Nama/NIM : Rafli Limandijaya/1103210243

BAGIAN 1

A. ANALISIS SEMUA MODEL

1. Linear Regression

Linear Regression adalah model statistik dasar yang digunakan untuk memprediksi nilai dari sebuah variabel target (Y) berdasarkan satu atau lebih variabel input (X). Hubungan antara X dan Y diasumsikan linear.

2. Polynomial Regression (degree=2)

Ini adalah perluasan dari linear regression, di mana hubungan antara variabel input dan target dapat berbentuk kurva (non-linear). Dengan degree=2, kamu memasukkan kuadrat dari variabel input sebagai fitur.

3. Decision Tree Regressor

Model berbasis pohon yang membagi data ke dalam cabang berdasarkan fitur-fitur hingga mencapai prediksi nilai target. Tidak perlu asumsi linearitas atau normalitas.

4. K-Nearest Neighbors (KNN) Regressor

Model ini memprediksi nilai target suatu titik berdasarkan rata-rata nilai target dari k tetangga terdekat dalam ruang fitur.

5. Bagging Regressor

"Bagging" (Bootstrap Aggregating) adalah teknik ensemble yang membuat banyak model (biasanya decision tree) dengan data yang di-*resample*, lalu menggabungkan hasilnya (misalnya rata-rata).

6. AdaBoost Regressor

AdaBoost (Adaptive Boosting) membangun model secara bertahap. Tiap model baru lebih fokus pada data yang sebelumnya diprediksi buruk oleh model sebelumnya.

B. ANALISIS MODEL TERBAIK

1. Analisis Model Terbaik: Polynomial Regression (degree=2)

Polynomial Regression memperluas Linear Regression dengan menambahkan fitur polinomial (kuadrat), memungkinkan model menangkap hubungan non-linier.

Kelebihan:

- Dapat menangkap hubungan non-linier yang sederhana
- Tetap mempertahankan interpretabilitas
- Komputasi relatif cepat

Keterbatasan:

- Mudah overfit pada derajat polinomial tinggi
- Sensitif terhadap outlier
- Peningkatan dimensi yang signifikan (curse of dimensionality)

2. Evaluasi metrik untuk semua model:

Linear Regression:

- MSE: 94.8549
- RMSE: 9.7393
- MAE: 6.9468
- R-squared: 0.2009

Polynomial Regression (degree=2):

- MSE: 88.0869
- RMSE: 9.3855
- MAE: 6.6468
- R-squared: 0.2579

Decision Tree:

- MSE: 178.0124
- RMSE: 13.3421
- MAE: 9.0312
- R-squared: -0.4996

KNN:

- MSE: 93.3519

- RMSE: 9.6619

- MAE: 6.8709

- R-squared: 0.2136

Bagging Regressor:

- MSE: 92.8746

- RMSE: 9.6371

- MAE: 6.8464

- R-squared: 0.2176

AdaBoost:

- MSE: 144.6210

- RMSE: 12.0258

- MAE: 10.2933

- R-squared: -0.2183

C. ANALISIS KOMPREHENSIF MODEL TERBAIK

Polynomial Regression (degree=2) memberikan performa terbaik baik dari segi RMSE maupun R-squared.

Polynomial Regression (degree=2) memberikan performa terbaik karena:

1. Struktur model yang sesuai dengan pola data yang ada
2. Kemampuan untuk menangkap hubungan penting antara fitur dan target
3. Keseimbangan yang baik antara bias dan varians
4. Optimalisasi yang efektif dari fungsi loss

D. KESIMPULAN

Berdasarkan analisis yang dilakukan:

1. Model Polynomial Regression (degree=2) memberikan performa prediksi terbaik dengan R-squared 0.2579 dan RMSE 9.3855.

2. Feature selection berbasis Mutual Information sangat penting untuk dataset ini karena dapat menangkap hubungan non-linier antara fitur dan target.

3. Penanganan multikolinearitas dengan menghapus fitur yang berkorelasi tinggi membantu meningkatkan stabilitas model.

4. Standardisasi fitur sangat penting untuk model seperti SVR dan KNN yang sensitif terhadap skala.

5. Residual plot dan visualisasi actual vs predicted memberikan wawasan tentang kualitas prediksi dan pola kesalahan model.

BAGIAN 2

1. Jika Linear Regression atau Decision Tree mengalami underfitting, strategi apa untuk meningkatkan performa?

Pendekatan 1: Transformasi Fitur (contoh: Polynomial Features)

Apa yang dilakukan: Menambahkan fitur polinomial (derajat 2, 3, dst.) untuk memperkaya representasi data.

Dampak ke bias-variance:

Bias menurun: Model jadi lebih fleksibel, bisa menangkap hubungan non-linear.

Variance meningkat: Ada risiko overfitting kalau derajat terlalu tinggi.

Relevansi: Ini terbukti di percobaan, Polynomial Regression degree=2 meningkatkan R^2 dibanding Linear Regression biasa.

Pendekatan 2: Ganti Model ke Algoritma Lebih Kompleks (contoh: Random Forest, Gradient Boosting)

Apa yang dilakukan: Pindah dari Decision Tree tunggal ke ensemble model (banyak pohon ->lebih kompleks).

Dampak ke bias-variance:

Bias menurun: Model lebih kuat menangkap pola kompleks.

Variance menurun (kalau pakai teknik ensemble dengan baik, seperti averaging di Random Forest).

Relevansi: Karena Decision Tree saya underperforming (R^2 = negatif), moving ke Bagging atau Boosting bisa memperbaiki generalisasi.

2. Selain MSE, dua alternatif loss function untuk regresi:

A. Mean Absolute Error (MAE)

Keunggulan:

Lebih robust terhadap outlier (tidak mengkuadratkan error, jadi outlier tidak mendominasi).

Kelemahan:

Tidak smooth (karena turunan MAE tidak kontinu di nol), agak sulit untuk optimisasi berbasis gradient descent.

Cocok digunakan:

Saat dataset punya banyak outlier.

B. Huber Loss

Keunggulan:

Kombinasi MSE dan MAE: MSE untuk error kecil (smooth optimization), MAE untuk error besar (robust outlier).

Kelemahan:

Perlu memilih hyperparameter delta (threshold transisi MAE–MSE).

Cocok digunakan:

Kalau ingin model robust terhadap outlier tanpa mengorbankan kemampuan optimisasi.

3. Metode untuk mengukur pentingnya fitur tanpa mengetahui nama fitur:

A. Koefisien Regresi (untuk model linear)

Prinsip:

Besarnya koefisien (setelah standardisasi fitur) menunjukkan seberapa besar pengaruh fitur terhadap target.

Keterbatasan:

Hanya akurat kalau tidak ada multikolinearitas.

Tidak menangkap hubungan non-linear.

B. Feature Importance berdasarkan Impurity Reduction (untuk Decision Tree / Random Forest)

Prinsip:

Fitur yang paling banyak menurunkan impurity (seperti Gini, Entropy, MSE) di seluruh tree dianggap paling penting.

Keterbatasan:

Bias ke fitur dengan banyak kategori atau range nilai besar.

Tidak mengukur interaksi fitur secara langsung.

4. Mendesain eksperimen untuk memilih hyperparameter optimal:

Metode: Grid Search atau Random Search dengan Cross-Validation

Langkah-langkah:

Tentukan space hyperparameter (contoh: max_depth untuk Decision Tree dari 3–15).

Gunakan k-fold cross-validation (misal: k=5) untuk menguji setiap kombinasi.

Pilih kombinasi dengan performa rata-rata terbaik di validasi.

Tradeoff:

Komputasi:

Grid Search exhaustive, mahal di waktu -> Random Search lebih hemat.

Stabilitas:

Cross-validation mengurangi ketergantungan ke dataset tertentu (mengurangi variance hasil training).

Generalisasi:

Cross-validation memastikan model tidak hanya bagus di training data tapi juga unseen data.

5. Jika residual plot menunjukkan pola non-linear + heteroskedastisitas, langkah yang akan diambil:

A. Transformasi Data

Coba transformasi target (contoh: log(y), sqrt(y)) untuk mengatasi heteroskedastisitas.

Alasan: Transformasi bisa membuat variance residual lebih stabil.

B. Ubah Model ke Non-linear Model

Contohnya: Polynomial Regression atau pakai model seperti Decision Tree / Ensemble.

Alasan: Model linear tidak cocok untuk pola non-linear -> perlu model yang lebih fleksibel.

C. Gunakan Weighted Least Squares (WLS)

Jika heteroskedastisitas tetap ada, berikan bobot lebih kecil ke data dengan error besar.

Alasan: WLS bisa mengatasi residuals yang tidak homogen.