Materi 10

Machine Learning Regresi Linear dan Polinomial

Moch. Alldho Candra Ramadhan (1227030020)

Soal Nomor 1:

Modifikasi kode program contoh 3 dan diberikan Dataset X = [1,2,3,4,5,6,7,8,9,10] Y = [3,7,13,21,31,43,57,73,91,111]!

Jawaban:

|  |
| --- |
| # Nomor 1 (Dataset)  X = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]).  Y = np.array([3, 7, 13, 21, 31, 43, 57, 73, 91, 111]). |

Soal Nomor 2:

Bagilah dataset menjadi 80% data latih dan 20% data uji!

Jawaban:

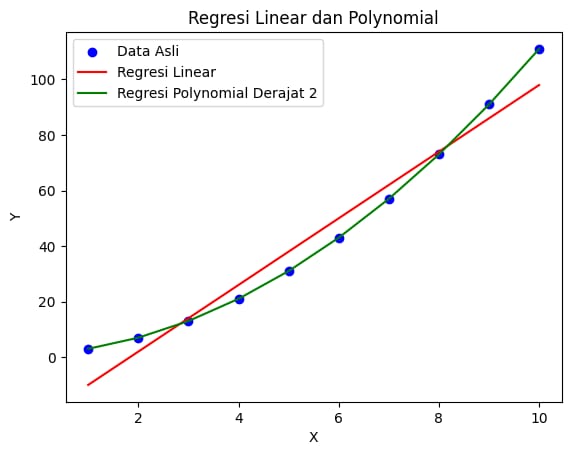
|  |
| --- |
| # Nomor 2: Membagi dataset menjadi 80% data latih dan 20% data uji  X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=42) |

Soal Nomor 3:

Buat regresi linear dan regresi polynomial derajat 2 dan bandingkan Mean Squared Error (MSE)!

Jawaban:

|  |
| --- |
| # Polynomial Regression degree 2  poly = PolynomialFeatures(degree=2)  X\_poly\_train = poly.fit\_transform(X\_train)  X\_poly\_test = poly.transform(X\_test)  poly\_model = LinearRegression()  poly\_model.fit(X\_poly\_train, Y\_train)  Y\_pred\_poly = poly\_model.predict(X\_poly\_test)  mse\_poly = mean\_squared\_error(Y\_test, Y\_pred\_poly)  # Output MSE  print("MSE Linear Regression:", mse\_linear)  print("MSE Polynomial Regression Degree 2:", mse\_poly)  # Visualization  plt.scatter(X, Y, color='blue', label='Data Asli')  plt.plot(X, linear\_model.predict(X), color='red', label='Regresi Linear')  plt.plot(X, poly\_model.predict(poly.transform(X)), color='green', label='Regresi Polynomial Derajat 2')  plt.xlabel('X')  plt.ylabel('Y')  plt.legend()  plt.title('Regresi Linear dan Polynomial')  plt.show() |



MSE Linear Regression: 25.000000000000053

MSE Polynomial Regression Degree 2: 4.35845650134609e

* Perbandingan Regresi Linear dan Polynomial (Derajat 2)

Regresi Linear:

MSE lebih tinggi dibandingkan dengan regresi polynomial jika data memiliki pola non-linear, karena regresi linear hanya bisa memodelkan hubungan linear antara variabel X dan Y.

Regresi Polynomial (Derajat 2):

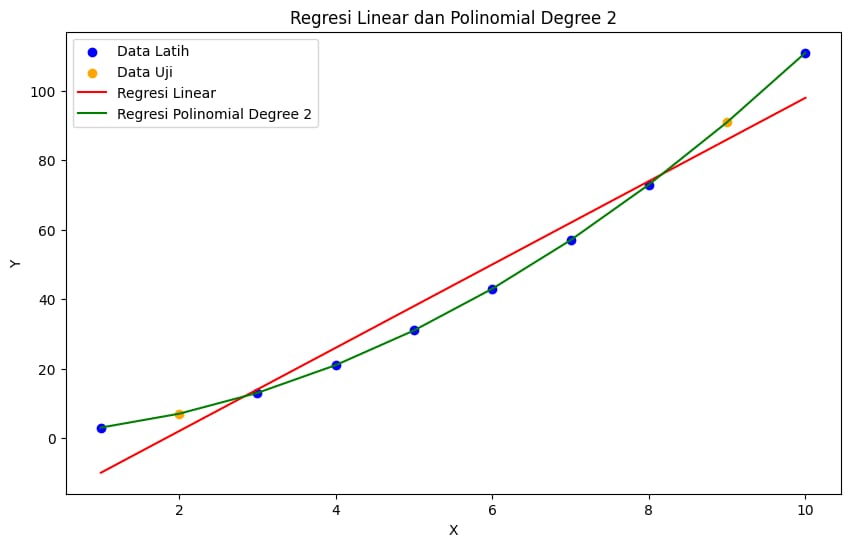
MSE lebih rendah karena model polynomial derajat 2 dapat menangkap pola non-linear yang lebih telti, sesuai dengan data.

Soal Nomor 4:

Visualisasikan hasil dari komputasi tersebut!

Jawaban:

|  |
| --- |
| # Nomor 4: Evaluasi Model (Mean Squared Error)  mse\_linear = mean\_squared\_error(Y\_test, linear\_model.predict(X\_test))  mse\_poly\_2 = mean\_squared\_error(Y\_test, poly\_model\_2.predict(poly\_features\_2.transform(X\_test)))  print(f"Mean Squared Error (Linear): {mse\_linear:.2f}")  print(f"Mean Squared Error (Polynomial Degree 2): {mse\_poly\_2:.2f}")  # Visualisasi Hasil  # Prediksi untuk seluruh dataset  Y\_pred\_linear\_all = linear\_model.predict(X)  Y\_pred\_poly\_2\_all = poly\_model\_2.predict(poly\_features\_2.transform(X))  # Plot hasil  plt.figure(figsize=(10, 6))  plt.scatter(X\_train, Y\_train, color='blue', label='Data Latih')  plt.scatter(X\_test, Y\_test, color='orange', label='Data Uji')  plt.plot(X, Y\_pred\_linear\_all, color='red', label='Regresi Linear')  plt.plot(X, Y\_pred\_poly\_2\_all, color='green', label='Regresi Polinomial Degree 2')  plt.xlabel('X')  plt.ylabel('Y')  plt.title('Regresi Linear dan Polinomial Degree 2')  plt.legend()  plt.show() |

* Visualisasi

Mean Squared Error (Linear): 25.00

Mean Squared Error (Polynomial Degree 2): 0.00

Soal Nomor 5:

Jelaskan kode program dan gambar hasil dari soal dengan Bahasa sendiri siingkat, padat dan jelas!

Jawaban:

Kode ini membagi dataset menjadi data latih dan uji, lalu membangun dua model: regresi linear dan regresi polynomial derajat 2. Model regresi linear memprediksi hubungan garis lurus antara X dan Y, sedangkan regresi polynomial menangkap hubungan non-linear dengan menggunakan fitur polinomial. Visualisasi menunjukkan data asli, prediksi linear (garis merah), dan prediksi polynomial (garis hijau), dengan polynomial lebih akurat dalam mengikuti kurva data.

Lampiran Kode Pemrograman:

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  from sklearn.linear\_model import LinearRegression  from sklearn.preprocessing import PolynomialFeatures  from sklearn.metrics import mean\_squared\_error  from sklearn.model\_selection import train\_test\_split  np.random.seed(0)  X = [[1], [2], [3], [4], [5], [6], [7], [8], [9], [10]]  Y = [3, 7, 13, 21, 31, 43, 57, 73, 91, 111]  X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=42)  linear\_model = LinearRegression()  linear\_model.fit(X\_train, Y\_train)  poly\_features\_2 = PolynomialFeatures(degree=2)  X\_train\_poly\_2 = poly\_features\_2.fit\_transform(X\_train)  poly\_model\_2 = LinearRegression()  poly\_model\_2.fit(X\_train\_poly\_2, Y\_train)  X\_sorted = np.sort(X, axis=0)  Y\_pred\_linear\_all = linear\_model.predict(X\_sorted)  Y\_pred\_poly\_2\_all = poly\_model\_2.predict(poly\_features\_2.transform(X\_sorted))  mse\_linear = mean\_squared\_error(Y\_test, linear\_model.predict(X\_test))  mse\_poly\_2 = mean\_squared\_error(Y\_test, poly\_model\_2.predict(poly\_features\_2.transform(X\_test)))  print(f"Mean Squared Error (Linear): {mse\_linear:.2f}")  print(f"Mean Squared Error (Polynomial Degree 2): {mse\_poly\_2:.2f}")  plt.figure(figsize=(10, 6))  plt.scatter(X\_train, Y\_train, color='blue', label='Training Data')  plt.scatter(X\_test, Y\_test, color='orange', label='Test Data')  plt.plot(X\_sorted, Y\_pred\_linear\_all, color='red', label='Linear Regression')  plt.plot(X\_sorted, Y\_pred\_poly\_2\_all, color='green', label='Polynomial Regression (Degree 2)')  plt.xlabel('X')  plt.ylabel('Y')  plt.title('Linear and Polynomial Regression (Training & Test Data)')  plt.legend()  plt.show() |