Kinanti Aria Widaswara
1103213165

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?

Penyebab utama ketidaksesuaian ini:

- AUC-ROC mengukur kemampuan model membedakan kelas secara umum (across all thresholds).
- Presisi bergantung pada threshold klasifikasi default (biasanya 0.5).
- Ketika threshold tidak optimal, model bisa memberikan banyak false positives → Presisi rendah, meskipun ranking model (AUC) tetap bagus.

Strategi tuning untuk meningkatkan Presisi tanpa mengorbankan AUC:

- **Threshold tuning:** Ubah cut-off decision threshold ke nilai lebih tinggi (misal 0.7–0.9) untuk menurunkan false positives.
- **Class weight tuning:** Gunakan class_weight='balanced' atau loss yang menekan false positives.
- **Hyperparameter tuning:** Misalnya, C dan gamma di SVC, max_depth di tree, learning_rate di boosting bisa dikontrol untuk mengurangi overfitting terhadap noise.

Recall sebagai pertimbangan kritis:

- Dalam banyak kasus (misal: **fraud detection, medical diagnosis**), **false negative jauh lebih berisiko** daripada false positive.
- **Recall** tinggi menjamin bahwa sebagian besar kasus positif berhasil ditemukan (mengurangi false negatives).

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?

Dampaknya:

- **Estimasi koefisien** menjadi tidak stabil (terutama untuk model linier) karena jumlah variabel dummy besar.
- Bisa menyebabkan **overfitting** karena noise antar kategori yang terlalu granular.
- Presisi bisa **turun drastis** karena noise mengganggu distribusi label di tiap kategori.

Risiko Target Encoding:

- Jika dilakukan sebelum train-test split, target encoding "membocorkan label" ke fitur → **data leakage**.
- Ini membuat model belajar dari informasi yang tidak tersedia saat prediksi aktual, menaikkan AUC-ROC validasi secara palsu.

Alternatif lebih aman:

- **Leave-One-Out Encoding** atau **Cross-validated Target Encoding** (fit hanya di train folds).
- **Frequency encoding** (mengencode kategori berdasarkan frekuensi muncul).
- **Entity embedding** (dalam neural network) untuk menangani high-cardinality secara efisien.

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?

SVM Linear:

- SVM sangat sensitif terhadap skala karena jarak antar titik menentukan margin.
- MinMax scaling bisa memperbesar margin kelas mayoritas → mempersempit margin kelas minoritas → Recall menurun karena **decision boundary bergeser**.

Gradient Boosting:

- Tidak terpengaruh oleh skala karena pohon keputusan bekerja dengan split berdasarkan nilai, bukan jarak atau gradient eksplisit.
- Justru scaling bisa menghilangkan kontras alami fitur → menurunkan performa pohon.

Kesimpulan:

- **SVM** → butuh scaling untuk bekerja optimal.
- **Tree-based model (GBM, XGBoost)** → sebaiknya tidak diskalakan, atau hanya jika ada alasan eksplisit.

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?

Mekanisme Matematis:

- Interaksi non-linear (misalnya feature_A * feature_B) menciptakan fitur baru yang tidak bisa ditangkap model linier biasa.
- Model jadi bisa membentuk **non-linear decision boundary** yang lebih sesuai dengan pola kompleks di data.

Kenapa Chi-Square Gagal Deteksi?

- Uji chi-square hanya mengukur **asosiasi satu fitur terhadap label**, bukan **kombinasi antar fitur**.
- Interaksi dua fitur bisa tidak signifikan secara individual, tapi sangat kuat secara gabungan.

Alternatif:

- **Domain knowledge**: Misalnya, gabungan "jumlah transaksi x waktu login" lebih relevan untuk fraud detection.
- **PCA/Polynomial Features** untuk eksplorasi interaksi otomatis.
- **SHAP interaction values** untuk eksplorasi interaksi dari model kompleks.

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?

Masalah:

- Oversampling (SMOTE, ADASYN, dll) menambahkan data sintetis berdasarkan keseluruhan distribusi, termasuk test set → informasi dari test bocor ke model.

Solusi:

- Lakukan **split dulu**, lalu **oversampling hanya di data training**. Test set tetap untouched.

Kenapa Temporal Split Aman?

- Untuk kasus seperti fraud, data bersifat time-sensitive. Split berdasarkan waktu mencegah **model belajar dari "masa depan"**, yang tidak realistis dalam deployment nyata.

Stratified sampling bisa memperparah masalah jika:

- Dilakukan sebelum split dan digunakan untuk oversampling → model jadi tahu distribusi test.
- Mengelabui kita dengan metrik validasi yang overestimated.

Desain preprocessing yang benar:

1. Split data terlebih dahulu (train-test atau train-val-test).
2. Lakukan oversampling hanya di training.
3. Gunakan pipeline atau cross-validation yang menjaga urutan ini.
4. Evaluasi menggunakan metrik seperti **Precision/Recall** yang sensitif terhadap imbalance.