

PRAKTIUM FISIKA KOMPUTASI

MODUL 10-MACHINE LEARNING REGRESI LINEAR DAN POLINOMIAL

Disusun Oleh:

Dewi Rahmawati(1227030010)

1. Pada kode program untuk contoh ketiga dapat dimodifikasi dengan dataset $X = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ $Y = [3, 7, 13, 21, 31, 43, 57, 73, 91, 111]$, sehingga kode programnya menjadi.

```
X = [[1], [2], [3], [4], [5], [6], [7], [8], [9], [10]]
Y = [3, 7, 13, 21, 31, 43, 57, 73, 91, 111]
```

2. Kemudian pada kode program nilai untuk datasetnya dibagi menjadi 80% data latih dan 20% data uji, sehingga penulisan kode programnya menjadi.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=42)
```

3. Kemudian pada kode program dibuat model regresi linear dan regresi polynomial derajat 2 dan dibandingkan Mean Squared Error atau MSE nya, sehingga kode programnya :

```
linear_model = LinearRegression()
linear_model.fit(X_train, Y_train)

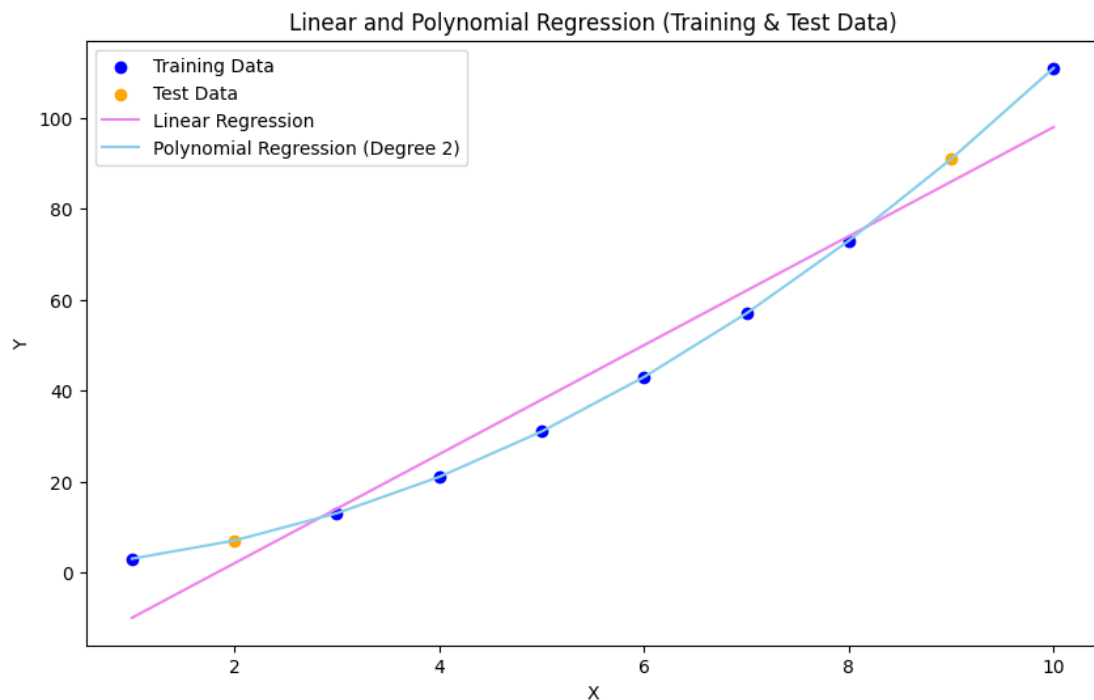
poly_features_2 = PolynomialFeatures(degree=2)
X_train_poly_2 = poly_features_2.fit_transform(X_train)
poly_model_2 = LinearRegression()
poly_model_2.fit(X_train_poly_2, Y_train)

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)))
```

Setelah dirunning dapat dilihat bahwa nilai MSE regresi linear lebih besar daripada MSE regresi polynomial dengan nilai MSE regresi linear 25.00 dan nilai MSE regresi polynomial 0.00.

4. Pada kode program dapat dilihat visualisasi data dari hasil komputasi tersebut yaitu



Mean Squared Error (Linear): 25.00

Mean Squared Error (Polynomial Degree 2): 0.00

5. Kode program dan penjelasan gambar

```
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
```

Kode program *machine learning* regresi linear dan polynomial diawali dengan mengimport *library* yang diperlukan seperti *library* numpy untuk perhitungan numerik, *library* matplotlib untuk membuat grafik visualisasi data yang dihasilkan, dan *library* sklearn untuk membuat model regresi linear dan regresi polynomial.

```
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]
```

Selanjutnya pada kode program dimasukkan nilai dataset X dan Y nya.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=42)
```

Setelah itu pada nilai dataset X dan Y dibuat kode program untuk membagi dataset menjadi 80% data latih dan 20% data uji yang dimuat dalam kode program “train_test_split(X, Y, test_size=0.2, random_state=42)”.

```
linear_model = LinearRegression()
linear_model.fit(X_train, Y_train)
```

Kemudian dibuat kode program untuk model regresi linear pada nilai dataset x dan y.

```
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))
```

Selanjutnya dibuat juga kode program untuk model regresi polynomial derajat 2 untuk nilai dataset x dan y.

```
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)))
```

Lalu dibuat kode program untuk nilai error pada masing-masing metode regresi linear dan regresi polynomial derajat 2.

```
print(f"Mean Squared Error (Linear): {mse_linear:.2f}")
print(f"Mean Squared Error (Polynomial Degree 2): {mse_poly_2:.2f}")
```

Selanjutnya ditampilkan nilai hasil perhitungan error untuk metode regresi linear dan regresi polynomial derajat 2.

```
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()
```

Selanjutnya dibuat plot grafik untuk visualisasi data.

Setelah itu kode programnya dapat di running dan dilihat plot grafik yang dihasilkannya. Pada hasil plot grafik dari data yang diperoleh melalui model regresi linear dan regresi polynomial derajat dua dapat dilihat bahwa model regresi linear berupa grafik linear yang tidak menghimpun seluruh data uji dan latih pada dataset. Sedangkan model regresi polynomial derajat 2 grafiknya berbentuk eksponensial dan menghimpun seluruh data uji dan latih data yang telah diberikan. Dapat dilihat juga pada nilai data error yang dihasilkan dari grafik. Untuk model regresi linear memiliki nilai error 25.00, sedangkan nilai error untuk model regresi polynomial derajat 2 nya yaitu sebesar 0.00. Dari hal tersebut dapat dianalisis bahwa model yang cukup baik dari kedua model tersebut untuk memprediksi data yaitu model polynomial derajat 2, karena dapat menghimpun nilai data uji dan data latihnya sehingga nilai error yang dihasilkan jauh lebih kecil dari pada nilai error pada model regresi linier.

Kode Program Lengkap

```
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()
```