Analisis Regresi

Analisis regresi adalah teknik statistik yang digunakan untuk memodelkan hubungan antara variabel dependen (target) dan satu atau lebih variabel independen (prediktor). Tujuan utama dari analisis regresi adalah untuk memahami bagaimana perubahan pada variabel independen mempengaruhi variabel dependen dan membuat prediksi berdasarkan hubungan tersebut.

## Jenis-jenis Analisis Regresi

Terdapat beberapa jenis analisis regresi, antara lain:

* **Regresi Linear Sederhana:** Melibatkan satu variabel independen dan satu variabel dependen, dengan hubungan linear.
* **Regresi Polinomial:** Melibatkan satu variabel independen dan satu dependen, dengan hubungan polinomial (berbentuk kurva)
* **Regresi Logistik:** Melibatkan satu atau lebih variabel independen dan dependen kategorikal biner (dua kemungkinan hasil).

# Regresi Linear

Regresi linear adalah jenis analisis regresi yang paling umum digunakan. Tujuannya adalah untuk menemukan garis lurus terbaik (dalam kasus regresi linear sederhana) atau bidang (dalam kasus regresi linear berganda) yang paling sesuai dengan data.

Persamaan regresi linear sederhana:

y = b0 + b1x + e

dimana:

* y: variabel dependen (target)
* x: variabel independen (prediktor)
* b0: intercept (nilai y saat x = 0)
* b1: koefisien regresi (kemiringan garis)
* e: error (perbedaan antara nilai prediksi dan nilai aktual)

## Evaluasi Model Regresi

Matrik evaluasi yang umum digunakan untuk model regresi antara lain:

* **Mean Absolute Error (MSE):** Rata-rata nilai absolut dari kesalahan prediksi.
* **Mean Squared Error (MSE):** Rata-rata kuadrat dari kesalahan prediksi.
* **Root Mean Squared Error (RMSE):** Akar kuadrat dari MSE.
* **R-squared:** Mengukur proporsi varians variabel dependen yang dapat dijelaskan oleh model.

## Penerapan Regresi Linear dengan Scikit-Learn

Scikit-learn menyediakan modul Linear Regression untuk membangun model regresi linear.

**Tugas Hari 17**

Dataset ini berisi informasi tentang pengeluaran iklan (dalam ribuan dolar) di tiga media berbeda (TV, radio, dan koran) serta penjualan produk (dalam ribuan unit).

**Tugas**

1. **Membuat dan Visualisasikan Data:**

* Unduh dataset “Advertising” dari kaggle.
* Gunakan Pandas untuk membaca dataset.
* Buat scatter plot untuk memvisualisasikan hubungan antara masing-masing variabel pengeluaran iklan (TV, radio, newspaper) dengan penjualan (sales).

1. **Membuat Model Regresi Linear:**

* Gunakan Linear Regression dari scikit-learn untuk membuat model regresi linear berganda.
* Gunakan semua tiga variabel pengeluaran iklan sebagai fitur dan penjualan sebagai target.
* Latih model pada data.

1. **Evaluasi Model:**

* Hitung MAE, MSE, RMSE, dan R-squared dari model pada data yang sama yang digunakan untuk melatih.

**Contoh Kode (Pandas dan Scikit-Learn)**

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

# ... (kode untuk membaca dan mempersiapkan dataset Advertising)

# Scatter plot

fig, axs = plt.subplots(1, 3, figsize=(15, 5))

titanic\_df.plot(x='TV', y='sales', kind='scatter', ax=axs[0])

titanic\_df.plot(x='radio', y='sales', kind='scatter', ax=axs[1])

titanic\_df.plot(x='newspaper', y='sales', kind='scatter', ax=axs[2])

plt.show()

# Model regresi linear

model = LinearRegression()

model.fit(X, y)

# Evaluasi model

y\_pred = model.predict(X)

mae = mean\_absolute\_error(y, y\_pred)

mse = mean\_squared\_error(y, y\_pred)

rmse = np.sqrt(mse)

r2 = r2\_score(y, y\_pred)

print('MAE:', mae)

print('MSE:', mse)

print('RMSE:', rmse)

print('R-squared:', r2)

**Catatan:**

* Pastikan Anda sudah menginstall library scikit-learn.
* Sesuaikan path file CSV jika dataset Advertising.csv tidak berada di direktori yang sama dengan kode Anda.
* Untuk plot gunakan kind=scatter pada method plot di DataFrame pandas.
* Jangan lupa untuk memisahkan antara variabel independen/prediktor (X) dan variabel dependen/target (y).

***Selamat Mengerjakan Tugas! 😀***

**Sumber:**

1. [**is.muni.cz/th/fs06e/Numericke\_metody\_ve\_statistice.pdf**](https://is.muni.cz/th/fs06e/Numericke_metody_ve_statistice.pdf)
2. [**stackoverflow.com/questions/64459039/csv-file-does-not-exist**](https://stackoverflow.com/questions/64459039/csv-file-does-not-exist)
3. [**bardai.ai/artificial-intelligence/mastering-the-art-of-regression-evaluation-5-key-metrics-every-data-scientist-should-knowthe-residuals1-the-mean-squared-error-mse2-the-basis-mean-square-error-rmse3-the-mean-absolute-erro/**](https://bardai.ai/artificial-intelligence/mastering-the-art-of-regression-evaluation-5-key-metrics-every-data-scientist-should-knowthe-residuals1-the-mean-squared-error-mse2-the-basis-mean-square-error-rmse3-the-mean-absolute-erro/)

**Tugas:**

1. **Memuat dan Visualisasi Data:**

**Visualisasi Data (Scatter Plot)**

**X\_transformed = preprocessor.transform(X)**

**feature\_names = preprocessor.get\_feature\_names\_out()**

**# Membuat DataFrame baru untuk visualisasi**

**X\_vis = pd.DataFrame(X\_transformed, columns=feature\_names)**

**X\_vis['Sales'] = y.values**

**# Menentukan jumlah subplot sesuai dengan jumlah fitur yang ditransformasikan**

**num\_features = len(feature\_names)**

**fig, axs = plt.subplots(1, num\_features, figsize=(5 \* num\_features, 5)) # Sesuaikan ukuran figurnya**

**# Plotting scatter plot untuk setiap fitur**

**for i, feature in enumerate(feature\_names):**

**X\_vis.plot(x=feature, y='Sales', kind='scatter', ax=axs[i])**

**axs[i].set\_title(f'Hubungan antara {feature} dan Sales')**

**plt.tight\_layout()**

**plt.show()**

1. **Membuat Model Regresi Linear:**

**Membagi Data menjadi Data Latih dan Data Uji**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**# 5. Fit dan Transform Data**

**X\_train = preprocessor.fit\_transform(X\_train)**

**X\_test = preprocessor.transform(X\_test)**

**# 6. Membuat Model Regresi Linear**

**model = LinearRegression()**

**# 7. Melatih Model**

**model.fit(X\_train, y\_train)**

1. **Evaluasi Model:**

**Evaluasi Model**

**y\_pred = model.predict(X\_test)**

**mae = mean\_absolute\_error(y\_test, y\_pred)**

**mse = mean\_squared\_error(y\_test, y\_pred)**

**rmse = np.sqrt(mse)**

**r2 = r2\_score(y\_test, y\_pred)**

**print(f'MAE: {mae:.2f}')**

**print(f'MSE: {mse:.2f}')**

**print(f'RMSE: {rmse:.2f}')**

**print(f'R-squared: {r2:.2f}')**