Analisa Regression Model

Nama : Atanasius Pradiptha Sampurno
Kelas : TK-45-02
NIM : 1103213036

1. Jika model *linear regression* atau *decision tree* mengalami *underfitting* pada dataset ini, strategi apa yang akan digunakan untuk meningkatkan performanya? Bandingkan setidaknya dua pendekatan berbeda (misal: transformasi fitur, penambahan *features*, atau perubahan model ke algoritma yang lebih kompleks), dan jelaskan bagaimana setiap solusi memengaruhi *bias-variance tradeoff*!

   Jawaban :

   Jika model seperti *Linear Regression* atau *Decision Tree* mengalami *underfitting*, artinya model tidak cukup kompleks untuk menangkap pola yang ada dalam data. Salah satu pendekatan yang dapat digunakan adalah transformasi fitur, seperti menambahkan fitur polinomial (x^2, x^3, dst). Transformasi ini dapat menggambarkan data dengan lebih detail sehingga model linear pun dapat menangkap hubungan non-linear. Secara matematis, model akan berubah dari bentuk linear sederhana $y = \beta_0 + \beta_1 x$ menjadi bentuk polynomial $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x$.

   Pendekatan lain adalah mengganti model ke algoritma yang lebih kompleks seperti *Random Forest* atau *Gradient Boosting*. Model-model ini menggunakan gabungan dari banyak pohon keputusan, sehingga mampu menangkap interaksi antar fitur dan pola yang lebih dalam.

   Keduanya memengaruhi *bias-variance tradeof,* dimana transformasi fitur atau penggunaan model kompleks akan mengurangi bias (model jadi lebih fleksibel), namun bisa meningkatkan varians (risiko *overfitting*). Karena itu, perlu strategi validasi seperti *cross-validation* untuk mengontrol generalisasi model terhadap data baru.

2. Selain MSE, jelaskan dua alternatif *loss function* untuk masalah regresi (misal: MAE, *Huber Loss*) dan bandingkan keunggulan serta kelemahannya. Dalam skenario apa setiap *loss function* lebih cocok digunakan? (Contoh: data dengan outlier, distribusi target non-Gaussian, atau kebutuhan interpretasi model).

Jawaban:

Selain *Mean Squared Error* (MSE), terdapat dua fungsi loss yang juga sering digunakan dalam regresi, yaitu *Mean Absolute Error* (MAE) dan *Huber Loss*.

a. *Mean Absolute Error* (MAE) menghitung rata-rata dari selisih absolut antara nilai aktual dan prediksi dengan menggunakan rumus [$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$]. MAE lebih tahan terhadap pengaruh *outlier* karena tidak melibatkan pangkat dua pada *error,* sehingga kesalahan yang ekstrem tidak terlalu mendominasi nilai akhir. Namun, kelemahannya adalah model bisa kurang peka terhadap kesalahan besar, karena semua *error* diperlakukan sama.

b. *Huber Loss* merupakan kompromi antara MSE dan MAE. Fungsi ini bersifat kuadratik untuk error kecil, namun berubah menjadi linier jika error melebihi ambang tertentu (δ) yang dapat dihitung dengan menggunakan rumus :

$$L_\delta(a) = \begin{cases} \frac{1}{2} a^2 \; jika \; |a| \leq \delta \\ \delta |a| \; jika \; |a| > \delta \end{cases}$$

Fungsi ini berguna ketika data mengandung sebagian *outlier*, tetapi kita tetap ingin mempertahankan sensitivitas terhadap error kecil. Dengan begitu, Huber Loss bisa menghasilkan model yang lebih seimbang dalam hal bias dan robust terhadap noise ekstrem.

3. Tanpa mengetahui nama fitur, metode apa yang dapat digunakan untuk mengukur pentingnya setiap fitur dalam model? Jelaskan prinsip teknikal di balik metode tersebut (misal: koefisien regresi, feature importance berdasarkan impurity reduction) serta keterbatasannya!

Jawaban:

Meskipun nama fitur tidak diketahui, kita masih dapat menilai seberapa besar kontribusi tiap fitur terhadap prediksi model menggunakan beberapa metode teknis:

a. Pada *Linear Regression*, pentingnya fitur dapat dilihat dari *koefisien regresinya*. Koefisien β_i menunjukkan perubahan yang diharapkan pada target y untuk setiap perubahan satu unit pada fitur x_i,dengan asumsi fitur lain tetap. Namun, jika fitur memiliki skala yang berbeda, interpretasi bisa menyesatkan, sehingga normalisasi sangat disarankan.

b. Pada *Decision Tree* atau model pohon lainnya, kita bisa menggunakan *feature importance berdasarkan impurity reduction*. Konsep ini menghitung seberapa besar pengurangan error (seperti MSE) yang dihasilkan setiap fitur saat memecah data:

$$Feature\ Importance\ (j) = \sum_{Node\ menggunakan\ x} Reduction\ in\ Imputurity$$

Namun metode ini dapat bias terhadap fitur dengan banyak kemungkinan nilai (seperti fitur kontinu atau kategori dengan banyak kelas).

4. Bagaimana mendesain eksperimen untuk memilih hyperparameter optimal (misal: learning rate untuk SGDRegressor, max_depth untuk Decision Tree) pada dataset ini? Sertakan analisis tradeoff antara komputasi, stabilitas pelatihan, dan generalisasi model!

Jawaban:

Untuk mendapatkan nilai hyperparameter terbaik (misalnya: learning_rate pada *SGDRegressor* atau max_depth pada *Decision Tree*), kita bisa menggunakan pendekatan seperti *Grid Search* atau *Randomized Search* yang dikombinasikan dengan *cross-validation* (CV). Strategi ini membagi data pelatihan ke dalam beberapa lipatan (folds), lalu model diuji berkali-kali untuk menghindari overfitting pada data tertentu. Akan tetapi ada beberapa hal yang perlu diperhatikan, seperti:

- Akurasi dan Generalisasi: Cross-validation membantu menilai performa model pada data yang belum pernah dilihat, mendorong model untuk generalisasi lebih baik.

- Biaya Komputasi: Grid Search sangat mahal secara waktu jika jumlah hyperparameter dan kombinasi besar. Randomized Search bisa jadi alternatif lebih efisien.
- Stabilitas Pelatihan: Jika hyperparameter tidak optimal (misalnya learning rate terlalu besar), proses pelatihan bisa tidak konvergen atau menghasilkan prediksi tidak stabil.

5. Jika menggunakan model linear regression dan residual plot menunjukkan pola non-linear serta heteroskedastisitas, langkah-langkah apa yang akan diambil? (contohnya: Transformasi data/ubah model yang akan dipakai/etc)

Jawaban:

Jika residual plot dari *Linear Regression* menunjukkan pola yang tidak acak atau terdapat peningkatan variasi error seiring naiknya nilai prediksi (disebut heteroskedastisitas), maka ini menandakan pelanggaran terhadap asumsi dasar regresi linear, yaitu error harus memiliki variansi konstan dan hubungan antara fitur dan target harus linear. Untuk mengatasi hal ini, beberapa strategi yang dapat diterapkan adalah:

a. Untuk menangani hubungan non-linear, kita bisa mengubah fitur menggunakan logaritma, akar kuadrat, atau bentuk pangkat lainnya. Misalnya, jika hubungan antara x dan y tampak eksponensial, kita bisa mencoba model seperti $y = \beta_0 + \beta_1 \log(x) + \varepsilon$. Transformasi ini bertujuan untuk membuat hubungan menjadi lebih linear dan stabil dari sisi variansi.
b. Bila error tampak membesar pada nilai target yang tinggi, kita bisa menerapkan transformasi seperti log(y) untuk menstabilkan variasinya.
c. Model seperti *Polynomial Regression* atau *Decision Tree* mampu menangkap pola non-linear lebih baik daripada *Linear Regression* standar.
d. Alternatif lainnya adalah memakai pendekatan seperti *Generalized Least Squares* (GLS) atau *regresi robust*, yang secara eksplisit memperhitungkan perubahan variansi residual (non-homoskedastisitas).