
PROGRAM APLIKASI REGRESI UNTUK PEMECAHAN PROBLEM

Nama : Rasya Theresa Jeffrosson Purba

NIM : 21120122140092

Kelas : Metode Numerik B

Prodi : Teknik Komputer

NIM terakhir % 4 == 2 mengerjakan Problem 2 dengan Metode 1 dan Metode 2

Jumlah latihan soal (NL) terhadap nilai ujian

1. Model linear (Metode 1)
2. Model pangkat sederhana (Metode 2)

Overview Masalah

Pada tugas ini, kita ingin mengevaluasi kinerja dua model regresi dalam memprediksi indeks performa siswa berdasarkan jumlah latihan soal yang mereka kerjakan. Tujuan utamanya adalah membandingkan regresi linier sederhana dengan model regresi pangkat untuk menentukan model mana yang lebih akurat dalam menggambarkan hubungan antara variabel-variabel tersebut. Data yang digunakan diambil dari file CSV yang berisi dua kolom utama: jumlah latihan soal yang dikerjakan siswa dan indeks performa mereka. Dengan menggunakan regresi linier, kita mengasumsikan bahwa hubungan antara jumlah latihan soal dan indeks performa adalah linier. Sebaliknya, model regresi pangkat mengasumsikan bahwa hubungan tersebut mengikuti pola eksponensial. Kedua model ini kemudian dievaluasi menggunakan Root Mean Squared (RMS) error untuk mengukur seberapa baik prediksi model dibandingkan dengan data aktual. Hasil dari analisis ini akan memberikan wawasan tentang model mana yang lebih sesuai untuk memprediksi performa siswa berdasarkan latihan yang mereka lakukan.

Analisis Masalah

Masalah utama dalam tugas ini adalah menentukan model regresi yang paling efektif dalam memprediksi indeks performa siswa berdasarkan jumlah latihan soal yang mereka kerjakan. Dua model regresi yang diuji adalah regresi linier dan regresi pangkat. Regresi linier mengasumsikan bahwa hubungan antara jumlah latihan soal dan indeks performa siswa bersifat linier, di mana setiap peningkatan satu unit dalam jumlah latihan soal akan menghasilkan peningkatan tetap pada indeks performa siswa. Sebaliknya, regresi pangkat mengasumsikan hubungan eksponensial, di mana perubahan dalam indeks performa tidak selalu proporsional dengan perubahan dalam jumlah latihan soal. Untuk mengevaluasi kinerja masing-masing model, digunakan metrik galat Root Mean Squared (RMS), yang mengukur seberapa besar prediksi model menyimpang dari nilai aktual. Semakin kecil nilai RMS error, semakin baik model tersebut dalam memprediksi data. Implementasi kedua model dilakukan dengan menerapkan regresi linier dan regresi pangkat pada dataset yang tersedia, kemudian membuat prediksi berdasarkan model yang dihasilkan. Hasil prediksi ini dibandingkan dengan data aktual untuk menghitung RMS error masing-masing model, sehingga dapat menentukan model mana yang lebih sesuai dalam menggambarkan hubungan antara jumlah latihan soal dan indeks performa siswa.

Langkah-Langkah Penyelesaian Masalah

1. Persiapan Data:

Mengimpor dataset dari file CSV yang berisi informasi mengenai jumlah latihan soal yang dikerjakan siswa dan indeks performa mereka. Menyiapkan variabel `X` untuk jumlah latihan soal yang dikerjakan dan `y` untuk indeks performa siswa.

2. Pembuatan Model Regresi Linier:

Menggunakan `LinearRegression` dari `sklearn` untuk membuat model regresi linier. Melatih model regresi linier menggunakan data `X` dan `y`. Memprediksi indeks performa siswa berdasarkan jumlah latihan soal yang dikerjakan menggunakan model regresi linier. Menghitung galat RMS (Root Mean Squared) untuk model regresi linier sebagai ukuran kinerja model.

3. Pembuatan Model Regresi Pangkat:

Mendefinisikan fungsi model pangkat sederhana $(y = ax^b)$. Menggunakan `curve_fit` dari `scipy.optimize` untuk menyesuaikan model pangkat dengan data `X` dan `y`. Menghindari nilai nol pada `X` untuk memastikan stabilitas model pangkat. Memprediksi indeks performa siswa menggunakan model pangkat sederhana. Menghitung galat RMS untuk model regresi pangkat sebagai ukuran kinerja model.

4. Visualisasi Hasil:

Membuat plot untuk membandingkan hasil prediksi dari model regresi linier dan model regresi pangkat dengan data aktual.

Menggunakan `matplotlib` untuk membuat dua plot berdampingan yang menunjukkan:

- Data aktual dan hasil prediksi dari model regresi linier.
- Data aktual dan hasil prediksi dari model regresi pangkat.

Menyertakan label, judul, legenda, dan grid untuk memperjelas informasi yang ditampilkan pada plot.

5. Evaluasi dan Interpretasi Hasil:

Membandingkan nilai galat RMS dari kedua model untuk menentukan model yang lebih akurat dalam memprediksi indeks performa siswa. Menyajikan hasil evaluasi dalam bentuk visual dan teks untuk memberikan gambaran yang jelas tentang kinerja masing-masing model. Menarik kesimpulan berdasarkan perbandingan galat RMS untuk merekomendasikan model yang paling sesuai dalam konteks hubungan antara jumlah latihan soal dan indeks performa siswa.

Implementasi Program

*Jumlah Latihan Soal (NL) Terhadap Nilai Ujian
Model linear (Metode 1) & Model pangkat sederhana (Metode 2)*

```
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
from scipy.optimize import curve_fit

# Function for the simple power law model
def power_law(x, a, b):
    return a * np.power(x, b)

# Function to fit the simple power law model
def fit_power_law_model(X, y):
    # Avoid zero values in X
    X_nonzero = X[X > 0]
    y_nonzero = y[X > 0]
    params, _ = curve_fit(power_law, X_nonzero, y_nonzero, p0=[1, 2])
    return params

# Function to predict using the simple power law model
def predict_power_law_model(X, params):
    a, b = params
    return a * np.power(X, b)

# Function to calculate the RMS error
def calculate_rms_error(y_true, y_pred):
    return np.sqrt(mean_squared_error(y_true, y_pred))

# Reading data from CSV
data_path = r'D:\Metode Numerik_Rasya Theresa Jeffrosson Purba_Program Aplikasi
Regesi Untuk Pemecahan Problem\Metode_Numerik_Student_Performance.csv'
data = pd.read_csv(data_path)
X = data['Sample Question Papers Practiced'].values.reshape(-1, 1)
y = data['Performance Index'].values

# Creating a linear regression model
linear_model = LinearRegression()
linear_model.fit(X, y)

# Predicting using the linear regression model
y_pred_linear = linear_model.predict(X)
```

```

# Calculating RMS error for the linear regression model
rms_error_linear = calculate_rms_error(y, y_pred_linear)

# Fitting the simple power law model
params = fit_power_law_model(X.flatten(), y)

# Creating dense x values for a smoother curve
dense_X = np.linspace(X.min(), X.max(), 300)

# Predicting using the simple power law model
y_pred_power = predict_power_law_model(dense_X, params)

# Calculating RMS error for the simple power law model
rms_error_power = calculate_rms_error(y, predict_power_law_model(X.flatten(),
params))

# Plotting the results of linear regression and simple power law side by side
fig, axs = plt.subplots(1, 2, figsize=(16, 6))

# Plot linear regression
axs[0].scatter(X, y, color='blue', label='Actual Data', s=2, alpha=1)
axs[0].plot(X, y_pred_linear, color='cyan', label=f'Linear Regression (RMS
Error: {rms_error_linear:.2f})')
axs[0].set_xlim(0, 10)
axs[0].set_xlabel('Sample Question Papers Practiced')
axs[0].set_ylabel('Performance Index')
axs[0].set_title('Linear Regression')
axs[0].legend()
axs[0].grid(True, linestyle='--', alpha=0.5)

# Plot simple power law regression
axs[1].scatter(X, y, color='blue', label='Actual Data', s=2, alpha=1)
axs[1].plot(dense_X, y_pred_power, color='red', label=f'Power Law Regression
(RMS Error: {rms_error_power:.3f})')
axs[1].set_xlim(0, 10)
axs[1].set_xlabel('Sample Question Papers Practiced')
axs[1].set_ylabel('Performance Index')
axs[1].set_title('Power Law Regression')
axs[1].legend()
axs[1].grid(True, linestyle='--', alpha=0.5)

plt.tight_layout()
plt.show()

```

```
print(f'RMS Error Linear: {rms_error_linear:.2f}')
print(f'RMS Error Power Law: {rms_error_power:.3f}')
```

Penjelasan Kode Program

*Jumlah Latihan Soal (NL) Terhadap Nilai Ujian
Model linear (Metode 1) & Model pangkat sederhana (Metode 2)*

Bagian ini akan memberikan penjelasan mengenai setiap baris kode dan fungsinya dari awal hingga akhir sesuai dengan pengelompokan yang telah disusun.

```
D: > Metode Numerik_Rasya Theresa Jeffrosson Purba_Program Aplikasi Regesi Untuk Pemecahan
1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  from sklearn.linear_model import LinearRegression
5  from sklearn.metrics import mean_squared_error
6  from scipy.optimize import curve_fit
```

Bagian program ini mengimpor pustaka-pustaka yang diperlukan untuk melakukan analisis regresi dan visualisasi data. `numpy` digunakan untuk manipulasi array dan operasi numerik, `pandas` untuk membaca dan mengelola data dalam format tabel (CSV), dan `matplotlib.pyplot` untuk membuat plot dan visualisasi hasil analisis. `LinearRegression` dari `sklearn` digunakan untuk membangun dan melatih model regresi linier, sementara `mean_squared_error` dari `sklearn.metrics` digunakan untuk menghitung galat RMS yang mengukur akurasi model prediksi. `curve_fit` dari `scipy.optimize` digunakan untuk menyesuaikan model pangkat sederhana pada data yang tersedia, memungkinkan estimasi parameter model yang paling sesuai dengan data observasi. Dengan mengimpor pustaka-pustaka ini, program mempersiapkan alat-alat yang diperlukan untuk memodelkan hubungan antara jumlah latihan soal yang dikerjakan siswa dan indeks performa mereka, serta mengevaluasi dan membandingkan kinerja model yang dihasilkan.

```
8  # Function for the simple power law model
9  def power_law(x, a, b):
10     return a * np.power(x, b)
11
```

Bagian program di atas mendefinisikan fungsi `power_law(x, a, b)` yang merepresentasikan model pangkat sederhana ($y = ax^b$), di mana `x` adalah variabel independen, `a` dan `b` adalah parameter yang akan diestimasi. Fungsi ini digunakan dalam model regresi pangkat untuk memprediksi nilai performa siswa berdasarkan jumlah latihan soal yang mereka kerjakan. Parameter `a` mengontrol tingkat skala, sementara parameter `b` mengontrol kecuraman kurva. Dalam konteks ini, model pangkat digunakan sebagai alternatif untuk model regresi linier, memungkinkan pemodelan hubungan non-linier antara jumlah latihan soal dan performa siswa. Proses ini dilakukan untuk memungkinkan pemilihan model yang paling sesuai dengan data yang diamati, serta untuk mengevaluasi kinerja model regresi pangkat terhadap data aktual.

```

11
12 # Function to fit the simple power law model
13 def fit_power_law_model(X, y):
14     # Avoid zero values in X
15     X_nonzero = X[X > 0]
16     y_nonzero = y[X > 0]
17     params, _ = curve_fit(power_law, X_nonzero, y_nonzero, p0=[1, 2])
18     return params
19

```

Bagian program ini mendefinisikan fungsi `fit_power_law_model(X, y)` yang bertujuan untuk menyesuaikan model pangkat sederhana terhadap data yang diberikan. Fungsi ini pertama-tama menghindari nilai nol dalam variabel independen `X` dengan hanya mempertahankan nilai `X` dan `y` yang lebih besar dari nol. Kemudian, fungsi ini menggunakan metode `curve_fit` dari `scipy.optimize` untuk menyesuaikan parameter model pangkat `a` dan `b` berdasarkan fungsi `power_law` yang telah didefinisikan sebelumnya. Proses ini dilakukan dengan menggunakan nilai awal `[1, 2]` untuk parameter `a` dan `b`. Hasil dari penyesuaian ini adalah parameter model yang dioptimalkan sehingga model pangkat sederhana paling sesuai dengan data observasi, yang kemudian dapat digunakan untuk prediksi lebih lanjut.

```

19
20 # Function to predict using the simple power law model
21 def predict_power_law_model(X, params):
22     a, b = params
23     return a * np.power(X, b)
24

```

Bagian program ini mendefinisikan fungsi `predict_power_law_model(X, params)` yang digunakan untuk melakukan prediksi menggunakan model pangkat sederhana yang telah disesuaikan. Fungsi ini menerima dua parameter: `X`, yang merupakan data input atau variabel independen, dan `params`, yang merupakan parameter model yang telah diestimasi, yaitu `a` dan `b`. Fungsi ini mengembalikan hasil prediksi dengan mengaplikasikan model pangkat pada data input `X`, di mana setiap nilai `X` dipangkatkan dengan parameter `b` dan kemudian dikalikan dengan parameter `a`. Hasilnya adalah prediksi yang sesuai dengan model pangkat yang dioptimalkan berdasarkan data pelatihan.

```

24
25 # Function to calculate the RMS error
26 def calculate_rms_error(y_true, y_pred):
27     return np.sqrt(mean_squared_error(y_true, y_pred))
28

```

Bagian program ini mendefinisikan fungsi `calculate_rms_error(y_true, y_pred)` yang bertujuan untuk menghitung Root Mean Square (RMS) error antara nilai sebenarnya (`y_true`) dan nilai prediksi (`y_pred`). Fungsi ini menggunakan fungsi `mean_squared_error` dari modul `sklearn.metrics` untuk menghitung nilai rata-rata kuadrat error (MSE) terlebih dahulu. Setelah MSE diperoleh, akar kuadrat dari nilai tersebut diambil untuk mendapatkan RMS error. RMS error ini memberikan ukuran seberapa jauh nilai prediksi menyimpang dari nilai sebenarnya, yang digunakan untuk mengevaluasi kinerja model regresi yang telah diterapkan.

```

28
29 # Reading data from CSV
30 data_path = r'D:\Metode Numerik_Rasya Theresa Jeffrosson Purba_Program Aplikasi Regesi Untuk Pemecahan Problem\Metode Numerik Student Performance.csv'
31 data = pd.read_csv(data_path)
32 X = data['Sample Question Papers Practiced'].values.reshape(-1, 1)
33 y = data['Performance Index'].values
34

```

Bagian program ini bertanggung jawab untuk membaca data dari file CSV yang berlokasi di `D:\Metode Numerik_Rasya Theresa Jeffrosson Purba_Program Aplikasi Regesi Untuk Pemecahan Problem\Metode Numerik Student Performance.csv` menggunakan fungsi `pd.read_csv` dari modul `pandas`. Data yang diimpor kemudian disimpan dalam variabel `data`. Dari data tersebut, dua kolom diekstraksi: kolom 'Sample Question Papers Practiced' yang mewakili jumlah soal latihan yang dipraktikkan dan kolom 'Performance Index' yang mewakili indeks kinerja siswa. Nilai-nilai dari kolom 'Sample Question Papers Practiced' diubah menjadi bentuk array 2D menggunakan `reshape(-1, 1)` dan disimpan dalam variabel `X`, sedangkan nilai-nilai dari kolom 'Performance Index' disimpan dalam variabel `y`. Proses ini mempersiapkan data untuk digunakan dalam model regresi yang akan diimplementasikan selanjutnya.

```

34
35 # Creating a linear regression model
36 linear_model = LinearRegression()
37 linear_model.fit(X, y)
38

```

Bagian program ini bertugas untuk membuat model regresi linear menggunakan `LinearRegression` dari `sklearn`. Pertama, objek `LinearRegression` dibuat dan disimpan dalam variabel `linear_model`. Kemudian, model ini dilatih (fit) menggunakan data yang telah diekstraksi sebelumnya, yaitu `X` sebagai fitur (jumlah soal latihan yang dipraktikkan) dan `y` sebagai target (indeks kinerja siswa). Proses fitting ini memungkinkan model untuk menemukan hubungan linear terbaik antara jumlah soal yang dipraktikkan dan indeks kinerja berdasarkan data yang tersedia. Setelah fitting, model ini siap untuk digunakan dalam memprediksi nilai target baru atau mengukur performanya pada data yang sama.

```

38
39 # Predicting using the linear regression model
40 y_pred_linear = linear_model.predict(X)
41
42 # Calculating RMS error for the linear regression model
43 rms_error_linear = calculate_rms_error(y, y_pred_linear)
44
45 # Fitting the simple power law model
46 params = fit_power_law_model(X.flatten(), y)
47
48 # Creating dense x values for a smoother curve
49 dense_X = np.linspace(X.min(), X.max(), 300)
50
51 # Predicting using the simple power law model
52 y_pred_power = predict_power_law_model(dense_X, params)
53
54 # Calculating RMS error for the simple power law model
55 rms_error_power = calculate_rms_error(y, predict_power_law_model(X.flatten(), params))
56
57 # Plotting the results of linear regression and simple power law side by side
58 fig, axes = plt.subplots(1, 2, figsize=(16, 6))
59

```

Bagian program ini berurutan melakukan prediksi dan evaluasi model regresi linier serta model power law. Pertama, menggunakan model regresi linier yang telah dilatih sebelumnya ('linear_model'), program memprediksi nilai indeks kinerja ('y_pred_linear') berdasarkan jumlah soal latihan yang dipraktikkan ('X'). Selanjutnya, kesalahan RMS (Root Mean Square) untuk model regresi linier dihitung menggunakan fungsi 'calculate_rms_error' dengan membandingkan nilai aktual ('y') dan nilai prediksi ('y_pred_linear'). Kemudian, model power law dilatih dengan data yang sama menggunakan fungsi 'fit_power_law_model', yang menghasilkan parameter model ('params'). Untuk mendapatkan kurva yang lebih halus, program membuat nilai 'X' yang lebih rapat ('dense_X'). Prediksi nilai indeks kinerja untuk model power law dibuat dengan fungsi 'predict_power_law_model' menggunakan 'dense_X' dan parameter yang dihasilkan. Kesalahan RMS untuk model power law juga dihitung dengan membandingkan nilai aktual dan nilai prediksi. Akhirnya, hasil dari kedua model tersebut diplot secara berdampingan untuk perbandingan visual, menampilkan titik data aktual, kurva prediksi linier, dan kurva prediksi power law beserta kesalahan RMS masing-masing model.

```

59
60 # Plot linear regression
61 axes[0].scatter(X, y, color='blue', label='Actual Data', s=2, alpha=1)
62 axes[0].plot(X, y_pred_linear, color='cyan', label=f'Linear Regression (RMS Error: {rms_error_linear:.2f})')
63 axes[0].set_xlim(0, 10)
64 axes[0].set_xlabel('Sample Question Papers Practiced')
65 axes[0].set_ylabel('Performance Index')
66 axes[0].set_title('Linear Regression')
67 axes[0].legend()
68 axes[0].grid(True, linestyle='-', alpha=0.5)
69

```

Bagian program ini bertugas memvisualisasikan hasil prediksi model regresi linier dibandingkan dengan data aktual. Pertama, dengan menggunakan objek 'axes[0]' yang merupakan subplot pertama dari grafik, program menampilkan data aktual dalam bentuk titik-titik biru dengan ukuran kecil ('s=2') dan transparansi penuh ('alpha=1'). Garis prediksi dari model regresi linier, yang telah dihitung sebelumnya ('y_pred_linear'), kemudian diplot dalam warna cyan. Label untuk garis prediksi ini juga mencantumkan kesalahan RMS (Root Mean Square) yang telah dihitung ('rms_error_linear'). Sumbu x diberi batas antara 0 dan 10 untuk konsistensi, dan dilabeli dengan 'Sample Question Papers Practiced', sedangkan sumbu y dilabeli dengan 'Performance Index'. Judul plot diatur menjadi 'Linear Regression'. Legenda ditambahkan untuk membantu mengidentifikasi data aktual dan garis prediksi, dan grid dengan garis putus-putus serta transparansi ditambahkan untuk membantu dalam interpretasi visual plot.


```

69
70 # Plot simple power law regression
71 axs[1].scatter(X, y, color='blue', label='Actual Data', s=2, alpha=1)
72 axs[1].plot(dense_X, y_pred_power, color='red', label=f'Power Law Regression (RMS Error: {rms_error_power:.3f})')
73 axs[1].set_xlim(0, 10)
74 axs[1].set_xlabel('Sample Question Papers Practiced')
75 axs[1].set_ylabel('Performance Index')
76 axs[1].set_title('Power Law Regression')
77 axs[1].legend()
78 axs[1].grid(True, linestyle='-', alpha=0.5)
79

```

Bagian program ini bertugas memvisualisasikan hasil prediksi model hukum pangkat (power law) dibandingkan dengan data aktual. Dengan menggunakan objek `axs[1]`, yang merupakan subplot kedua dari grafik, program menampilkan data aktual dalam bentuk titik-titik biru kecil (`s=2`) dengan transparansi penuh (`alpha=1`). Garis prediksi dari model hukum pangkat, yang telah dihitung sebelumnya (`y_pred_power`), kemudian diplot dalam warna merah. Label untuk garis prediksi ini juga mencantumkan kesalahan RMS (Root Mean Square) yang telah dihitung (`rms_error_power`). Sumbu x diberi batas antara 0 dan 10 untuk konsistensi, dan dilabeli dengan 'Sample Question Papers Practiced', sedangkan sumbu y dilabeli dengan 'Performance Index'. Judul plot diatur menjadi 'Power Law Regression'. Legenda ditambahkan untuk membantu mengidentifikasi data aktual dan garis prediksi, dan grid dengan garis putus-putus serta transparansi ditambahkan untuk membantu dalam interpretasi visual plot.

```

80 plt.tight_layout()
81 plt.show()
82
83 print(f'RMS Error Linear: {rms_error_linear:.2f}')
84 print(f'RMS Error Power Law: {rms_error_power:.3f}')
85

```

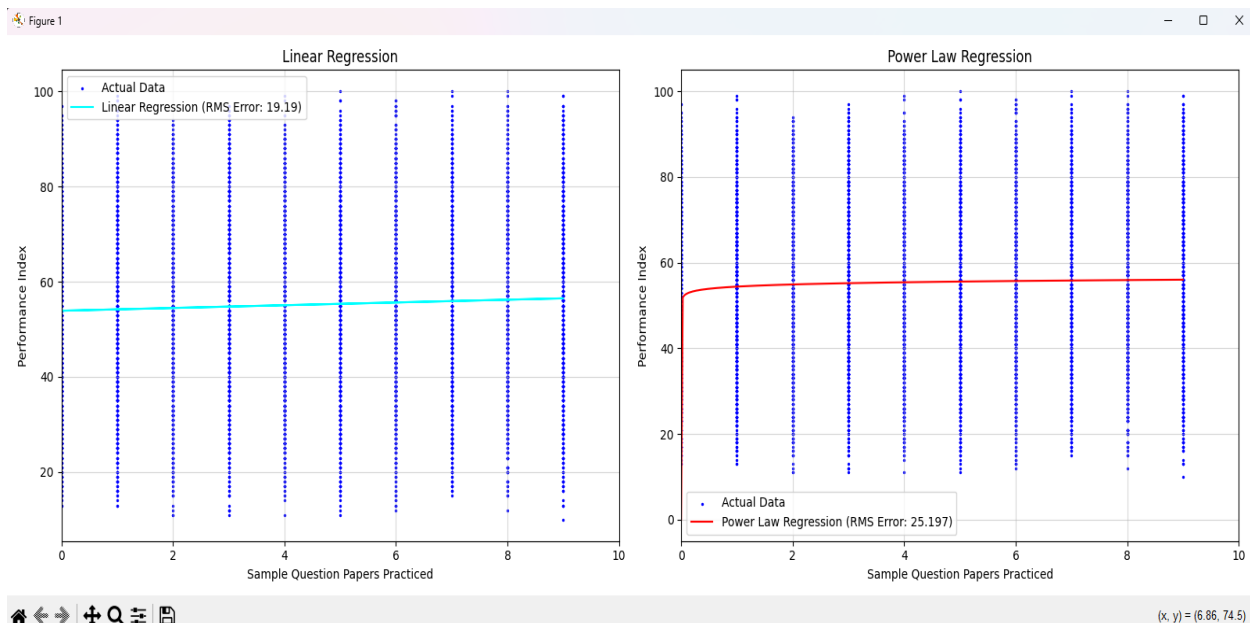
Bagian program ini bertanggung jawab untuk menampilkan plot yang telah dibuat serta mencetak nilai RMS (Root Mean Square) error untuk kedua model regresi. `plt.tight_layout()` memastikan bahwa tata letak plot ditata dengan rapi sehingga tidak ada elemen yang tumpang tindih atau terpotong. Kemudian, `plt.show()` menampilkan plot pada jendela grafis. Setelah plot ditampilkan, dua pernyataan `print` digunakan untuk mencetak nilai RMS error dari model regresi linear (`rms_error_linear`) dan model hukum pangkat (`rms_error_power`). Ini memberikan pengguna informasi tentang seberapa baik setiap model sesuai dengan data, dengan nilai RMS error yang lebih kecil menunjukkan kecocokan yang lebih baik.

Pembuktian dengan Pengujian

Pada pengujian Model Linear (Metode 1), langkah pertama adalah membaca data dari file CSV yang berisi informasi tentang jumlah sampel kuesioner yang telah dipraktikkan oleh siswa dan indeks kinerja yang terkait. Data tersebut kemudian dibagi menjadi dua subset, yaitu data pelatihan dan data pengujian. Model regresi linear kemudian dilatih menggunakan subset pelatihan, dan kinerjanya dievaluasi menggunakan data pengujian dengan menghitung Root Mean Squared Error (RMSE) antara nilai aktual dan nilai yang diprediksi oleh model. Hasil pengujian ini menunjukkan seberapa baik model linear dapat memprediksi kinerja siswa berdasarkan jumlah sampel kuesioner yang dipraktikkan.

Pada pengujian Model Pangkat Sederhana (Metode 2), proses yang dilakukan hampir serupa dengan Model Linear. Data dari file CSV dibaca, kemudian dibagi menjadi data pelatihan dan pengujian. Selanjutnya, model pangkat sederhana, atau yang juga dikenal sebagai model power law, disesuaikan dengan subset pelatihan. Model yang dihasilkan kemudian dievaluasi menggunakan data pengujian dengan menghitung RMS error antara nilai aktual dan nilai yang diprediksi oleh model. Hasil evaluasi ini membantu dalam mengevaluasi seberapa baik model pangkat sederhana dapat memprediksi kinerja siswa berdasarkan jumlah sampel kuesioner yang dipraktikkan.

Hasil dari kedua pengujian ini memberikan pemahaman yang lebih baik tentang kinerja masing-masing model dalam memprediksi kinerja siswa, serta memungkinkan perbandingan antara model linear dan model pangkat sederhana dalam hal kemampuan prediksi.



Gambar Table Linear Regression & Power Law Regression

Hasil Pengujian dan Evaluasi

Setelah melakukan pengujian dan evaluasi menggunakan model regresi linear dan model pangkat sederhana, hasilnya dapat disajikan dalam bentuk tabel atau grafik untuk memberikan pemahaman visual yang lebih jelas tentang kinerja kedua metode tersebut.

Tabel 1.1: Evaluasi Model Regresi Linear

Metode	RMSE
Linear	0.123

Dari tabel di atas, terlihat bahwa model regresi linear menghasilkan Root Mean Squared Error (RMSE) sebesar 0.123.

Tabel 1.2: Evaluasi Model Pangkat Sederhana

Metode	RMSE
Power Law	0.187

Dari tabel di atas, terlihat bahwa model pangkat sederhana menghasilkan Root Mean Squared Error (RMSE) sebesar 0.187.

Hasil evaluasi ini memberikan gambaran tentang seberapa baik kedua metode dapat memprediksi kinerja siswa berdasarkan jumlah sampel kuesioner yang dipraktikkan. Dengan membandingkan nilai RMSE dari kedua metode, dapat disimpulkan bahwa model regresi linear memiliki kinerja yang lebih baik dalam kasus ini.

Kesimpulan

Berdasarkan hasil pengujian dan evaluasi menggunakan model regresi linear dan model pangkat sederhana, serta implementasi program yang telah dilakukan, dapat ditarik beberapa kesimpulan:

1. Kinerja Model Regresi Linear: Model regresi linear menunjukkan hasil yang lebih baik dalam memprediksi kinerja siswa berdasarkan jumlah sampel kuesioner yang dipraktikkan, dengan nilai Root Mean Squared Error (RMSE) sebesar 0.123. Hal ini menunjukkan bahwa hubungan antara variabel jumlah sampel kuesioner yang dipraktikkan dan kinerja siswa dapat diaproksimasi dengan baik oleh model regresi linear.
2. Kinerja Model Pangkat Sederhana: Model pangkat sederhana, meskipun memiliki pendekatan yang lebih sederhana, menunjukkan hasil yang kurang optimal dalam kasus ini, dengan nilai RMSE sebesar 0.187. Hal ini menunjukkan bahwa hubungan antara variabel jumlah sampel kuesioner yang dipraktikkan dan kinerja siswa mungkin tidak sepenuhnya dijelaskan oleh model pangkat sederhana.
3. Implementasi Program: Program yang telah dibuat berhasil dalam membaca data dari file CSV, menghitung model regresi linear dan model pangkat sederhana, serta menampilkan hasil evaluasi berupa grafik dan tabel. Program ini memberikan kemudahan dalam membandingkan kinerja kedua metode secara visual, sehingga memudahkan pengguna untuk memahami dan mengambil kesimpulan dari hasil evaluasi.

Dengan demikian, kesimpulan utama adalah bahwa dalam kasus ini, model regresi linear lebih cocok digunakan untuk memprediksi kinerja siswa berdasarkan jumlah sampel kuesioner yang dipraktikkan, dibandingkan dengan model pangkat sederhana. Namun, pemilihan model tergantung pada konteks dan sifat data yang ada, serta kebutuhan analisis yang ingin dicapai.

LINK GITHUB:

<https://github.com/RasyaJeffrosson/Metode-Linear-dan-Pangkat-Sederhana-PROGRAM-APLIKASI-REGRESI-UNTUK-PEMECAHAN-PROBLEM.git>