

Kamis, 6 Juni 2024

Nama : Kadek Wisnu Parijata Putra
NIM : 21120122140036
Prodi : Teknik Komputer / 2022
Mata Kuliah : Metode Numerik / D
Github : https://github.com/wisnuprjt/Aplikasi-Regresi_Kadek-Wisnu_Metode-Numerik.git

Aplikasi Regresi untuk Pemecahan Problem

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

Berdasarkan ketentuan NIM Terakhir, $6\%4 = 2$. Maka, Mengerjakan Problem 2 dengan Metode 1 dan Metode 2. Diinginkan untuk mencari hubungan faktor yang mempengaruhi nilai ujian siswa (NT):

- Jumlah latihan soal (NL) terhadap nilai ujian (Problem 2)
- Model Linear (Metode 1)
- Model Pangkat Sederhana (Metode 2)

Sertakan kode testing untuk menguji kode sumber tersebut untuk menyelesaikan problem dalam gambar. Plot grafik titik data dan hasil regresinya masing-masing, hitung Galat RMS dari setiap Metode yang digunakan.

Aplikasi Regresi Linear

Regresi Linear adalah metode statistik yang digunakan untuk memodelkan hubungan yang hanya melibatkan dua *variable*, diantaranya *variable independen* (prediktor) dan *variable dependen* (respon). Tujuan dari Aplikasi Regresi Linear adalah untuk mendapatkan estimasi parameter (koefisien) dari model Regresi. Model Regresi dapat digunakan untuk eksplanatori maupun prediksi, selain itu tujuan yang lain adalah menemukan garis regresi yang memprediksi nilai variabel dependen berdasarkan nilai independen. Secara umum, model regresi linear sederhana dengan satu variabel independen dan fungsi linear dalam X dapat ditulis sebagai berikut :

$$Y_i = \beta_0 + \beta_1 X_{1i} + \epsilon_i$$

y_i adalah nilai Variabel independen observasi ke - i

β_0 adalah intercept (titik di mana garis memotong sumbu y).

β_1 adalah koefisien regresi kemiringan (*slope*).

x adalah variabel independen (prediktor).

ϵ merupakan nilai random error atau residu.

Aplikasi Pangkat Sederhana

Regresi Metode Pangkat Sederhana atau (*Power Low Regression*) adalah teknik pemodelan yang digunakan untuk menggambarkan antara dua hubungan variabel yang mengikuti pola fungsi Pangkat, dalam Model Pangkat Sederhana, hubungan antara variabel X (independen) dan variabel Y (dependen) dapat ditulis sebagai berikut :

$$Y = aX^b$$

Y adalah variabel dependen.

X adalah variabel independen.

a adalah koefisien skala.

b merupakan eksponen yang menunjukkan tingkat hubungan *non-linear* X dan Y .

Implementasi Kode :

```
# Kadek Wisnu Parijata Putra / 21120122140036
Aplikasi Regresi - Metode Numerik Kelas D

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 = '/content/Student_Performance.csv'
data = pd.read_csv(file_path)

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

```

NT = data['Performance Index'].values

# Reshape NL for sklearn
NL_resaped = NL.reshape(-1, 1)

# Metode 1: Regresi Linear
linear_model = LinearRegression()
linear_model.fit(NL_resaped, NT)
NT_pred_linear = linear_model.predict(NL_resaped)

# Metode 2: Regresi Pangkat Sederhana
#  $y = Cx^b \Rightarrow \log(y) = \log(C) + b \cdot \log(x)$ 
# Remove non-positive values for log transformation
NL_positive = NL[NL > 0]
NT_positive = NT[NL > 0]

# Transform to log scale
log_NL = np.log(NL_positive)
log_NT = np.log(NT_positive)

# Fit power model
power_model = LinearRegression()
power_model.fit(log_NL.reshape(-1, 1), log_NT)
log_C_power = power_model.intercept_
b_power = power_model.coef_[0]
C_power = np.exp(log_C_power)
NT_pred_log_power = power_model.predict(log_NL.reshape(-1, 1))
NT_pred_power = np.exp(NT_pred_log_power)

# Prepare NL and NT with filtered zero values for plotting and comparison
NL_filtered = NL[NL > 0]

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

# Plot Metode Linear
plt.subplot(1, 2, 1)
plt.scatter(NL, NT, color='blue', label='Data Asli')
plt.plot(NL, NT_pred_linear, color='red', label='Regresi Linear')
plt.xlabel('Jumlah Latihan Soal (NL)')
plt.ylabel('Nilai Ujian (NT)')

```

```

plt.title('Regresi Linear')
plt.legend()

# Plot Metode Pangkat Sederhana
plt.subplot(1, 2, 2)
plt.scatter(NL_filtered, NT_positive, color='blue', label='Data Asli')
plt.plot(NL_filtered, NT_pred_power, color='orange', label='Regresi
Pangkat Sederhana')
plt.xlabel('Jumlah Latihan Soal (NL)')
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_power = np.sqrt(mean_squared_error(NT_positive, NT_pred_power))

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

```

Ringkasan :

Analisis berikut merupakan bagian dari proses pengembangan Model Prediksi Kinerja Siswa. Data yang diperoleh dari File `'/content/Student_Performance.csv'`. berisi informasi tentang jumlah Latihan soal terhadap ujian yang telah dikerjakan oleh siswa (NL) dan Faktor yang mempengaruhi nilai ujian siswa (NT). Dua model Aplikasi Regresi yaitu Linear dan Pangkat sederhana diterapkan untuk memahami hubungan antara kedua variabel tersebut.

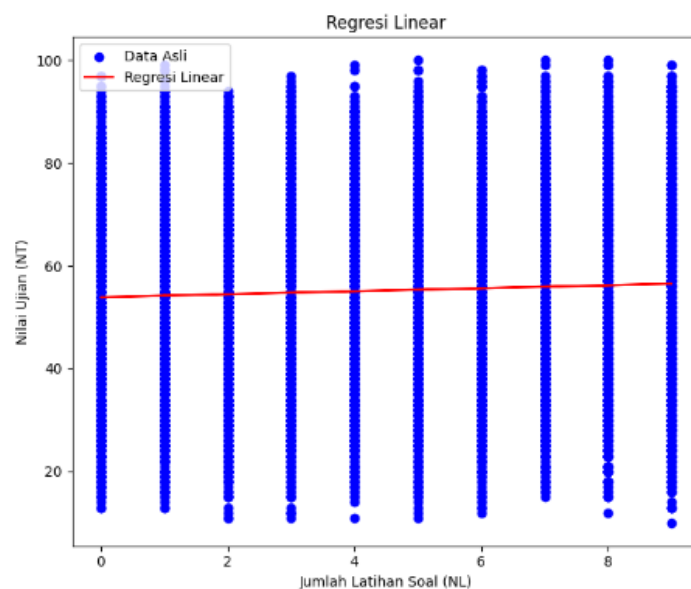
Model Regresi Linear digunakan dengan metode `sklearn.linear_model.LinearRegression`. Data Latihan soal terhadap ujian diubah ke dalam bentuk yang sesuai dengan permodelan. Selanjutnya Model Regresi Pangkat Sederhana digunakan dengan metode `from sklearn.metrics import mean_squared_error`. Data Latihan soal terhadap ujian diubah ke dalam bentuk yang sesuai dengan permodelan, memodelkan hubungan antara jam

belajar dan nilai ujian menggunakan regresi linear dan regresi pangkat sederhana dan mengevaluasi kesalahan model menggunakan *Root Mean Squared Error* (RMSE).

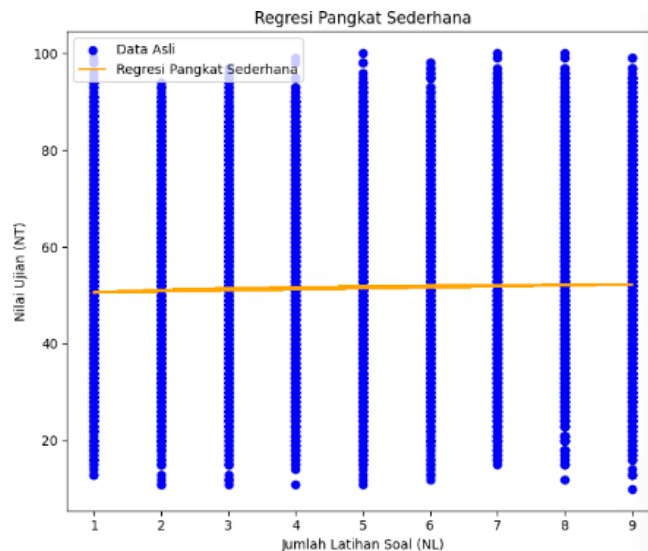
Konsep :

1. Data Preprocessing : Data Latihan soal terhadap ujian yang telah dikerjakan oleh siswa (NL) dan Faktor yang mempengaruhi nilai ujian siswa (NT) diekstraksi dari file CSV dan diproses untuk menganalisis regresi.
2. Model Regresi Linear : Model ini menggunakan metode `LinearRegression` dengan data (NL) yang di representasikan ulang sesuai dengan format model. Teknik untuk memodelkan hubungan antara Variabel Y (dependen) dan variabel X (independen) dengan menyesuaikan garis lurus (linear) ke data yang diamati.
3. Model Regresi Pangkat Sederhana : Teknik yang memodelkan hubungan antara Variabel Y (dependen) dan variabel X (independen) dalam bentuk fungsi pangkat. Model yang digunakan adalah $y = a \cdot X^b$, di mana a dan b sebagai parameter model
4. Plotting dan Evaluasi : Hasil dari masing-masing model akan ditampilkan dalam subplot yang terpisah, untuk membandingkan *visual*, RMS digunakan untuk mengevaluasi kinerja model.

Output :



Gambar 2.1 Output Regresi Linear



Gambar 2.2 Output Regresi Pangkat Sederhana

Galat RMS (*Root Means Squared Error*) :

```
# Hitung galat RMS
rms_linear = np.sqrt(mean_squared_error(NT, NT_pred_linear))
rms_power = np.sqrt(mean_squared_error(NT_positive, NT_pred_power))

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

```
RMS galat - Regresi Linear: 19.19360524526426
RMS galat - Regresi Pangkat Sederhana: 19.565035257712687
```

Gambar 2.3 Output *Root Means Squared Error* 2 Model

Galat RMS dihitung untuk mengevaluasi performa dari masing masing model. Nilai Galat RMS untuk Regresi Linear adalah sekitar 19.19, sedangkan untuk Regresi Pangkat Sederhana sekitar 19.56. Ini menunjukkan bahwa, secara keseluruhan, Regresi Linear memiliki tingkat galat yang sedikit lebih rendah dibandingkan dengan Regresi Pangkat Sederhana.

Analisis Hasil :

1. Hasil Visual : Garis Regresi Linear berwarna merah terlihat sedikit mendatar, menunjukkan perubahan jumlah (NL) atau Latihan soal dan (NT) indeks kinerja dengan garis lurus. Sedangkan Garis Regresi Pangkat sederhana berwarna *orange* sedikit mendatar, menunjukkan bahwa regresi pangkat sederhana tidak menangkap pola dalam data dengan baik.

2. RMS Error : Nilai Galat RMS Error untuk Regresi Linear adalah sekitar 19.19, sedangkan untuk Regresi Pangkat Sederhana sekitar 19.56. yang dimana menunjukkan bahwa, secara keseluruhan, Regresi Linear memiliki tingkat galat yang sedikit lebih rendah dibandingkan dengan Regresi Pangkat Sederhana.

Berdasarkan hasil visual dan nilai RMS Error yang dihitung oleh 2 Model yang digunakan yaitu (Linear dan Pangkat Sederhana) didapatkan Kesimpulan bahwa Regresi Linear maupun Regresi Pangkat Sederhana tidak mampu menangkap pola data dengan baik. Garis regresi linear berwarna merah dan garis regresi pangkat sederhana berwarna *orange* terlihat sedikit mendatar, menunjukkan bahwa perubahan jumlah latihan soal (NL) tidak memberikan dampak yang signifikan terhadap nilai ujian (NT) dalam model ini. Regresi linear memiliki nilai RMS Error yang sedikit lebih rendah sekitar 19.19, dibandingkan dengan regresi pangkat sederhana yaitu 19.56, perbedaan ini menunjukkan bahwa kedua model memiliki performa yang hampir sama dalam memprediksi data asli.