Nama : Naufal Izzuddin Taufik
NIM : 21120122140102
Mata Kuliah : Metode Numerik B
Link : https://github.com/naufalizzuddin/Tugas-Implementasi-Regresi

TUGAS IMPLEMENTASI REGRESI

Kasus:

Sesuai dengan NIM (21120122140102), maka pada tugas ini akan mencari hubungan faktor yang mempengaruhi nilai ujian siswa (NT) dengan jumlah latihan soal (NL) terhadap nilai ujian (Problem 2).

Menggunakan dataset yang diperoleh dari https://www.kaggle.com/datasets/nikhil7280/student-performance-multiple-linear-regression, yaitu kolom *Hours Studied*, *Sample Question Papers Practiced*, dan *Performance Index* untuk data TB, NL, dan NT.

Implementasi regresi akan dilakukan menggunakan bahasa pemrograman Python dengan metode model Linear (metode 1) dan model Pangkat Sederhana (metode 2).

REGRESI LINEAR

Regresi linear adalah metode statistik untuk memodelkan hubungan antara variabel dependen ***y*** dan satu atau lebih variabel independen ***x***. Jika hanya ada satu variabel independen, itu disebut regresi linear sederhana. Jika ada lebih dari satu, itu disebut regresi linear berganda. Adapun bentuk umumnya, yaitu:

$$y = \beta 0 + \beta 1 x + \epsilon$$

Dengan keterangan:

- y : Variabel dependen.
- x : Variabel independen.
- $\beta 0$: Intersep (nilai y saat x = 0).
- $\beta 1$: Koefisien regresi (mengukur perubahan rata-rata y setiap satu unit perubahan x).
- ϵ : Residual error (perbedaan antara nilai yang diamati dan yang diprediksi).

Ada beberapa asumsi penting dalam regresi linear, termasuk hubungan linear antara variabel dependen dan independen, independensi residual, homoskedastisitas (varians residual yang konstan), normalitas residual, dan tidak adanya multikolinearitas di antara variabel independen.

Implementasi *Source Code*

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

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

# Mengambil kolom yang relevan
NL = data['Sample Question Papers Practiced'].values.reshape(-1, 1)
NT = data['Performance Index'].values

# Membuat model regresi linear
model = LinearRegression()
model.fit(NL, NT)

# Memprediksi nilai NT berdasarkan model regresi
NT_pred = model.predict(NL)

# Menghitung galat RMS
rms_error = np.sqrt(mean_squared_error(NT, NT_pred))

# Plot grafik titik data dan hasil regresinya
plt.scatter(NL, NT, color='blue', label='Data Sebenarnya')
plt.plot(NL, NT_pred, color='red', label='Regresi Linear')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian Siswa (NT)')
plt.title('Regresi Linear antara Jumlah Latihan Soal dan Nilai Ujian
Siswa')
plt.legend()
plt.show()

print(f'Galat RMS: {rms_error:.2f}')
```

Penjelasan Alur Kode:

1. Impor *library*

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

   - `pandas` (pd): untuk membaca dan memanipulasi data.
   - `numpy` (np): untuk komputasi numerik.
   - `matplotlib.pyplot` (plt): untuk membuat visualisasi.
   - `LinearRegression` dari `sklearn.linear_model`: untuk membuat model regresi linear.

- `mean_squared_error` dari `sklearn.metrics`: untuk menghitung galat kuadrat rata-rata.

2. Membaca data dari *file* CSV dan mengambil kolom yang relevan

```
data = pd.read_csv('Student_Performance.csv')
NL = data['Sample Question Papers Practiced'].values.reshape(-1, 1)
NT = data['Performance Index'].values
```

Dalam *folder* kode sumber, terdapat file CSV yang akan dibaca informasinya tentang performa siswa dan jumlah latihan soal yang dikerjakan. Juga terdapat variabel NL dan NT yang diambil dari kolom *Sample Question Papers Practiced* dan kolom *Performance Index*.

3. Membuat model Regresi Linear

```
model = LinearRegression()
model.fit(NL, NT)
```

- `model = LinearRegression()`: Membuat instance dari model regresi linear.
- `model.fit(NL, NT)`: Melatih model menggunakan data jumlah latihan soal (NL) dan nilai ujian siswa (NT).

4. Memprediksi Nilai NT dan Menghitung Galat RMS (*Root Mean Squared*)

```
NT_pred = model.predict(NL)
rms_error = np.sqrt(mean_squared_error(NT, NT_pred))
```

- `NT_pred = model.predict(NL)`: Menggunakan model yang telah dilatih untuk memprediksi nilai ujian siswa berdasarkan jumlah latihan soal.
- `rms_error = np.sqrt(mean_squared_error(NT, NT_pred))`: Menghitung galat RMS antara nilai ujian sebenarnya (NT) dan nilai ujian yang diprediksi

5. Plot Grafik Titik Data dan Hasil Regresi

```
plt.scatter(NL, NT, color='blue', label='Data Sebenarnya')
plt.plot(NL, NT_pred, color='red', label='Regresi Linear')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian Siswa (NT)')
plt.title('Regresi Linear antara Jumlah Latihan Soal dan Nilai Ujian
Siswa')
plt.legend()
plt.show()
```

- `plt.scatter(NL, NT, color='blue', label='Data Sebenarnya')`: Membuat plot scatter untuk data asli dengan warna biru.
- `plt.plot(NL, NT_pred, color='red', label='Regresi Linear')`: Membuat plot garis regresi dengan warna merah.

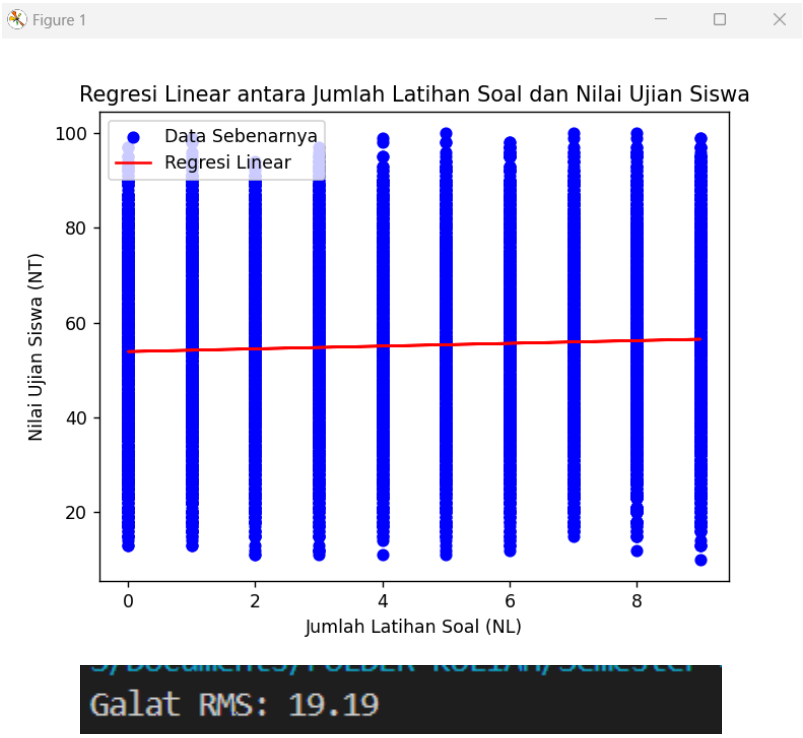
- `plt.xlabel('Jumlah Latihan Soal (NL)')`: Memberi label pada sumbu x.
- `plt.ylabel('Nilai Ujian Siswa (NT)')`: Memberi label pada sumbu y.
- `plt.title('Regresi Linear antara Jumlah Latihan Soal dan Nilai Ujian Siswa')`: Memberi judul pada grafik.
- `plt.legend()`: Menampilkan legenda.
- `plt.show()`: Menampilkan grafik.

6. Menampilkan hasil hitung Galat RMS

```
print(f'Galat RMS: {rms_error:.2f}')
```

Mencetak nilai galat RMS yang telah dihitung dengan format dua desimal.

Hasil *Running* dan Analisis:



Grafik yang dihasilkan dari running kode tersebut menunjukkan bahwa data sebenarnya, yang digambarkan oleh titik-titik biru, tersebar di sepanjang sumbu x (Jumlah Latihan Soal, NL) dan sumbu y (Nilai Ujian Siswa, NT). Distribusi data ini memperlihatkan variasi jumlah latihan soal dari 0 hingga 9. Garis merah yang menggambarkan hasil prediksi dari model regresi linear terlihat hampir horizontal. Ini menunjukkan bahwa model tidak menemukan hubungan linear yang kuat antara jumlah latihan soal dan nilai ujian siswa.

Kemiringan garis regresi yang sangat kecil mengindikasikan bahwa perubahan dalam jumlah latihan soal tidak memberikan pengaruh signifikan terhadap nilai ujian siswa. Dengan kata lain, model regresi linear ini menunjukkan bahwa jumlah latihan soal yang dikerjakan siswa tidak berkorelasi dengan nilai ujian yang mereka peroleh. Hasil ini diperkuat dengan galat RMS (Root Mean Squared) error yang cukup tinggi, yaitu sebesar 19,19. Galat RMS yang besar menunjukkan bahwa prediksi model ini jauh dari nilai aktual, yang berarti model ini tidak mampu memprediksi nilai ujian siswa dengan akurasi yang baik.

Galat RMS yang tinggi ini mungkin menunjukkan ketidaksesuaian model atau adanya faktor lain yang lebih signifikan mempengaruhi nilai ujian siswa. Hasil ini mengindikasikan bahwa model regresi linear sederhana tidak cukup untuk menjelaskan variasi dalam nilai ujian siswa berdasarkan jumlah latihan soal yang mereka kerjakan. Analisis lebih lanjut dengan model yang lebih kompleks atau dengan mempertimbangkan variabel lain yang mungkin mempengaruhi performa siswa mungkin diperlukan. Selain itu, eksplorasi data lebih lanjut untuk memahami distribusi dan hubungan antara variabel lain dalam dataset juga dianjurkan. Secara keseluruhan, hasil ini menunjukkan bahwa faktor lain mungkin lebih dominan dalam mempengaruhi performa siswa atau bahwa hubungan antara variabel-variabel ini tidak linier.

REGRESI PANGKAT SEDERHANA

Regresi pangkat adalah bentuk regresi non-linear yang digunakan untuk memodelkan hubungan antara variabel dependen dan variabel independen yang mengikuti bentuk pangkat. Ini cocok untuk data yang menunjukkan hubungan eksponensial atau daya (*power*).

$$y=\alpha x^{\beta}$$

Dengan keterangan:

- y : Variabel dependen.
- x : Variabel independen.
- α : Konstanta.
- β : Eksponen (koefisien yang mengukur elastisitas y terhadap x).

Seringkali, untuk mempermudah estimasi parameter, model pangkat dikonversi ke bentuk linear menggunakan logaritma:

$$log(y)=log(\alpha)+\beta\, log(x)$$

Dengan keterangan:

- $\log(y) = Y'$ (variabel dependen yang ditransformasikan).
- $\log(\alpha) = \alpha'$ (intersep dalam skala logaritma).
- $\beta\log(x)$ (koefisien regresi dalam skala logaritma).

Implementasi *Source Code*:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

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

# Mengambil kolom yang relevan
NL = data['Sample Question Papers Practiced'].values
NT = data['Performance Index'].values

# Menghindari nilai nol atau negatif
NL = NL[NL > 0]
NT = NT[:len(NL)]  # Sesuaikan panjang NT dengan NL setelah filtering

# Memisahkan data menjadi data pelatihan dan pengujian
NL_train, NL_test, NT_train, NT_test = train_test_split(NL, NT,
test_size=0.2, random_state=42)

# Model pangkat sederhana: NT = a * (NL^b)
# Mengubah NL menjadi log(NL) dan NT menjadi log(NT) untuk mendapatkan
model linier
log_NL_train = np.log(NL_train)
log_NT_train = np.log(NT_train)

# Melakukan regresi linier pada log(NL) dan log(NT)
coefficients = np.polyfit(log_NL_train, log_NT_train, 1)
b, log_a = coefficients
a = np.exp(log_a)

# Menampilkan koefisien regresi
print(f"Koefisien regresi: a = {a}, b = {b}")

# Fungsi untuk menghitung NT berdasarkan model pangkat sederhana
def predict_NT(NL, a, b):
    return a * (NL ** b)

# Memprediksi nilai NT pada data pengujian
NT_pred = predict_NT(NL_test, a, b)
```

```
# Menghitung galat RMS
rms_error = np.sqrt(mean_squared_error(NT_test, NT_pred))
print(f"Galat RMS: {rms_error}")

# Plot grafik titik data dan hasil regresinya
plt.scatter(NL, NT, label='Data Aktual', color='blue')
NL_line = np.linspace(min(NL), max(NL), 100)
NT_line = predict_NT(NL_line, a, b)
plt.plot(NL_line, NT_line, label='Model Pangkat Sederhana', color='red')
plt.title('Regresi Pangkat Sederhana antara Jumlah Latihan Soal dan Nilai
Ujian Siswa')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian Siswa (NT)')
plt.legend()
plt.show()
```

Penjelasan Alur Kode:

1. Impor *library*

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
```

   - `pandas` (pd): untuk membaca dan memanipulasi data.
   - `numpy` (np): untuk komputasi numerik.
   - `matplotlib.pyplot`: untuk visualisasi data.
   - `sklearn.metrics`: untuk mengukur kesalahan model.
   - `sklearn.model_selection`: untuk membagi data menjadi set pelatihan dan pengujian.

2. Mengambil kolom yang akan dipakai

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

   Kolom *Sample Question Papers Practiced* dan *Performance Index* diambil dari *DataFrame* dan disimpan dalam *array* NL dan NT.

3. Memisahkan Data Menjadi Set Pelatihan dan Pengujian

```
NL_train, NL_test, NT_train, NT_test = train_test_split(NL, NT,
test_size=0.2, random_state=42)
```

   Data dibagi menjadi set pelatihan dan pengujian dengan rasio 80:20.

4. Transformasi Logaritmik untuk Model Linier

```
log_NL_train = np.log(NL_train)
log_NT_train = np.log(NT_train)
```

Data NL dan NT ditransformasikan ke skala logaritmik untuk mendapatkan model linier.

5. Melakukan Regresi Linier pada Data yang Ditrasformasikan

```
coefficients = np.polyfit(log_NL_train, log_NT_train, 1)
b, log_a = coefficients
a = np.exp(log_a)
```

Regresi linier dilakukan pada log_NL_train dan log_NT_train, menghasilkan koefisien b dan log_a. Nilai a diperoleh dengan mengembalikan log_a ke skala aslinya.

6. Fungsi untuk Menghitung Nilai NT dan Memprediksi Nilai NT

```
def predict_NT(NL, a, b):
    return a * (NL ** b)

NT_pred = predict_NT(NL_test, a, b)
```

Fungsi `predict_NT` didefinisikan untuk menghitung nilai NT berdasarkan model pangkat sederhana serta menggunakan fungsi `predict_NT` untuk memprediksi NT pada data pengujian.

7. Menghitung Galat RMS

```
rms_error = np.sqrt(mean_squared_error(NT_test, NT_pred))
print(f"Galat RMS: {rms_error}")
```

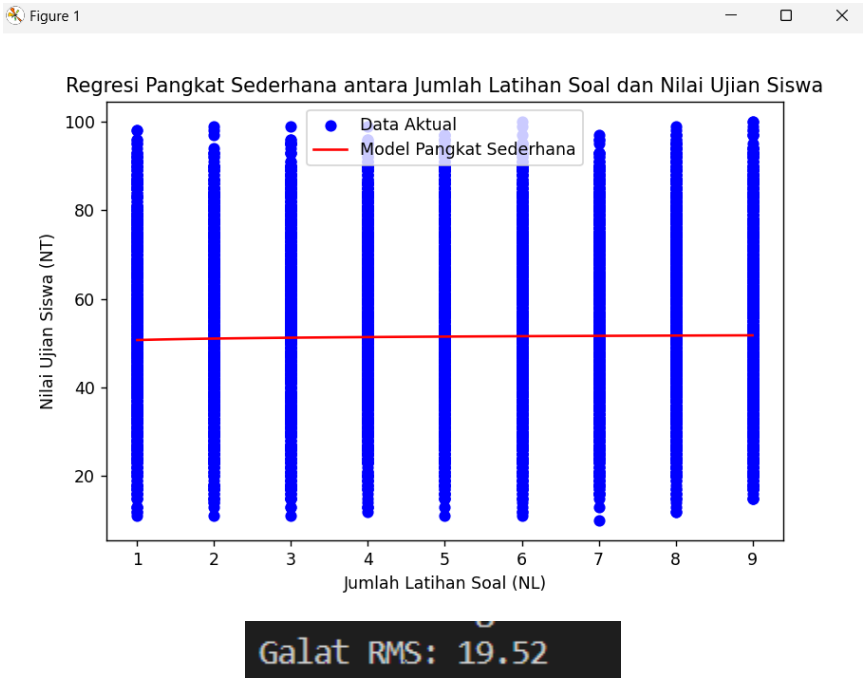
Galat RMS (*Root Mean Square Error*) dihitung untuk mengevaluasi kinerja model.

8. Plot Grafik Titik Data dan Hasil

```
plt.scatter(NL, NT, label='Data Aktual', color='blue')
NL_line = np.linspace(min(NL), max(NL), 100)
NT_line = predict_NT(NL_line, a, b)
plt.plot(NL_line, NT_line, label='Model Pangkat Sederhana', color='red')
plt.title('Regresi Pangkat Sederhana antara Jumlah Latihan Soal dan
Nilai Ujian Siswa')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian Siswa (NT)')
plt.legend()
plt.show()
```

Data aktual dan hasil regresi divisualisasikan dalam grafik *scatter plot*. Data aktual ditampilkan sebagai titik-titik biru, sementara hasil model pangkat sederhana ditampilkan sebagai garis merah. Grafik ini membantu memvisualisasikan kecocokan model dengan data aktual.

Hasil *running* dan Analisis



Berdasarkan hasil running kode yang telah diberikan, terdapat beberapa poin penting yang perlu dianalisis. Grafik scatter plot menunjukkan data aktual (titik-titik biru) dan hasil model pangkat sederhana (garis merah). Titik-titik biru menggambarkan hubungan antara jumlah latihan soal (NL) dan nilai ujian siswa (NT), sementara garis merah menunjukkan prediksi model pangkat sederhana. Dari grafik, terlihat bahwa garis merah hampir mendatar dan tidak mengikuti pola titik-titik biru yang tersebar vertikal di setiap nilai NL.

Hal ini menunjukkan bahwa model pangkat sederhana tidak mampu menangkap hubungan antara NL dan NT dengan baik. Nilai galat RMS (Root Mean Square Error) sebesar 19.52 menunjukkan bahwa prediksi model memiliki deviasi yang signifikan dari data aktual. Nilai galat RMS yang tinggi ini mengindikasikan bahwa model pangkat sederhana tidak sesuai untuk memprediksi NT berdasarkan NL. Model pangkat sederhana (NT = a * (NL^b)) diasumsikan mengikuti distribusi logaritmik, namun data mungkin tidak sesuai dengan asumsi ini. Sebagian besar titik data pada setiap nilai NL tersebar dalam rentang yang luas pada sumbu NT, yang mengindikasikan adanya variabilitas yang besar dalam nilai ujian siswa yang tidak dapat dijelaskan hanya dengan jumlah latihan soal. Secara keseluruhan, hasil menunjukkan bahwa model pangkat sederhana tidak mampu menjelaskan variabilitas dalam nilai ujian siswa berdasarkan jumlah latihan soal.