Evaluasi Model Regresi (Lanjutan)

**Evaluasi Model dengan K-Fold Cross Validation:**

Selain MAE, MSE, RMSE, kita juga dapat menggunakan teknik validasi silang (cross-validation) untuk mengevaluasi kinerja model regresi.

### **Contoh Kode (K-Fold Cross-Validation):**

from sklearn.model\_selection import cross\_val\_score

# ... (kode untuk membaca, mengolah data, dan membuat model)

# Validasi Silang untuk setiap model

models = [model, ridge\_model, lasso\_model, elastic\_model]

model\_names = ['Linear Regression', 'Ridge Regression', 'Lasso Regression', 'Elastic Net']

scoring = ['neg\_mean\_absolute\_error', 'neg\_mean\_squared\_error', 'neg\_root\_mean\_squared\_error']

for model, name in zip(models, model\_names):

scores = cross\_val\_score(model, X, y, cv=5, scoring=scoring)

print(f"\n{name}:")

print(f'MAE: {-scores[:, 0].mean():.2f} (+/- {scores[:, 0].std():.2f})')

print(f'MSE: {-scores[:, 1].mean():.2f} (+/- {scores[:, 1].std():.2f})')

print(f'RMSE: {-scores[:, 2].mean():.2f} (+/- {scores[:, 2].std():.2f})')

# Interpretasi Hasil Cross-Validation:

Hasil cross-validation memberikan estimasi kinerja yang lebih daripada evaluasi pada satu pembagian data latih dan uji. Anda dapat membandingkan rata-rata dan standar deviasi dari MAE, MSE, dan RMSE untuk masing-masing model. Model dengan rata-rata MAE, MSE, dan RMSE yang lebih rendah dan standar deviasi yang lebih kecil umumnya dianggap lebih baik.

# Pemilihan Model dengan Grid Search:

Kita dapat menggunakan Grid Search untuk menemukan kombinasi hyperparameter terbaik untuk model regresi. Grid Search mencoba semua kombinasi hyperparameter yang mungkin dan mengevaluasi kinerja model pada setiap kombinasi menggunakan cross-validation.

### **Contoh kode (Grid Search untuk Ridge Regression)**

from sklearn.model\_selection import GridSearchCV

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

# Define parameter grid

param\_grid = {'alpha': [0.1, 1.0, 10.0]}

# Membuat GridSearchCV object

grid\_search = GridSearchCV(estimator=ridge\_model, param\_grid=param\_grid, cv=5, scoring='neg\_mean\_absolute\_error')

# Fit ke data

grid\_search.fit(X\_train, y\_train)

# Print hyperparameter terbaik

print(f"Hyperparameter terbaik: {grid\_search.best\_params\_}")

**Tugas Hari 21:**

1. Lakukan K-Fold Cross-Validation untuk keempat model (Linear Regression, Ridge, Lasso, Elastic Net) pada dataset “Advertising” menggunakan scoring=’neg\_mean\_absolute\_error’ , scoring=’neg\_mean\_squared\_error’ , dan scoring=’neg\_root\_mean\_squared\_error’ . **Bandingkan Hasilnya.**
2. Lakukan Grid Search untuk mencari hyperparameter alpha terbaik untuk Ridge Regression dengan param\_grid={‘alpha’: [0.1, 1.0, 10.0]} dan scoring=’neg\_mean\_absolute\_error’.

**Catatan:**

* Install scikit-learn
* File csv (Advertising.csv)

**Selamat Mengerjakan Tugas! 😖**

**Tugas**

1. **K-Fold Cross-Validation 4 Model**

from sklearn.model\_selection import cross\_val\_score

import numpy as np

models = [model, ridge\_model, lasso\_model, elastic\_model]

model\_names = ['Linear Regression', 'Ridge Regression', 'Lasso Regression', 'Elastic Net']

scoring = ['neg\_mean\_absolute\_error', 'neg\_mean\_squared\_error', 'neg\_root\_mean\_squared\_error']

for model, name in zip(models, model\_names):

scores = []

for score in scoring:

cv\_scores = cross\_val\_score(model, X, y, cv=5, scoring=score)

scores.append(cv\_scores)

scores = np.array(scores).T

print(f"\n{name}: ")

print(f'MAE: {-scores[:, 0].mean():.2f} (+/- {scores[:, 0].std():.2f})')

print(f'MSE: {-scores[:, 1].mean():.2f} (+/- {scores[:, 1].std():.2f})')

print(f'RMSE: {-scores[:, 2].mean():.2f} (+/- {scores[:, 2].std():.2f})')

1. **Grid Search, mencari hyperparameter**

from sklearn.model\_selection import GridSearchCV

from sklearn.linear\_model import Ridge

# Definisikan parameter grid untuk Ridge Regression

param\_grid = {

'alpha': [0.1, 1.0, 10.0, 100.0],

'fit\_intercept': [True, False],

'solver': ['auto', 'svd', 'cholesky', 'lsqr', 'sparse\_cg', 'sag', 'saga']

}

# Inisialisasi model Ridge Regression

ridge\_model = Ridge()

# Inisialisasi GridSearchCV

grid\_search = GridSearchCV(estimator=ridge\_model, param\_grid=param\_grid, cv=5, scoring='neg\_mean\_squared\_error')

# Melakukan Grid Search

grid\_search.fit(X, y)

# Mendapatkan hasil terbaik

best\_params = grid\_search.best\_params\_

best\_score = -grid\_search.best\_score\_ # Ubah menjadi positif untuk interpretasi lebih baik

print(f"Best parameters: {best\_params}")

print(f"Best MSE score: {best\_score:.2f}")

# Penjelasan Kode (K-Fold Cross-Validation):

1. **Inisialisasi Model dan Scoring:**
   * Anda mendefinisikan daftar models yang berisi model-model regresi yang akan dievaluasi.
   * Anda mendefinisikan daftar model\_names yang berisi nama-nama model yang sesuai.
   * Anda mendefinisikan daftar scoring yang berisi metrik evaluasi yang akan digunakan (neg\_mean\_absolute\_error, neg\_mean\_squared\_error, dan neg\_root\_mean\_squared\_error).
2. **Looping untuk Setiap Model dan Metrik:**
   * Anda menggunakan loop for untuk melakukan cross-validation untuk setiap model dan setiap metrik evaluasi.
   * Pada setiap iterasi, Anda:
     + Melakukan 5-fold cross-validation (cv=5) menggunakan cross\_val\_score.
     + Menyimpan hasil cross-validation dalam list scores.
   * Setelah loop selesai, Anda mengubah scores menjadi array NumPy dan menampilkan rata-rata dan standar deviasi dari setiap metrik untuk model tersebut.

# Penjelasan Kode (Grid Search):

1. **Parameter Grid:**
   * Anda mendefinisikan param\_grid, yaitu kamus yang berisi hyperparameter yang ingin Anda coba dan nilai-nilai yang mungkin untuk setiap hyperparameter.
   * Anda mencoba beberapa nilai alpha (parameter regularisasi) dan fit\_intercept untuk melihat bagaimana pengaruhnya terhadap kinerja model. Anda juga mencoba beberapa solver yang berbeda untuk menemukan solver terbaik untuk kasus ini.
2. **Grid Search:**
   * Anda membuat objek GridSearchCV untuk melakukan grid search.
   * estimator=ridge\_model: Model yang akan digunakan adalah model Ridge Regression.
   * param\_grid=param\_grid: Parameter grid yang sudah didefinisikan sebelumnya.
   * cv=5: Menggunakan 5-fold cross-validation untuk mengevaluasi setiap kombinasi hyperparameter.
   * scoring='neg\_mean\_squared\_error': Metrik evaluasi yang digunakan adalah MSE (Mean Squared Error). Tanda negatif digunakan karena scikit-learn menganggap skor yang lebih tinggi lebih baik, sedangkan MSE yang lebih rendah lebih baik.
3. **Fit dan Hasil Terbaik:**
   * Anda menjalankan grid search dengan .fit(X, y). Ini akan mencoba semua kombinasi hyperparameter dan mengevaluasinya menggunakan cross-validation.
   * Anda mencetak hyperparameter terbaik (best\_params\_) dan skor MSE terbaik (best\_score\_). Skor MSE diubah menjadi positif untuk memudahkan interpretasi.

**Interpretasi Hasil:**

* **K-Fold Cross-Validation:** Hasil cross-validation akan menunjukkan bagaimana kinerja setiap model bervariasi pada fold yang berbeda. Ini memberikan gambaran yang lebih baik tentang seberapa baik model dapat digeneralisasi pada data baru dibandingkan dengan evaluasi pada satu pembagian data latih dan uji. Anda dapat membandingkan rata-rata dan standar deviasi dari MAE, MSE, dan RMSE untuk memilih model terbaik.
* **Grid Search:** Hasil grid search akan menunjukkan kombinasi hyperparameter terbaik untuk model Ridge Regression berdasarkan metrik evaluasi MSE. Anda dapat menggunakan hyperparameter ini untuk membangun model final Anda.

**Catatan:**

Jika Anda memiliki lebih banyak waktu dan sumber daya komputasi, Anda dapat mencoba nilai hyperparameter yang lebih banyak dan lebih beragam dalam grid search.