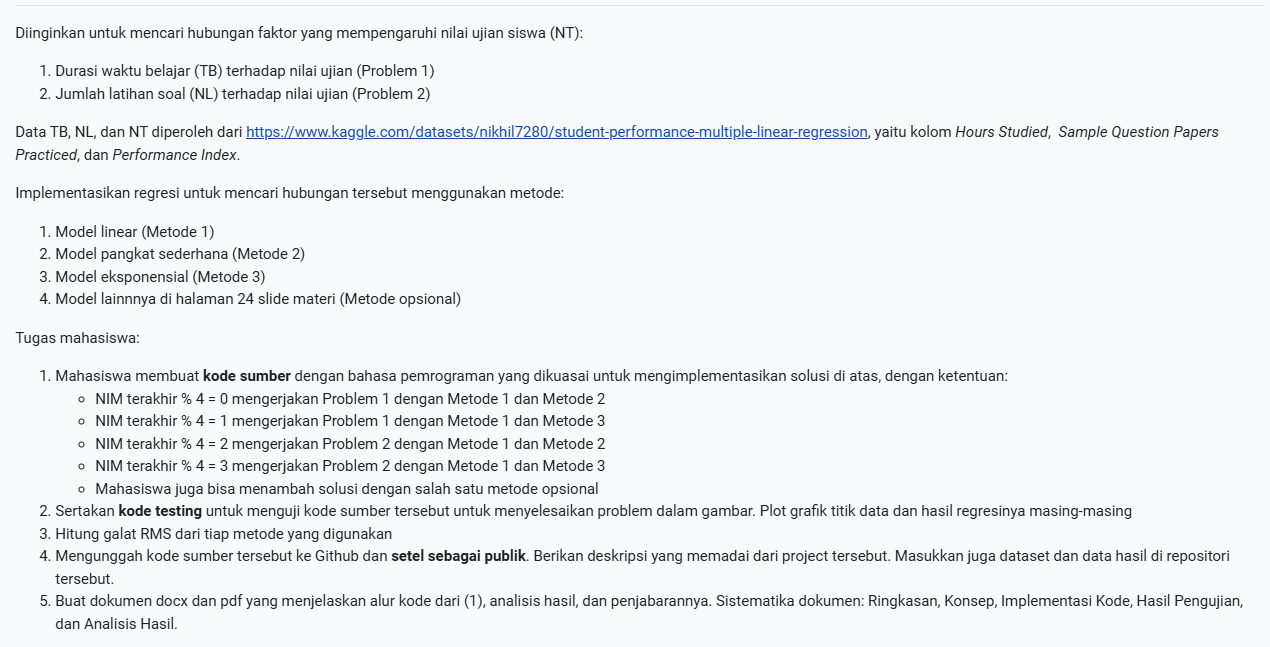
Nama : Finodya Yahdun

NIM : 21120122130065

Mata Kuliah : Metode Numerik

Kelas : D



|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from sklearn.linear\_model import LinearRegression  from scipy.optimize import curve\_fit  from sklearn.metrics import mean\_squared\_error  # Memuat data dari Kaggle  data = pd.read\_csv("Student\_Performance.csv")  # Problem 1: Durasi waktu belajar (TB) terhadap nilai ujian (NT)  TB = data["Hours Studied"].values.reshape(-1, 1)  NT = data["Performance Index"].values  # Metode 1: Model Linear  linear\_model = LinearRegression()  linear\_model.fit(TB, NT)  NT\_pred\_linear = linear\_model.predict(TB)  # Metode 3: Model Eksponensial  def exponential\_model(x, a, b):      return a \* np.exp(b \* x)  popt\_exponential, \_ = curve\_fit(exponential\_model, TB.flatten(), NT)  NT\_pred\_exponential = exponential\_model(TB.flatten(), \*popt\_exponential)  # Metode Opsional: Model Laju Pertumbuhan Jenuh  def logistic\_growth\_model(x, a, b, c):      return a / (1 + np.exp(-b \* (x - c)))  popt\_logistic, \_ = curve\_fit(logistic\_growth\_model, TB.flatten(), NT, p0=[max(NT), 1, np.median(TB)])  NT\_pred\_logistic = logistic\_growth\_model(TB.flatten(), \*popt\_logistic)  # Menghitung galat RMS  rms\_linear = np.sqrt(mean\_squared\_error(NT, NT\_pred\_linear))  rms\_exponential = np.sqrt(mean\_squared\_error(NT, NT\_pred\_exponential))  rms\_logistic = np.sqrt(mean\_squared\_error(NT, NT\_pred\_logistic))  print(f"RMS Error - Model Linear: {rms\_linear}")  print(f"RMS Error - Model Eksponensial: {rms\_exponential}")  print(f"RMS Error - Model Laju Pertumbuhan Jenuh: {rms\_logistic}")  # Plot hasil regresi  plt.figure(figsize=(14, 7))  # Plot model linear  plt.subplot(1, 3, 1)  plt.scatter(TB, NT, label='Data asli', color='blue')  plt.plot(TB, NT\_pred\_linear, label='Regresi Linear', color='red')  plt.xlabel('Durasi waktu belajar (TB)')  plt.ylabel('Nilai ujian (NT)')  plt.title('Model Linear')  plt.legend()  # Plot model eksponensial  plt.subplot(1, 3, 2)  plt.scatter(TB, NT, label='Data asli', color='blue')  plt.plot(TB, NT\_pred\_exponential, label='Regresi Eksponensial', color='red')  plt.xlabel('Durasi waktu belajar (TB)')  plt.ylabel('Nilai ujian (NT)')  plt.title('Model Eksponensial')  plt.legend()  # Plot model laju pertumbuhan jenuh  plt.subplot(1, 3, 3)  plt.scatter(TB, NT, label='Data asli', color='blue')  plt.plot(TB, NT\_pred\_logistic, label='Regresi Laju Pertumbuhan Jenuh', color='red')  plt.xlabel('Durasi waktu belajar (TB)')  plt.ylabel('Nilai ujian (NT)')  plt.title('Model Laju Pertumbuhan Jenuh')  plt.legend()  plt.tight\_layout()  plt.show() |

Penjelasan Source Code

Source code ini digunakan untuk memodelkan hubungan antara durasi waktu belajar (TB) dan nilai ujian (NT) menggunakan tiga metode regresi: regresi linear, regresi eksponensial, dan regresi laju pertumbuhan jenuh. Data diambil dari file CSV "Student\_Performance.csv". Hasil regresi dievaluasi menggunakan Root Mean Square Error (RMSE), dan visualisasi dari ketiga model regresi ditampilkan dalam bentuk grafik.

Source code ini mengandalkan beberapa library penting untuk memuat, mengelola, dan memproses data serta untuk melakukan dan mengevaluasi model regresi. Library pandas digunakan untuk memuat dan mengelola data, menyediakan struktur data yang fleksibel dan efisien seperti DataFrame yang mempermudah manipulasi dan analisis data. Library numpy digunakan untuk operasi numerik pada data, menyediakan array multidimensi dan fungsi-fungsi matematis yang cepat. Untuk visualisasi data, matplotlib.pyplot digunakan untuk membuat grafik dan plot yang informatif dan menarik. sklearn.linear\_model menyediakan alat untuk membangun model regresi linear, memungkinkan pembuatan dan penyesuaian model linear terhadap data. scipy.optimize digunakan untuk menyesuaikan model regresi non-linear, seperti regresi eksponensial dan model laju pertumbuhan jenuh, dengan menemukan parameter model yang paling sesuai. Terakhir, sklearn.metrics digunakan untuk evaluasi kinerja model dengan menghitung Root Mean Square Error (RMSE), memberikan ukuran kuantitatif dari kesalahan prediksi.

Implementasi Source Code

1. Muat data

|  |
| --- |
| data = pd.read\_csv("Student\_Performance.csv") |

Data dari file CSV "Student\_Performance.csv" dimuat ke dalam DataFrame menggunakan pandas.

1. Mengambil Kolom yang relevan

|  |
| --- |
| TB = data["Hours Studied"].values.reshape(-1, 1)  NT = data["Performance Index"].values |

Kolom yang relevan diambil dari DataFrame dan disimpan dalam variabel TB (durasi waktu belajar) dan NT (nilai ujian).

1. Regresi Linear

|  |
| --- |
| linear\_model = LinearRegression()  linear\_model.fit(TB, NT)  NT\_pred\_linear = linear\_model.predict(TB) |

Model regresi linear dibangun dan disesuaikan dengan data menggunakan LinearRegression dari sklearn.linear\_model.

1. Regresi Eksponensial

|  |
| --- |
| def exponential\_model(x, a, b):  return a \* np.exp(b \* x)  popt\_exponential, \_ = curve\_fit(exponential\_model, TB.flatten(), NT)  NT\_pred\_exponential = exponential\_model(TB.flatten(), \*popt\_exponential) |

Model regresi eksponensial didefinisikan dan parameter model dioptimalkan menggunakan curve\_fit dari scipy.optimize.

1. Regresi Laju Pertumbuhan Jenuh

|  |
| --- |
| def logistic\_growth\_model(x, a, b, c):  return a / (1 + np.exp(-b \* (x - c)))  popt\_logistic, \_ = curve\_fit(logistic\_growth\_model, TB.flatten(), NT, p0=[max(NT), 1, np.median(TB)])  NT\_pred\_logistic = logistic\_growth\_model(TB.flatten(), \*popt\_logistic) |

Model regresi laju pertumbuhan jenuh didefinisikan dan parameter model dioptimalkan menggunakan curve\_fit.

1. Menghitung Galat RMS

|  |
| --- |
| rms\_linear = np.sqrt(mean\_squared\_error(NT, NT\_pred\_linear))  rms\_exponential = np.sqrt(mean\_squared\_error(NT, NT\_pred\_exponential))  rms\_logistic = np.sqrt(mean\_squared\_error(NT, NT\_pred\_logistic))  print(f"RMS Error - Model Linear: {rms\_linear}")  print(f"RMS Error - Model Eksponensial: {rms\_exponential}")  print(f"RMS Error - Model Laju Pertumbuhan Jenuh: {rms\_logistic}") |

RMSE dihitung untuk masing-masing model regresi untuk mengevaluasi kinerja prediksinya.

1. Membuat Grafik

|  |
| --- |
| plt.figure(figsize=(14, 7))  # Plot model linear  plt.subplot(1, 3, 1)  plt.scatter(TB, NT, label='Data asli', color='blue')  plt.plot(TB, NT\_pred\_linear, label='Regresi Linear', color='red')  plt.xlabel('Durasi waktu belajar (TB)')  plt.ylabel('Nilai ujian (NT)')  plt.title('Model Linear')  plt.legend()  # Plot model eksponensial  plt.subplot(1, 3, 2)  plt.scatter(TB, NT, label='Data asli', color='blue')  plt.plot(TB, NT\_pred\_exponential, label='Regresi Eksponensial', color='red')  plt.xlabel('Durasi waktu belajar (TB)')  plt.ylabel('Nilai ujian (NT)')  plt.title('Model Eksponensial')  plt.legend()  # Plot model laju pertumbuhan jenuh  plt.subplot(1, 3, 3)  plt.scatter(TB, NT, label='Data asli', color='blue')  plt.plot(TB, NT\_pred\_logistic, label='Regresi Laju Pertumbuhan Jenuh', color='red')  plt.xlabel('Durasi waktu belajar (TB)')  plt.ylabel('Nilai ujian (NT)')  plt.title('Model Laju Pertumbuhan Jenuh')  plt.legend()  plt.tight\_layout()  plt.show() |

Grafik dari data asli dan prediksi masing-masing model regresi dibuat dan ditampilkan menggunakan matplotlib.pyplot.

Setelah menghitung regresi menggunakan ketiga metode, RMSE dihitung untuk masing-masing model. Hasil RMSE dicetak untuk membandingkan seberapa baik setiap model dalam memprediksi nilai ujian berdasarkan durasi waktu belajar:

|  |
| --- |
| print(f"RMS Error - Model Linear: {rms\_linear}")  print(f"RMS Error - Model Eksponensial: {rms\_exponential}")  print(f"RMS Error - Model Laju Pertumbuhan Jenuh: {rms\_logistic}") |

Grafik yang dihasilkan menunjukkan visualisasi data asli serta kurva yang dihasilkan oleh masing-masing model regresi. RMSE memberikan ukuran kuantitatif dari galat prediksi masing-masing model. Model dengan RMSE terkecil dianggap paling akurat dalam memprediksi hubungan antara durasi waktu belajar dan nilai ujian.

