

ANALISA REGRESSION MODEL

1. Strategi Mengatasi Underfitting pada Linear Regression/Decision Tree

Underfitting terjadi saat model terlalu sederhana untuk menangkap pola dalam data, ditandai dengan performa buruk di data pelatihan maupun pengujian.

a. Transformasi Fitur (Feature Engineering)

- **Contoh:** Menambahkan polynomial features untuk linear regression.
- **Efek:** Mengurangi bias karena model bisa menangkap relasi non-linear.
- **Tradeoff:** Variance bisa meningkat karena model menjadi lebih kompleks, berpotensi overfitting jika tidak di-regularisasi.

b. Beralih ke Model Lebih Kompleks

- **Contoh:** Ganti linear regression ke Random Forest atau Gradient Boosting.
- **Efek:** Menurunkan bias karena model mampu menangkap interaksi dan non-linearitas yang lebih kompleks.
- **Tradeoff:** Variance dan waktu komputasi meningkat, perlu regularisasi dan tuning yang lebih hati-hati.

2. Alternatif Loss Function untuk Regresi: MAE vs Huber Loss

Loss Function	Keunggulan	Kelemahan	Cocok Untuk
MAE (Mean Absolute Error)	Robust terhadap outlier, tidak memperbesar penalti	Tidak smooth (gradien tidak kontinu di 0), lebih sulit dioptimalkan	Data dengan banyak outlier
Huber Loss	Kombinasi MSE (kuadrat kecil) dan MAE (linear besar), stabil	Perlu memilih threshold delta secara hati-hati	Situasi dengan outlier tetapi tetap ingin sensitivitas terhadap kesalahan kecil
<ul style="list-style-type: none">• Huber loss ideal saat ingin stabilitas MSE namun tetap toleran terhadap outlier ekstrim.• MAE cocok bila interpretasi median lebih relevan daripada mean.			

3. Metode Menilai Pentingnya Fitur tanpa Nama Fitur

a. Koefisien pada Linear Regression

- **Prinsip:** Besar kecilnya koefisien (dengan asumsi fitur telah dinormalisasi) menunjukkan pengaruh fitur terhadap target.
- **Keterbatasan:** Sensitif terhadap multikolinearitas; tidak mampu tangkap interaksi antar fitur.

b. Feature Importance dari Tree-Based Models

- **Prinsip:** Mengukur penurunan impurity (seperti MSE) setiap kali fitur dipakai untuk split.
- **Keterbatasan:** Bias terhadap fitur kategorikal dengan banyak level atau fitur dengan skala lebih besar.

c. Permutation Importance (Model Agnostic)

- **Prinsip:** Mengacak nilai satu fitur dan melihat penurunan performa model.
 - **Kelebihan:** Bisa diterapkan pada model apapun.
 - **Keterbatasan:** Mahal secara komputasi, bisa bias jika fitur saling berkorelasi.
-

4. Desain Eksperimen untuk Hyperparameter Tuning

a. Grid Search atau Random Search + Cross-Validation

- Gunakan k-fold CV untuk menilai stabilitas generalisasi.
- **Random Search** lebih efisien untuk ruang parameter besar.
- **Bayesian Optimization** (mis. dengan Optuna) dapat lebih efisien secara statistik.

b. Tradeoff Analisis

- **Komputasi:** Grid search mahal jika banyak parameter dan nilai.
 - **Stabilitas:** Cross-validation meningkatkan estimasi stabil, tetapi mahal secara waktu.
 - **Generalisasi:** Hindari overfitting pada validation set dengan nested CV.
-

5. Penanganan Pola Non-Linear & Heteroskedastisitas pada Residual Plot

a. Transformasi Variabel

- **Contoh:** Log-transformasi pada target jika varian residual meningkat dengan nilai target.
- **Efek:** Menstabilkan varians, mengurangi heteroskedastisitas.

b. Gunakan Model Non-Linear

- Ganti Linear Regression dengan model non-linear seperti:
 - **Polynomial Regression**
 - **Tree-based models**
 - **Kernel Methods (SVR with RBF kernel)**

c. Modeling Variance secara Eksplisit

- Gunakan model heteroskedastik seperti **Generalized Least Squares (GLS)**.

d. Residual Diagnostics

- Plot studentized residuals, uji White test atau Breusch-Pagan untuk memastikan heteroskedastisitas.
-