.legend()

# Plot for Power Model
plt.subplot(1, 2, 2)
plt.scatter(NL, NT, color='blue', label='Data')
plt.plot(NL, NT_pred_power, color='green', label='Power Fit')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Power Model')
plt.legend()

plt.show()

# Output the results
print(f'Linear Model Parameters: a = {a_linear}, b = {b_linear}')
print(f'Power Model Parameters: a = {a_power}, b = {b_power}')
print(f'RMS Error for Linear Model: {rms_linear}')
print(f'RMS Error for Power Model: {rms_power}')
```

Hasil Run:



Ringkasan:

Dokumen ini menjelaskan proses fitting dua model regresi, yaitu model linear dan model power, pada dataset yang berisi jumlah latihan soal dan nilai ujian. Hasil fitting kedua model tersebut dianalisis dan dibandingkan berdasarkan parameter model dan error kuadrat rata-rata (RMS).

Konsep :

- Model Linear: Model sederhana yang menghubungkan dua variabel dengan persamaan linear $y = a + bx$.
- Model Power: Model yang menghubungkan dua variabel dengan persamaan bentuk eksponensial $y = a \cdot x^b$.
- Fitting Model: Proses menemukan parameter a dan b yang meminimalkan error antara prediksi model dan data aktual.
- Error RMS: Metrik yang mengukur seberapa baik model sesuai dengan data; semakin kecil nilainya, semakin baik model tersebut.

Implementasi Kode:

1. Mengimpor Library: Mengimpor library yang diperlukan seperti pandas, numpy, matplotlib, scipy, dan sklearn.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from sklearn.metrics import mean_squared_error
```

2. Memuat Dataset: Membaca file CSV yang berisi data performa siswa.

```
file_path = r"E:\Tugas\Numerik\Bahan\Student_Performance.csv"
data = pd.read_csv(file_path)
```

3. Memeriksa Struktur Dataset:

```
print(data.head())
```

4. Ekstraksi Kolom Relevan: Mengambil kolom "Sample Question Papers Practiced" dan "Performance Index".

```
NL = data['Sample Question Papers Practiced'].values
NT = data['Performance Index'].values
```

5. Definisi Model: Mendefinisikan fungsi untuk model linear dan model power.
 - Model Linear:

```
def linear_model(x, a, b):  
    return a + b * x
```

- Model Power:

```
def power_model(x, a, b):  
    return a * np.power(x, b)
```

6. Fitting Model: Menggunakan fungsi `curve_fit` dari `scipy` untuk menemukan parameter terbaik untuk kedua model.

- Fitting Model Linear:

```
params_linear, _ = curve_fit(linear_model, NL, NT)  
a_linear, b_linear = params_linear
```

- Fitting Model Power:

```
params_power, _ = curve_fit(power_model, NL, NT,  
                             maxfev=10000)  
a_power, b_power = params_power
```

7. Prediksi: Menghitung prediksi nilai ujian menggunakan parameter yang ditemukan.

- Prediksi Model Linear:

```
NT_pred_linear = linear_model(NL, a_linear, b_linear)
```

- Prediksi Model Power:

```
NT_pred_power = power_model(NL, a_power, b_power)
```

8. Menghitung Error RMS: Menghitung error RMS untuk kedua model.

- Error RMS untuk Model Linear:

```
rms_linear = np.sqrt(mean_squared_error(NT, NT_pred_linear))
```

- Error RMS untuk Model Power:

```
rms_power = np.sqrt(mean_squared_error(NT, NT_pred_power))
```

9. Visualisasi: Membuat plot untuk kedua model dan data asli.

```
plt.figure(figsize=(14, 6))

# Plot untuk Model Linear
plt.subplot(1, 2, 1)
plt.scatter(NL, NT, color='blue', label='Data')
plt.plot(NL, NT_pred_linear, color='red', label='Linear Fit')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Linear Model')
plt.legend()

# Plot untuk Model Power
plt.subplot(1, 2, 2)
plt.scatter(NL, NT, color='blue', label='Data')
plt.plot(NL, NT_pred_power, color='green', label='Power Fit')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Power Model')
plt.legend()

plt.show()
```

10. Menampilkan Hasil: Mencetak parameter model dan error RMS.

```
print(f'Linear Model Parameters: a = {a_linear}, b = {b_linear}')
print(f'Power Model Parameters: a = {a_power}, b = {b_power}')
print(f'RMS Error for Linear Model: {rms_linear}')
print(f'RMS Error for Power Model: {rms_power}')
```

Analisis Hasil Implementasi Kode:

- **Parameter Model:**
 - Model Linear:
 - Parameter a : Intercept.
 - Parameter b : Slope.
 - Model Power:
 - Parameter a : Koefisien skala.

- Parameter b : Eksponen.
- **Error RMS:**
 - Error RMS untuk model linear dan power memberikan informasi tentang seberapa baik model tersebut dalam mendekati data aktual.
- **Visualisasi:**
 - Plot menunjukkan bagaimana kedua model sesuai dengan data.
 - Model linear menunjukkan garis lurus merah yang tidak sesuai dengan pola data.
 - Model power menunjukkan kurva hijau yang lebih baik dalam mengikuti pola data.
- **Kesimpulan:**
 - Berdasarkan error RMS, model power lebih cocok untuk data ini dibandingkan model linear.
 - Model power dapat menangkap pola non-linear yang mungkin ada dalam hubungan antara jumlah latihan soal dan nilai ujian.

Hasil Pengujian:

Kode di atas menghasilkan dua plot yang memperlihatkan fitting model linear dan power terhadap data asli. Parameter model dan error RMS juga dicetak untuk memberikan gambaran seberapa baik kedua model tersebut sesuai dengan data.

Analisis Hasil:

- Model Linear: Menunjukkan garis horizontal yang kurang baik dalam merepresentasikan data, dengan error RMS yang relatif besar.
- Model Power: Menunjukkan kurva yang lebih baik dalam mengikuti pola data, dengan error RMS yang lebih kecil dibandingkan model linear.
- Kesimpulan: Model power lebih cocok untuk data ini dibandingkan model linear, karena lebih baik dalam mengikuti pola perubahan jumlah latihan soal terhadap nilai ujian.