Evaluasi Model Regresi Lanjutan (Bagian 2)

Melanjutkan Pembahasan kemarin, hari ini kita akan fokus pada evaluasi model regresi dengan menggunakan teknik regularisasi dan validasi silang (cross-validation).

# Regularisasi dalam Regresi

Seperti yang telah dibahas sebelumnya, regularisasi adalah teknik penting untuk mencegah overfitting dalam model regresi. Ada tiga jenis regularisasi yang umum digunakan:

## L1 Regularization (LASSO):

* Menambahkan penalti L1 pada koefisien regresi, mendorong beberapa koefisien menjadi nol (pemilihan fitur)
* Berguna jika Anda memiliki banyak fitur dan ingin memilih fitur yang paling penting.

## L2 Regularization (Ridge):

* Menambahkan penalti L2 pada koefisien regresi, mencegah koefisien menjadi terlalu besar.
* Berguna jika Anda memiliki multikolinearitas (Korelasi tinggi antara fitur).

## Elastic Net:

* Kombinasi dari penalti L1 dan L2.
* Memberikan keseimbangan antara pemilihan fitur (L1) dan pencegahan koefisien besar (L2).

# Validasi Silang (Cross-Validation)

Validation silang adalah teknik untuk mengevaluasi kinerja model dengan cara membagi dataset menjadi beberapa bagian (fold). Model dilatih pada beberapa fold dan data diuji pada fold lainnya. Proses ini diulang beberapa kali, dan hasil evaluasi dirata-ratakan untuk mendapatkan estimasi kinerja yang lebih baik.

# Menggunakan Regularisasi dan Cross-Validation

Kita dapat menggabungkan regularisasi dan cross-validation untuk mencari model terbaik dan hyperparameter yang optimal.

**Tugas Hari 25:**

Lanjutkan dengan dataset “Advertising”

**Tugas:**

1. **Grid Search dengan Regularisasi:**

* Lakukan grid search untuk menemukan hyperparameter terbaik (alpha) untuk model Ridge dan Lasso.
* Gunakan 5-fold cross-validation dan metrik evaluasi MAE (Mean Absolute Error).

1. **Evaluasi Model Terbaik:**

* Latih model Ridge dan Lasso dengan hyperparameter terbaik yang ditemukan dari grid search.
* Evaluasi kinerja kedua model pada data uji menggunakan MAE, MSE, dan RMSE.
* Bandingkan kinerja kedua model dan pilih model terbaik.

**Contoh Kode (Scikit-Learn):**

from sklearn.linear\_model import Ridge, Lasso

from sklearn.model\_selection import GridSearchCV

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

# Grid Search untuk Ridge Regression

param\_grid = {'alpha': [0.01, 0.1, 1, 10, 100]}

ridge = Ridge()

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

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

print("Ridge - Hasil Grid Search:")

print("Hyperparameter terbaik:", grid\_search.best\_params\_)

print("MAE terbaik:", -grid\_search.best\_score\_)

# Grid Search untuk Lasso Regression

lasso = Lasso()

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

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

print("\nLasso - Hasil Grid Search:")

print("Hyperparameter terbaik:", grid\_search.best\_params\_)

print("MAE terbaik:", -grid\_search.best\_score\_)

# ... (latih dan evaluasi model Ridge dan Lasso terbaik)

**Selamat Mengerjakan Tugas! 🥲**

**Wulan Resmi Jadian 17/06/2024 😔**

**Tugas**

1. **Grid Search dengan Regularisasi:**

**from sklearn.linear\_model import Ridge, Lasso**

**from sklearn.model\_selection import GridSearchCV**

**# Grid Search untuk Ridge Regression**

**param\_grid = {'alpha': [0.01, 0.1, 1, 10, 100]}**

**ridge = Ridge()**

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

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

**print("Ridge - Hasil Grid Search:")**

**print("Hyperparameter terbaik:", grid\_search.best\_params\_)**

**print("MAE terbaik:", -grid\_search.best\_score\_)**

**# Grid Search untuk Lasso Regression**

**lasso = Lasso()**

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

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

**print("\nLasso - Hasil Grid Search:")**

**print("Hyperparameter terbaik:", grid\_search.best\_params\_)**

**print("MAE terbaik:", -grid\_search.best\_score\_)**

**# ... (Latih dan Evaluasi Model Ridge dan Lasso terbaik!)**

1. **Evaluasi Model Terbaik:**

**# ... (Latih dan Evaluasi Model Ridge dan Lasso terbaik!)**

**# Ridge Regression dengan Hyperparameter Terbaik**

**best\_ridge\_model = Ridge(alpha=grid\_search.best\_params\_['alpha'])**

**best\_ridge\_model.fit(X\_train, y\_train)**

**y\_pred\_ridge = best\_ridge\_model.predict(X\_test)**

**# Lasso Regression dengan Hyperparameter Terbaik**

**best\_lasso\_model = Lasso(alpha=grid\_search.best\_params\_['alpha'])**

**best\_lasso\_model.fit(X\_train, y\_train)**

**y\_pred\_lasso = best\_lasso\_model.predict(X\_test)**

**# Evaluasi Model Ridge Terbaik**

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

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

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

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

**print("\nRidge Regression (dengan hyperparameter terbaik):")**

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

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

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

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

**# Evaluasi Model Lasso Terbaik**

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

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

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

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

**print("\nRidge Regression (dengan hyperparameter terbaik):")**

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

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

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

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

**Penjelasan Kode:**

1. **Membuat Model dengan Hyperparameter Terbaik:**
   * best\_ridge\_model = Ridge(alpha=grid\_search.best\_params\_['alpha']): Membuat model Ridge dengan menggunakan nilai alpha terbaik yang ditemukan oleh Grid Search.
   * best\_ridge\_model.fit(X\_train, y\_train): Melatih model Ridge pada data latih.
   * Langkah yang sama dilakukan untuk membuat dan melatih model Lasso terbaik.
2. **Evaluasi Model Terbaik:**
   * y\_pred\_ridge = best\_ridge\_model.predict(X\_test): Membuat prediksi pada data uji menggunakan model Ridge terbaik.
   * mae\_ridge = mean\_absolute\_error(y\_test, y\_pred\_ridge): Menghitung MAE untuk model Ridge terbaik.
   * Langkah yang sama dilakukan untuk menghitung MSE, RMSE, dan R-squared untuk model Ridge dan Lasso terbaik.

**Hasil dan Analisis:**

Setelah menjalankan kode ini, Anda akan mendapatkan hasil evaluasi model Ridge dan Lasso dengan hyperparameter terbaik. Bandingkan hasil MAE, MSE, dan RMSE untuk kedua model. Model dengan nilai metrik evaluasi yang lebih rendah dapat dianggap lebih baik dalam hal kinerja prediksi.

**Langkah Selanjutnya:**

Setelah memilih model terbaik, Anda dapat menggunakan model tersebut untuk membuat prediksi pada data baru atau melakukan analisis lebih lanjut, seperti menginterpretasikan koefisien regresi untuk memahami pengaruh setiap variabel prediktor terhadap variabel target.