

1. Jika model Machine Learning menunjukkan AUC-ROC tinggi (0.92) tetapi Presisi sangat rendah (15%) pada dataset tersebut, jelaskan faktor penyebab utama ketidaksesuaian ini! Bagaimana strategi tuning hyperparameter dapat meningkatkan Presisi tanpa mengorbankan AUC-ROC secara signifikan? Mengapa Recall menjadi pertimbangan kritis dalam konteks ini, dan bagaimana hubungannya dengan cost false negative?

**Jawab :**

**Faktor Penyebab Utama:**

Ketidakseimbangan kelas (class imbalance) menjadi faktor utama. Dalam situasi ini, banyaknya data negatif dibanding data positif menyebabkan presisi menjadi rendah walaupun ROC-AUC tinggi karena ROC-AUC sensitif terhadap urutan ranking prediksi, bukan pada threshold prediksi secara eksplisit.

**Strategi Tuning Hyperparameter :**

Menggunakan pendekatan seperti adjusting classification threshold, class-weight balancing, atau cost-sensitive learning dapat meningkatkan presisi tanpa mengorbankan ROC-AUC secara signifikan. seperti `scale_pos_weight` (pada XGBoost) atau `class_weight` (pada Logistic Regression, Random Forest) dapat diatur agar lebih berpihak pada kelas minoritas, sehingga meningkatkan presisi.

**Mengapa Recall Kritis :**

Recall sangat penting dalam kasus sensitif seperti fraud detection atau diagnosis medis karena biaya False Negative jauh lebih tinggi dibanding False Positive. Hubungannya dengan biaya False Negative adalah bahwa jika Recall rendah, banyak kasus positif yang terlewatkan, yang berpotensi menyebabkan kerugian yang signifikan dalam aplikasi bisnis atau kesehatan.

2. Sebuah fitur kategorikal dengan 1000 nilai unik (high-cardinality) digunakan dalam model machine learning. Jelaskan dampaknya terhadap estimasi koefisien dan stabilitas Presisi! Mengapa target encoding berisiko menyebabkan data leakage dalam kasus dataset tersebut, dan alternatif encoding apa yang lebih aman untuk mempertahankan AUC-ROC?

**Jawab :**

**Dampak Terhadap Koefisien dan Stabilitas Presisi :** High-cardinality menyebabkan sparsity pada data yang berimbas pada estimasi koefisien yang tidak stabil (varians tinggi) karena banyak kategori yang memiliki sedikit observasi. Akibatnya, presisi prediksi menjadi tidak stabil karena kategori tertentu tidak mendapat cukup data representatif untuk estimasi yang robust.

**Risiko Data Leakage pada Target Encoding :** Target encoding menggunakan rata-rata nilai target, yang jika tidak hati-hati, akan menyebabkan kebocoran informasi (data leakage), karena nilai target dari validasi atau tes “bocor” ke set pelatihan. Dampaknya, model akan terlihat sangat baik di validasi internal tetapi buruk pada data baru.

**Alternatif Encoding Aman:** CatBoost encoding atau Leave-One-Out encoding dengan cross-validation lebih aman karena menghindari leakage dengan metode validasi yang ketat. Hashing encoding juga aman dan dapat mempertahankan performa AUC-ROC tanpa risiko kebocoran informasi

3. Setelah normalisasi Min-Max, model SVM linear mengalami peningkatan Presisi dari 40% ke 60% tetapi Recall turun 20%. Analisis dampak normalisasi terhadap decision boundary dan margin kelas minoritas! Mengapa scaling yang sama mungkin memiliki efek berlawanan jika diterapkan pada model Gradient Boosting?

**Jawab :**

**Dampak Normalisasi pada Decision Boundary dan Margin (SVM Linear):** Normalisasi min-max membuat skala fitur lebih seragam, sehingga memudahkan SVM menemukan hyperplane yang optimal. Ini meningkatkan presisi karena decision boundary menjadi lebih optimal terhadap kelas minoritas. Tetapi, margin decision boundary yang ketat ini justru mempersempit coverage positif, menurunkan recall.

**Mengapa Efek Bisa Berlawanan pada Gradient Boosting:** Gradient Boosting berbasis pohon tidak sensitif terhadap scaling karena decision tree membagi data berdasarkan nilai absolut fitur, bukan skala relatifnya. Bahkan, scaling dapat mengganggu distribusi asli data yang sudah optimal dalam konteks tree-based models, sehingga hasil bisa menurun atau tidak terpengaruh sama sekali.

4. Eksperimen feature interaction dengan menggabungkan dua fitur melalui perkalian meningkatkan AUC-ROC dari 0.75 ke 0.82. Jelaskan mekanisme matematis di balik peningkatan ini dalam konteks decision boundary non-linear! Mengapa uji statistik seperti chi-square gagal mendeteksi interaksi semacam ini, dan metode domain knowledge apa yang dapat digunakan sebagai alternatif?

**Jawab :**

**Mekanisme Matematis:** Menggabungkan fitur melalui perkalian ( $x_1 \times x_2$ ) menghasilkan fitur baru yang secara matematis memungkinkan terbentuknya non-linear decision boundaries. Ini memperkaya ruang fitur sehingga model linear dapat menemukan pemisah yang sebelumnya tidak bisa ditangkap oleh fitur individual. Secara matematis, perkalian dua fitur menciptakan representasi nonlinearitas seperti kurva parabola atau bidang lengkung, sehingga decision boundary dapat menjadi lebih fleksibel.

**Alasan Uji Statistik Chi-square Gagal:** Chi-square adalah tes statistik untuk independensi antar dua variabel secara kategorikal, bukan mendeteksi interaksi nonlinear antar fitur numerik. Interaksi nonlinear antar fitur numerik seringkali tidak dapat dideteksi oleh metode statistik sederhana karena tidak selalu berhubungan langsung dengan distribusi kategori atau frekuensi nilai absolut.

**Metode Alternatif (Domain Knowledge):** Domain knowledge-driven feature digunakan untuk menentukan interaksi fitur relevan. Misalnya dalam fraud detection, menggabungkan “jumlah transaksi” dengan “waktu transaksi” karena secara domain keduanya berhubungan erat dalam mendeteksi anomali. Alternatif lain adalah

menggunakan metode eksploratif seperti partial dependence plot (PDP) atau SHAP untuk mendeteksi interaksi yang kuat secara visual dan interpretatif.

5. Dalam pipeline preprocessing, penggunaan oversampling sebelum pembagian train-test menyebabkan data leakage dengan AUC-ROC validasi 0.95 tetapi AUC-ROC testing 0.65. Jelaskan mengapa temporal split lebih aman untuk fraud detection, dan bagaimana stratified sampling dapat memperparah masalah ini! Bagaimana desain preprocessing yang benar untuk memastikan evaluasi metrik Presisi/Recall yang realistis?

**Jawab :**

**Mengapa Terjadi Leakage:** Oversampling sebelum pemisahan train-test menyebabkan data artifisial dari kelas minoritas yang sama persis muncul baik di training maupun testing, mengakibatkan model “hafal” pola yang seharusnya tidak diketahui sebelumnya (leakage).

**Mengapa Temporal Split Lebih Aman (Khusus Fraud Detection):** Temporal split memastikan bahwa data training selalu lebih lama atau mendahului data testing, merefleksikan situasi nyata di mana prediksi dilakukan terhadap data yang belum pernah dilihat sebelumnya. Hal ini kritis dalam fraud detection, karena pola fraud sering berubah seiring waktu.

**Mengapa Stratified Sampling Dapat Memperburuk Masalah:** Stratified sampling memastikan proporsi kelas yang sama di training dan testing tetapi dapat menyebabkan leakage jika dilakukan setelah oversampling, karena mencampur data hasil sintesis dengan data asli dalam kedua set tersebut.

**Desain Preprocessing yang Benar:** Melakukan split terlebih dahulu (khususnya temporal split), baru kemudian melakukan oversampling hanya pada training set. Evaluasi menggunakan skenario realistis (time-based cross-validation atau out-of-time validation) untuk memastikan nilai presisi/recall mencerminkan performa yang sebenarnya.