TUGAS METODE NUMERIK APLIKASI REGRESI

Irfan Maulana Manaf
21120122140097
https://github.com/BroManaf/Aplikasi-Regresi_Irfan-Manaf_Metode-Numerik/tree/main

Diinginkan untuk mencari hubungan faktor yang mempengaruhi nilai ujian siswa (NT):

1. Durasi waktu belajar (TB) terhadap nilai ujian (Problem 1)
2. Jumlah latihan soal (NL) terhadap nilai ujian (Problem 2)

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

Implementasikan regresi untuk mencari hubungan tersebut menggunakan metode:

1. Model linear (Metode 1)
2. Model pangkat sederhana (Metode 2)
3. Model eksponensial (Metode 3)
4. Model lainnnya di halaman 24 slide materi (Metode opsional)

Tugas mahasiswa:

1. Mahasiswa membuat **kode sumber** dengan bahasa pemrograman yang dikuasai untuk mengimplementasikan solusi di atas, dengan ketentuan:
    - NIM terakhir % 4 = 0 mengerjakan Problem 1 dengan Metode 1 dan Metode 2
    - NIM terakhir % 4 = 1 mengerjakan Problem 1 dengan Metode 1 dan Metode 3
    - NIM terakhir % 4 = 2 mengerjakan Problem 2 dengan Metode 1 dan Metode 2
    - NIM terakhir % 4 = 3 mengerjakan Problem 2 dengan Metode 1 dan Metode 3
    - Mahasiswa juga bisa menambah solusi dengan salah satu metode opsional
2. Sertakan **kode testing** untuk menguji kode sumber tersebut untuk menyelesaikan problem dalam gambar. Plot grafik titik data dan hasil regresinya masing-masing
3. Hitung galat RMS dari tiap metode yang digunakan
4. Mengunggah kode sumber tersebut ke Github dan **setel sebagai publik**. Berikan deskripsi yang memadai dari project tersebut. Masukkan juga dataset dan data hasil di repositori tersebut.
5. Buat dokumen docx dan pdf yang menjelaskan alur kode dari (1), analisis hasil, dan penjabarannya. Sistematika dokumen: Ringkasan, Konsep, Implementasi Kode, Hasil Pengujian, dan Analisis Hasil.

Maka, dalam tugas ini, saya akan mencari hubungan antara jumlah latihan soal (NL) dengan nilai ujian siswa (NT) menggunakan regresi eksponensial dan linear. Data diperoleh dari dataset "student-performance-multiple-linear-regression" yang mengandung kolom "Sample Question Papers Practiced" dan "Performance Index". Kita akan mengimplementasikan solusi ini dengan Python, memplot hasil regresi, menghitung galat RMS, dan memberikan penjelasan mendetail tentang alur kode dan hasil analisis.

MODEL LINEAR

Regresi linier adalah metode statistik untuk memodelkan hubungan antara variabel dependen Y dan satu atau lebih variabel independen X. Dalam kasus regresi linier sederhana, hanya ada satu variabel independen. Model ini bertujuan untuk menemukan garis lurus (linear) yang paling baik menggambarkan hubungan antara X dan Y. Adapun rumus dari regresi linear sederhana ini adalah:

$$Y = \beta_0 + \beta_1 X$$

Dengan keterangan:

- Y : Variabel dependen (nilai ujian, NT)
- X : Variabel independen (jumlah latihan soal, NL)
- $\beta 0$: Intercept (nilai Y saat X=0)
- $\beta 1$: Slope atau kemiringan garis regresi (perubahan rata-rata pada Y untuk setiap unit perubahan pada X)

Metode model linear (regresi linier) digunakan dalam persoalan ini untuk memodelkan dan memprediksi nilai ujian siswa berdasarkan jumlah latihan soal yang mereka kerjakan. Hasilnya membantu memahami apakah ada hubungan linier antara dua variabel ini dan seberapa kuat hubungan tersebut.

PENERAPAN METODE MODEL LINEAR MENGGUNAKAN PYTHON TERHADAP SOAL

Secara keseluruhan, alur kode adalah:

1. Data Preparation: Mengambil data yang diperlukan dari dataset.
2. Model Training: Melatih model regresi linier dengan data training.
3. Prediction: Menggunakan model untuk memprediksi hasil pada data testing.
4. Evaluation: Mengukur akurasi prediksi dengan RMS error.
5. Visualization: Melihat hubungan data sebenarnya dan garis regresi.

```
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
from sklearn.model_selection import train_test_split

# Membaca dataset dari file lokal
file_path = 'Student_Performance (1).csv'  # Gantilah path ini dengan
lokasi sebenarnya dari file dataset Anda
data = pd.read_csv(file_path)

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

# Membagi data menjadi training dan testing set
NL_train, NL_test, NT_train, NT_test = train_test_split(NL, NT,
test_size=0.2, random_state=42)
```

```
# Membuat model regresi linier
model = LinearRegression()
model.fit(NL_train, NT_train)

# Prediksi
NT_pred_train = model.predict(NL_train)
NT_pred_test = model.predict(NL_test)

# Menghitung RMS error
rms_error_train = np.sqrt(mean_squared_error(NT_train, NT_pred_train))
rms_error_test = np.sqrt(mean_squared_error(NT_test, NT_pred_test))

# Plot grafik titik data dan hasil regresi
plt.figure(figsize=(10, 6))
plt.scatter(NL, NT, color='blue', label='Data Sebenarnya')
plt.plot(NL, model.predict(NL), color='red', label='Garis Regresi')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Regresi Linier Jumlah Latihan Soal terhadap Nilai Ujian')
plt.legend()
plt.show()

# Menampilkan hasil RMS error
print(f'RMS Error (Training set): {rms_error_train}')
print(f'RMS Error (Testing set): {rms_error_test}')
```

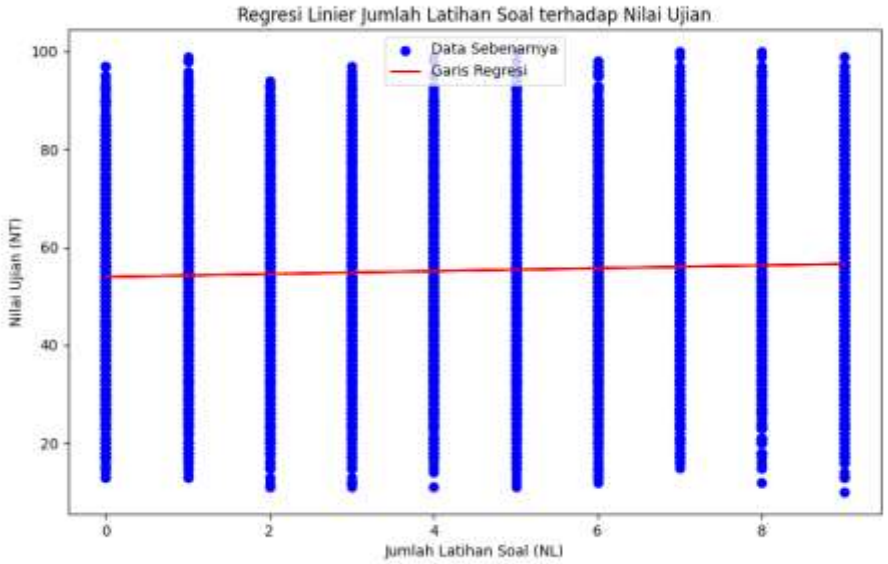
langkah-langkahnya secara detail:

1. **Import library:**
   Mengimpor pustaka yang diperlukan seperti pandas untuk manipulasi data, numpy untuk operasi numerik, matplotlib untuk visualisasi, dan scikit-learn untuk membangun model regresi linier serta menghitung error dan membagi dataset.
2. **Membaca dataset:**
   file_path: Menentukan path file lokal tempat dataset disimpan dan pd.read_csv(file_path): Membaca dataset dari file CSV yang terletak pada path yang ditentukan.
3. **Mengambil kolom yang relevan:**
   Untuk NL Mengambil kolom 'Sample Question Papers Practiced' dan mereshape-nya menjadi array 2D dengan satu kolom (karena sklearn mengharapkan input dalam bentuk ini) dan untuk NT Mengambil kolom 'Performance Index' sebagai array 1D.
4. **Membagi data menjadi training dan testing set:**
   train_test_split: Membagi dataset menjadi dua bagian, 80% untuk training dan 20% untuk testing, menggunakan random_state untuk memastikan pembagian yang sama setiap kali kode dijalankan.
5. **Membuat model regresi linier:**
   LinearRegression() untuk Membuat instance model regresi linier dan model.fit(NL_train, NT_train) untuk Melatih model menggunakan data training.

6. **Melakukan prediksi:**
   Gunakan model.predict(NL_train) untuk memprediksi nilai ujian pada data training dan model.predict(NL_test) untuk memprediksi nilai ujian pada data testing.
7. **Menghitung RMS Error:**
   mean_squared_error untuk menghitung rata-rata kesalahan kuadrat (MSE) antara nilai sebenarnya dan nilai prediksi dan np.sqrt: Menghitung akar kuadrat dari MSE untuk mendapatkan RMS error.
8. **Memvisualisasikan hasil.**
   - plt.figure(figsize=(10, 6)) untuk membuat figure untuk plot dengan ukuran tertentu.
   - plt.scatter(NL, NT, color='blue', label='Data Sebenarnya') untuk membuat scatter plot dari data sebenarnya.
   - plt.plot(NL, model.predict(NL), color='red', label='Garis Regresi') untuk menambahkan garis regresi berdasarkan prediksi model.
   - plt.xlabel, plt.ylabel, plt.title untuk memberi label pada sumbu dan judul pada grafik.
   - plt.legend() untuk menampilkan legenda pada grafik.
   - plt.show() untuk menampilkan grafik.
9. **Menampilkan hasil RMS Error.**
   Print (Menampilkan RMS error untuk training set dan testing set pada konsol)

Adapun outputnya adalah seperti berikut:

```
RMS Error (Training set): 19.182357989334243
RMS Error (Testing set): 19.23932015127285
```



Penjelasan terkait output:
Gambar yang ditampilkan adalah hasil regresi linier yang menunjukkan hubungan antara jumlah latihan soal (NL) dan nilai ujian (NT) siswa. Adapun penjelasan dari masing-masing output pada gambar adalah:

1. **Scatter Plot (Titik Biru)**

Setiap titik biru pada grafik mewakili pasangan nilai dari dataset, yaitu jumlah latihan soal yang dilakukan siswa (NL) dan nilai ujian mereka (NT). Titik-titik biru tersebar di sepanjang sumbu horizontal (NL) dari 0 hingga 9, yang menunjukkan berbagai jumlah latihan soal yang dilakukan siswa. Titik-titik biru tersebar di sepanjang sumbu vertikal (NT) dari 0 hingga 100, yang menunjukkan berbagai nilai ujian siswa.

2. **Garis Regresi (Garis Merah)**
   Garis merah adalah garis regresi linier yang dihasilkan dari metode model regresi linier. Garis ini mewakili prediksi nilai ujian (NT) berdasarkan jumlah latihan soal (NL). Garis merah yang hampir datar *menunjukkan bahwa ada sedikit atau tidak ada hubungan linier antara jumlah latihan soal dan nilai ujian.*
3. **Interpretasi Garis Regresi**
   Kemiringan garis regresi yang sangat kecil menunjukkan bahwa peningkatan jumlah latihan soal tidak secara signifikan mempengaruhi peningkatan nilai ujian yang artinya bahwa faktor lain mungkin lebih dominan dalam mempengaruhi nilai ujian siswa daripada hanya jumlah latihan soal.
4. **Horizontal/Vertikal.**
   Sumbu horizontal diberi label "Jumlah Latihan Soal (NL)" untuk menunjukkan variabel independen. Sumbu vertikal diberi label "Nilai Ujian (NT)" untuk menunjukkan variabel dependen.

Analisis Hasil
Dari plot, kita dapat melihat bahwa garis regresi hampir mendatar, yang menunjukkan bahwa jumlah latihan soal tidak memiliki korelasi yang kuat dengan nilai ujian siswa dalam dataset ini.

Nilai RMS error yang tinggi menunjukkan bahwa model tidak memprediksi nilai ujian dengan akurat. Ini bisa disebabkan oleh berbagai faktor seperti kurangnya fitur yang relevan, hubungan non-linear antara variabel, atau variabel yang tidak diperhitungkan. Mengingat rentang nilai ujian dari 0 hingga 100, kesalahan sebesar 19 poin bisa dianggap signifikan. Ini berarti bahwa prediksi model dapat menyimpang hingga hampir 20% dari nilai sebenarnya.

Kesimpulan
Dalam konteks nilai ujian dengan rentang 0 hingga 100, RMS error sebesar 19.18 dan 19.24 dapat dianggap cukup tinggi yang menunjukkan bahwa model regresi linier sederhana mungkin tidak cukup untuk memodelkan hubungan antara jumlah latihan soal dan nilai ujian secara akurat. Grafik menunjukkan bahwa hubungan antara jumlah latihan soal yang dilakukan siswa dan nilai ujian mereka sangat lemah. Ini mengindikasikan bahwa hanya menggunakan jumlah latihan soal untuk memprediksi nilai ujian tidak cukup dan faktor lain perlu dipertimbangkan untuk mendapatkan model prediksi yang lebih akurat.

MODEL REGRESI EKSPONENSIAL

Metode model regresi eksponensial adalah teknik statistik yang digunakan untuk memodelkan hubungan antara variabel dependen dan variabel independen dalam bentuk fungsi eksponensial. Model ini berguna ketika data menunjukkan pola pertumbuhan atau penurunan yang mengikuti bentuk eksponensial. Regresi eksponensial digunakan untuk menemukan hubungan eksponensial antara variabel independen (NL) dan variabel dependen (NT). Model eksponensial untuk penyelesaian soal memiliki bentuk:

$$NT = a \cdot e^{(b \cdot NL)}$$

Proses regresi eksponensial melibatkan menemukan nilai-nilai parameter aa dan bb yang paling sesuai dengan data yang diamati. Ini bisa dilakukan dengan berbagai metode, seperti metode kuadrat terkecil. Dimana a dan b adalah parameter yang harus kita estimasi. Untuk melakukan ini, kita akan mengambil logaritma dari kedua sisi persamaan untuk mengubahnya menjadi model linier yang lebih mudah diolah:

$$\boldsymbol{log}(NT) = \boldsymbol{log}(a) + b \cdot NL$$

Maka, dapat menggunakan regresi linier untuk menentukan parameter log(a) dan b.

PENERAPAN METODE MODEL EKSPONENSIAL MENGGUNAKAN PYTHON TERHADAP SOAL

Kode ini membaca data performa siswa, lalu mengidentifikasi hubungan eksponensial antara jumlah latihan soal dan nilai ujian, dan memodelkan hubungan tersebut menggunakan regresi eksponensial. Proses fitting dilakukan menggunakan curve_fit untuk penyelesaian dengan metode model regresi eksponensial, dan kualitas model diukur dengan galat RMS. Kemudian, hasilnya divisualisasikan dengan plot yang menunjukkan data asli dan kurva model eksponensial.

```
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

# Membaca data dari file CSV lokal
file_path = 'Student_Performance (1).csv'  # Ganti dengan path lengkap
ke file CSV Anda
data = pd.read_csv(file_path)

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

# Langkah-langkah Regresi Eksponensial
print("Langkah-langkah Regresi Eksponensial:")
# 1. Mendefinisikan fungsi model eksponensial
```

```
def exponential_model(x, a, b):
    return a * np.exp(b * x)

# 2. Menggunakan curve_fit untuk mencari parameter optimal a dan b
popt, pcov = curve_fit(exponential_model, NL.ravel(), NT, p0=(1,
0.01))

# 3. Memprediksi nilai NT berdasarkan model eksponensial yang telah
dilatih
NT_pred_exp = exponential_model(NL, *popt)

# 4. Menampilkan parameter dari model eksponensial
print(f"Parameter a: {popt[0]}")
print(f"Parameter b: {popt[1]}")

# Menghitung galat RMS untuk model regresi eksponensial
rms_exp = np.sqrt(mean_squared_error(NT, NT_pred_exp))

# Menampilkan galat RMS
print(f"\nRMS Regresi Eksponensial: {rms_exp}")

# Plot hasil regresi eksponensial
plt.scatter(NL, NT, color='blue', label='Data Asli')
plt.plot(NL, NT_pred_exp, color='green', label='Regresi Eksponensial')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian (NT)')
plt.title('Regresi Eksponensial')
plt.legend()
plt.show()
```

langkah-langkahnya secara detail:

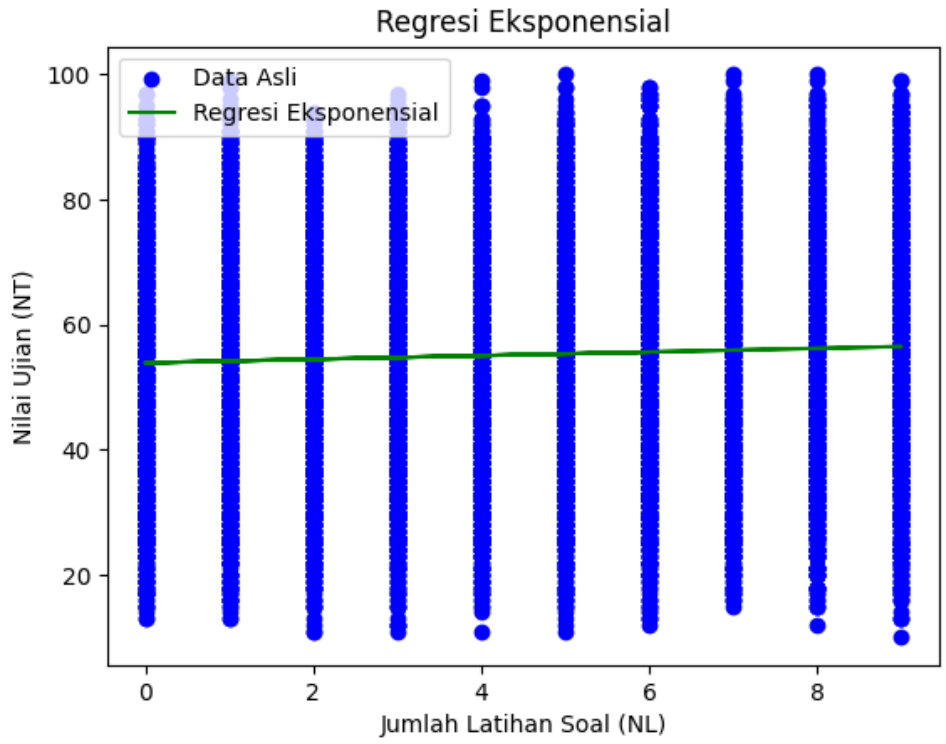
1. **Import library:**
   - pandas digunakan untuk membaca dan memanipulasi data.
   - numpy digunakan untuk operasi numerik.
   - matplotlib.pyplot digunakan untuk visualisasi data.
   - scipy.optimize.curve_fit digunakan untuk melakukan fitting kurva eksponensial.
   - sklearn.metrics.mean_squared_error digunakan untuk menghitung galat rata-rata kuadrat (RMS).
2. **Membaca dataset:**
   file_path: Menentukan path file lokal tempat dataset disimpan dan pd.read_csv(file_path): Membaca dataset dari file CSV yang terletak pada path yang ditentukan.
3. **Mengambil kolom yang relevan:**
   Untuk NL Mengambil kolom 'Sample Question Papers Practiced' dan mereshape-nya menjadi array 2D dengan satu kolom (karena sklearn mengharapkan input dalam bentuk ini) dan untuk NT Mengambil kolom 'Performance Index' sebagai array 1D.
4. **Mendifinisikan fungsi model eksponensial:**
   Fungsi exponential_model(x, a, b) untuk mendifinisikan model eksponensial y = a . e^ bx

5. **Menggunakan curve_fit untuk Mencari Parameter Optimal:**
   curve_fit digunakan untuk meminimalkan galat antara data yang diamati dan model eksponensial dan p0 adalah tebakan awal untuk parameter a dan b.
6. **Memprediksi Nilai NT Berdasarkan Model Eksponensial:**
   Menggunakan parameter optimal (popt) untuk memprediksi nilai NT.
7. **Menampilkan Parameter dari Model Eksponensial:**
   Cetak parameter a dan b dari model eksponensial dengan menggunakan print()
8. **Menghitung Galat RMS untuk Model Regresi Eksponensial:**
   Menggunakan mean_squared_error untuk menghitung galat rata-rata kuadrat (RMS) dimana RMS memberikan ukuran seberapa baik model cocok dengan data.
9. **Plot hasil regresi eksponensial:**
   Gunakan plt.scatter untuk menampilkan data asli, plt.plot untuk menampilkan kurva regresi eksponensial. Serta menambahkan label, judul, dan keterangan untuk plot.

Adapun outputnya adalah seperti berikut:

```
Parameter a: 53.908042983920076
Parameter b: 0.005240721479012398

RMS Regresi Eksponensial: 19.19364130292259
```



Penjelasan terkait output:
Gambar yang dihasilkan menunjukkan plot dari hubungan antara jumlah latihan soal (NL) dengan nilai ujian siswa (NT). Dalam gambar ini, titik biru menunjukkan data aktual, sedangkan garis merah menunjukkan hasil regresi eksponensial.

1. **Data Aktual (Titik Biru)**

Data titik biru tersebar merata di berbagai nilai NT untuk setiap nilai NL. Sebagian besar nilai NT berada di sekitar 50-70, terlepas dari nilai NL dan ada variabilitas yang tinggi dalam nilai NT untuk setiap nilai NL.

2. **Hasil Regresi Eksponensial (Garis Hijau)**
   Garis ini berada di sekitar nilai NT antara 50-60 untuk seluruh rentang NL. Garis hijau yang menunjukkan hasil regresi eksponensial tampak hampir datar yang menunjukkan bahwa model eksponensial yang diterapkan tidak mampu menangkap variabilitas yang tinggi dalam data NT terhadap NL.
3. **Horizontal/Vertikal.**
   Sumbu horizontal diberi label "Jumlah Latihan Soal (NL)" untuk menunjukkan variabel independen. Sumbu vertikal diberi label "Nilai Ujian (NT)" untuk menunjukkan variabel dependen.

Analisis Hasil

RMS Error 19.19364130292259 dalam konteks nilai NT yang berkisar antara 0 hingga 100, dianggap tinggi. Ini menunjukkan bahwa prediksi model memiliki deviasi yang cukup besar dari nilai aktual**.** RMS error yang tinggi Menunjukkan bahwa prediksi model memiliki deviasi yang besar dari nilai aktual, yang berarti model kurang akurat.

Sejalan dengan model eksponensial yang dihasilkan tampaknya tidak cocok dengan data yang ada. Garis merah yang hampir datar menunjukkan bahwa model eksponensial tidak dapat menjelaskan hubungan yang kompleks antara NL dan NT. Penyebab dari hal ini mungkin karena data memiliki variabilitas yang tinggi yang tidak dapat diakomodasi dengan model eksponensial sederhana.

Kesimpulan

Model eksponensial yang digunakan dalam analisis ini tidak berhasil menangkap hubungan antara jumlah latihan soal dan nilai ujian siswa yang dapat dilihat dari garis regresi yang hampir datar dan tidak sesuai dengan variabilitas data aktual. Pendekatan lain mungkin diperlukan untuk mendapatkan model yang lebih akurat dan sesuai dengan data.