Analisis Model Machine Learning – Klasifikasi Fraud Detection

1. AUC-ROC Tinggi, Presisi Rendah – Analisis dan Strategi

Jika model machine learning menunjukkan AUC-ROC tinggi (misalnya 0.92) tetapi presisi sangat rendah (misalnya 15%), artinya model dapat membedakan antara kelas positif dan negatif dengan baik secara umum, namun sering salah memprediksi kelas positif (fraud) pada data negatif.

Faktor penyebab utama ketidaksesuaian ini antara lain:
- Ketidakseimbangan kelas (class imbalance): Model lebih sering menebak kelas mayoritas (non-fraud).
- Threshold default: Banyak model menggunakan threshold 0.5, yang belum tentu optimal.
- Fokus model pada Recall (karena AUC lebih dipengaruhi oleh TPR dan FPR) bukan Presisi.

Strategi tuning hyperparameter untuk meningkatkan presisi tanpa mengorbankan AUC secara signifikan:
- Ubah threshold prediksi (misalnya 0.7 daripada 0.5) untuk mengurangi false positives.
- Gunakan class_weight='balanced' atau oversampling secara selektif.
- Gunakan precision-recall tradeoff dari Precision-Recall Curve.

Mengapa Recall penting dalam fraud detection:
- Fraud yang tidak terdeteksi (False Negative) sangat mahal secara finansial.
- Recall tinggi memastikan deteksi sebanyak mungkin kasus fraud, meskipun harus mengorbankan sedikit false positive.

Cost False Negative sangat tinggi dalam konteks fraud, karena satu transaksi fraud yang terlewat bisa menyebabkan kerugian besar.

2. High Cardinality Feature dan Encoding Aman

Fitur kategorikal dengan 1000 nilai unik (high-cardinality) dapat menyebabkan:
- Overfitting karena sparsitas dalam one-hot encoding.
- Koefisien estimasi menjadi tidak stabil, terutama pada model linear.
- Presisi menjadi buruk karena model 'menebak' pola dari noise.

Target encoding berisiko menyebabkan data leakage karena:
- Encoding dilakukan dengan informasi target, bisa "bocor" ke data uji jika tidak dilakukan dalam cross-validation.
- Model bisa 'tertipu' oleh encoding target yang overfit pada data train.

Alternatif encoding yang lebih aman:
- Leave-one-out encoding (menghindari nilai target data sendiri).
- Frequency encoding.
- Embedding (misalnya pada model neural network).

Tujuan utama encoding yang aman adalah mempertahankan distribusi yang tidak bocor ke data test, sehingga AUC-ROC tetap realistis.

3. Normalisasi Min-Max pada SVM vs Gradient Boosting

Normalisasi Min-Max meningkatkan presisi SVM linear karena:
- Decision boundary linear SVM sensitif terhadap skala fitur.
- Fitur yang sebelumnya 'dominan' secara skala jadi lebih seimbang.
- Margin antar kelas menjadi lebih optimal.

Penurunan Recall terjadi karena:
- SVM lebih yakin dengan margin sempit, tetapi mengorbankan beberapa kelas minoritas.

Pada Gradient Boosting, normalisasi bisa berdampak sebaliknya karena:
- Gradient Boosting berbasis pohon tidak sensitif terhadap skala.
- Normalisasi bisa menyebabkan fitur numerik yang penting kehilangan kekhasan distribusinya.
- Pohon menjadi kurang mampu membuat split yang baik karena fitur 'disamaratakan'.

4. Feature Interaction dan Non-Linear Boundary

Interaksi fitur (perkalian dua fitur) meningkatkan AUC karena menciptakan variabel baru yang mewakili hubungan non-linear.

Secara matematis:
- Interaksi menciptakan dimensi baru yang memungkinkan model menangkap pola kompleks.
- Decision boundary yang tadinya linear bisa membentuk kurva, elips, atau bentuk kompleks lainnya.

Chi-square test gagal mendeteksi ini karena:
- Mengukur asosiasi dua variabel secara independen, bukan efek gabungan.
- Tidak menguji interaksi non-linear atau kontinyu.

Alternatif dengan domain knowledge:
- Gunakan visualisasi scatter plot per kelas.
- Analisis berdasarkan pemahaman bisnis/fraud.
- Gunakan teknik seperti polynomial features atau interaction terms secara eksplisit.

5. Data Leakage karena Oversampling dan Solusi

Oversampling sebelum split menyebabkan data leakage karena:
- Informasi sintetis dari data test ‘masuk’ ke data train.
- Model belajar dari pola yang seharusnya tidak ada saat inferensi nyata.

AUC-ROC validasi tinggi (0.95) tapi testing rendah (0.65) menandakan overfitting akibat leakage.

Mengapa temporal split lebih aman dalam fraud detection:
- Fraud biasanya berkaitan dengan waktu (time-series).
- Split temporal mencerminkan kondisi nyata (model memprediksi masa depan dari masa lalu).

Stratified sampling bisa memperparah masalah karena:
- Menyamakan distribusi kelas di semua subset, tapi tidak mempertimbangkan urutan waktu.
- Dapat mencampur data fraud dari masa depan ke masa lalu.

Desain preprocessing yang benar:

1. Split data dulu (train-test secara temporal).

2. Lakukan preprocessing (scaling, encoding, sampling) hanya di data train.

3. Terapkan transformasi ke data test menggunakan parameter dari data train.

Hal ini memastikan evaluasi metrik seperti Presisi dan Recall benar-benar mencerminkan performa di dunia nyata.