ANALISA CLASSIFICATION MODEL

---

1. AUC-ROC Tinggi tapi Presisi Rendah

Penyebab utama:

- AUC-ROC mengukur kemampuan model membedakan kelas (ranking score), bukan threshold klasifikasi.
- Presisi sangat sensitif terhadap threshold; meski skor probabilitas benar, jika threshold terlalu rendah, akan banyak false positive.
- Ketidakseimbangan data (class imbalance) bisa menyebabkan prediksi terlalu banyak ke kelas mayoritas → Presisi rendah.

Strategi Tuning:

- **Threshold Tuning:** Sesuaikan decision threshold secara eksplisit untuk mengurangi false positives.
- **Class Weighting:** Menyeimbangkan bobot loss function untuk mendorong model penalti lebih besar terhadap kesalahan pada kelas minoritas.
- **Undersampling/Smarter Sampling:** Untuk mengurangi overfitting terhadap kelas mayoritas.

Recall dan False Negative:

- Recall tinggi berarti sedikit **false negative**, penting pada kasus seperti **fraud detection** atau **penyakit**, di mana **FN sangat mahal**.
- Dalam kasus seperti ini, **Presisi boleh dikorbankan sedikit**, tapi **Recall harus dijaga**.

---

2. Fitur High-Cardinality & Risiko Target Encoding

Dampak High-Cardinality:

- Menyebabkan **sparse feature space** → model sulit menangkap pola umum, overfitting terhadap nilai unik.
- Estimasi koefisien menjadi **tidak stabil** pada model linier.
- Presisi bisa turun karena noise dari nilai kategori langka.

Target Encoding & Data Leakage:

- Menggunakan **mean target label** berdasarkan kategori.

- Jika encoding dilakukan **sebelum split**, model "melihat" target dari data validasi/test → leakage → AUC-ROC palsu.

Alternatif Encoding yang Aman:

- **Leave-One-Out Encoding (dengan cross-validation loop)**.
- **Frequency Encoding** (menghitung frekuensi kategori).
- Gunakan **embedding layer** (misalnya pada model deep learning).
- **CatBoost Encoding** yang inherently menghindari leakage.

---

3. Min-Max Normalisasi pada SVM vs Gradient Boosting

Dampak pada SVM:

- SVM sangat bergantung pada skala fitur → normalisasi mempengaruhi **decision boundary** dan margin.
- Min-Max scaling bisa **memperkuat separabilitas** untuk fitur penting → meningkatkan Presisi.
- Tapi margin bisa bergeser jauh dari minoritas → **Recall menurun**.

Efek Berlawanan di Gradient Boosting:

- Gradient Boosting berbasis pohon → **tidak peka terhadap skala fitur**.
- Normalisasi malah bisa **menghapus informasi** yang berguna dari distribusi asli fitur.
- Bisa membuat model bekerja **lebih buruk**, terutama jika fitur bernilai ordinal atau skewed.

---

4. Feature Interaction dan Decision Boundary Non-Linear

Mekanisme Peningkatan AUC-ROC:

- Perkalian dua fitur bisa menghasilkan **interaksi polinomial** yang menciptakan **decision boundary non-linear**.
- Model linear (misalnya Logistic Regression) bisa menangkap interaksi ini lebih baik dengan fitur gabungan.

Chi-square Gagal Deteksi:

- Uji chi-square hanya mendeteksi **korelasi linier antar fitur dengan label kategori**, bukan interaksi non-linier.
- Interaksi semacam ini bisa muncul hanya saat dua fitur digabung.

Alternatif (berbasis domain knowledge):

- **Expert rules**: berdasarkan pengetahuan lapangan (misalnya, penggabungan "jumlah transaksi" dan "durasi akun" untuk mendeteksi penipuan).
- **PCA untuk interaksi** atau **Decision Tree-based importance analysis**.

---

5. Oversampling sebelum Train-Test Split = Data Leakage

Mengapa Data Leakage Terjadi:

- Oversampling menambahkan data sintetis (misal SMOTE) **berdasarkan seluruh dataset**, termasuk informasi dari test set.
- Model belajar dari data buatan yang **sangat mirip dengan test set** → validasi tinggi palsu (0.95), tapi testing gagal (0.65).

Mengapa Temporal Split Aman:

- Untuk fraud detection (data sekuensial), **future data tidak boleh digunakan untuk memprediksi masa lalu**.
- Temporal split menjaga urutan waktu → menghindari informasi bocor dari masa depan.

Stratified Sampling Bisa Memperparah:

- Jika data berurutan, stratified split bisa **mencampur waktu** dan menyebabkan test set mengandung contoh yang hampir sama dengan train set.

Desain Preprocessing yang Benar:

1. **Split dulu (train-test) berdasarkan waktu.**
2. Lakukan **oversampling hanya di training set.**
3. Validasi menggunakan teknik seperti **TimeSeriesSplit**.
4. Evaluasi metrik **Presisi, Recall, F1** secara adil di test set tanpa augmentasi.
