

Selamat Datang di Colaborator x Untitled8.ipynb - Colab x +

colab.research.google.com/drive/1L1p1cDpXOKa\_SslcVGJ0-vRVAFFtliq?authuser=1#scrollTo=C9De4H7ibVYN

Untitled8.ipynb

File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan telah disimpan

+ Kode + Teks

RAM Disk + Gemini

[1] Kode yang disarankan mungkin tunduk kepada lisensi | Abhi-nand/Simple-Linear-Regression | franceoliver/candidate\_assignment

import numpy as np  
from sklearn.linear\_model import LinearRegression  
  
x = [[2], [4], [6], [8], [10], [12], [14], [16], [18], [20]]  
y = [1, 4, 7, 10, 13, 16, 19, 21, 24, 27]  
  
regr = LinearRegression().fit(x,y)  
regr.score(x,y)  
  
predict = np.array([[6]])  
  
print ("Prediksi")  
print ("Input = ", predict)  
print ("Output = ", regr.predict(predict))  
  
Prediksi  
Input = [[6]]  
Output = [7.01818182]

[2] Kode yang disarankan mungkin tunduk kepada lisensi | ANTOX/ANTsPyNet | dindacahya9/Python-Machine-Learning-Prediksi-Data-Regresi-Linear-dan-Polinomia

from sklearn.preprocessing import PolynomialFeatures  
from sklearn import linear\_model  
import numpy as np  
  
x = [[0], [4], [8], [12], [16], [20], [24], [28], [32], [36]]  
y = [0, 3, 8, 15, 24, 35, 48, 63, 80, 99]  
  
predict = np.array([[12]])  
poly = PolynomialFeatures(degree=2)

0 d selesai pada 11.06

Selamat Datang di Colaborator x Untitled8.ipynb - Colab x +

colab.research.google.com/drive/1L1p1cDpXOKa\_SslcVGJ0-vRVAFFtliq?authuser=1#scrollTo=C9De4H7ibVYN

Untitled8.ipynb

File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan telah disimpan

+ Kode + Teks

RAM Disk + Gemini

[2] Kode yang disarankan mungkin tunduk kepada lisensi | ANTOX/ANTsPyNet | dindacahya9/Python-Machine-Learning-Prediksi-Data-Regresi-Linear-dan-Polinomia

from sklearn.preprocessing import PolynomialFeatures  
from sklearn import linear\_model  
import numpy as np  
  
x = [[0], [4], [8], [12], [16], [20], [24], [28], [32], [36]]  
y = [0, 3, 8, 15, 24, 35, 48, 63, 80, 99]  
  
predict = np.array([[12]])  
poly = PolynomialFeatures(degree=2)  
x\_ = poly.fit\_transform(x)  
predict\_ = poly.fit\_transform(predict)  
regr = linear\_model.LinearRegression()  
regr.fit(x\_, y)  
  
print ("Prediksi")  
print ("Input = ", predict)  
print ("Output = ", regr.predict(predict\_))  
  
Prediksi  
Input = [[12]]  
Output = [15.]

[3] Kode yang disarankan mungkin tunduk kepada lisensi | stackoverflow.com/questions/64279832/not-getting-any-output-while-trying-to-run-the-fit-function-of-logisticregressio | Donitoo/IA

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

0 d selesai pada 11.06

```
Untitled8.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan telah disimpan

+ Kode + Teks

Kode yang disarankan mungkin tunduk kepada lisensi | stackoverflow.com/questions/64279832/not-getting-any-output-while-trying-to-run-the-fit-function-of-logisticregressio | Donitoo/IA

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

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.fit_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.fit_transform(X_test)))

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

✓ 0 d selesai pada 11.06
```

```
Untitled8.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan telah disimpan

+ Kode + Teks

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.fit_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.fit_transform(X_test)))

print(f"Mean Squared Error (Linear): {mse_linear}")
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='Data Latih')
plt.scatter(X_test, Y_test, color='green', label='Data Uji')
plt.plot(X_sorted, Y_pred_linear_all, color='red', label='Regresi Linear')
plt.plot(X_sorted, Y_pred_poly_2_all, color='orange', label='Regresi Polynomial Degree 2')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Regresi Linear dan Polynomial Degree 2 (Data Latih & Uji)')
plt.legend()
plt.show()

Mean Squared Error (Linear): 25.0000000000000053
Mean Squared Error (Polynomial Degree 2): 0.00

Regresi Linear dan Polynomial Degree 2 (Data Latih & Uji)

✓ 0 d selesai pada 11.06
```

