

Nama : Faiz Hibatullah  
Kelas : TK-45-G05  
NIM : 1103210172

### Analisa Klasifikasi Model

1. Jika model Machine Learning menunjukkan AUC-ROC tinggi (0.92) tetapi Presisi sangat rendah (15%) pada dataset tersebut, jelaskan factor 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?
2. Sebuah fitur kategorikal dengan 1000 nilai unik (high-cardinality) digunakan dalam 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?
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?
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?
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 realistik?

Jawab:

1. Ketidaksesuaian antara AUC-ROC tinggi (0.92) dan presisi rendah (15%) biasanya terjadi karena imbalance dataset yang ekstrem. Model mampu membedakan kelas dengan baik secara keseluruhan (AUC-ROC bagus), tapi gagal pada kasus positif spesifik. Threshold default 0.5 mungkin tidak optimal untuk dataset ini.

Strategi tuning hyperparameter yang efektif meliputi penyesuaian threshold prediksi, penerapan `class_weight` pada algoritma, penggunaan regularisasi yang tepat (L1/L2), dan teknik resampling selektif. Teknik ensemble seperti bagging juga bisa membantu meningkatkan presisi tanpa menurunkan AUC-ROC secara signifikan.

Recall menjadi pertimbangan kritis karena berkaitan langsung dengan false negative. Dalam domain seperti fraud detection atau diagnosis penyakit, cost false negative (melewatkan kasus positif nyata) jauh lebih tinggi daripada false positive. Penyeimbangan presisi-recall perlu mempertimbangkan cost business aktual dari kedua jenis kesalahan.

2. Fitur kategorikal dengan 1000 nilai unik akan menghasilkan matriks sangat sparse saat one-hot encoding, menyebabkan curse of dimensionality. Ini berdampak pada estimasi

koefisien yang tidak stabil dan presisi rendah karena overfitting, terutama jika beberapa kategori jarang muncul dalam dataset.

Target encoding berisiko menyebabkan data leakage karena mengganti kategori dengan rata-rata nilai target, sehingga informasi label testing secara tidak langsung masuk ke proses training. Ini menghasilkan overestimasi performa model saat validasi.

Alternatif encoding yang lebih aman meliputi leave-one-out encoding (mengurangi leakage dengan mengecualikan sampel saat ini), hash encoding (mengelompokkan kategori mirip), atau Weight of Evidence encoding dengan proper cross-validation. Teknik regularisasi kategorikal juga dapat membantu mempertahankan AUC-ROC sambil mengurangi overfitting.

3. Normalisasi Min-Max meningkatkan presisi SVM linear namun menurunkan recall karena transformasi ini mengubah jarak relatif antar titik data. Normalisasi mempengaruhi margin SVM dan lokasi decision boundary, membuat beberapa titik data kelas minoritas yang sebelumnya terklasifikasi benar menjadi salah.

Secara matematis, SVM sangat sensitif terhadap skala fitur karena algoritmanya berbasis jarak. Normalisasi membuat semua fitur berkontribusi sama terhadap perhitungan jarak, mengubah geometri ruang fitur secara signifikan.

Gradient Boosting berbasis tree relatif kebal terhadap transformasi monoton seperti Min-Max scaling karena pemisahan pohon keputusan berdasarkan urutan/ranking nilai, bukan pada nilai absolut. Transformasi yang sama kemungkinan memiliki dampak minimal atau bahkan berlawanan pada model Gradient Boosting.

4. Peningkatan AUC-ROC dari 0.75 ke 0.82 melalui feature interaction (perkalian dua fitur) terjadi karena menciptakan representasi non-linear dari data. Secara matematis, perkalian dua fitur ( $X_1 \times X_2$ ) memperkenalkan komponen kuadratik ke decision boundary yang sebelumnya linear.

Jika model asli hanya dapat membuat pemisahan linear  $f(x) = w_1X_1 + w_2X_2 + b$ , dengan feature interaction model dapat membentuk  $f(x) = w_1X_1 + w_2X_2 + w_3(X_1 \times X_2) + b$ , menghasilkan decision boundary kurva/non-linear yang lebih fleksibel menangkap pola kompleks.

Uji statistik seperti chi-square gagal mendeteksi interaksi ini karena dirancang untuk menilai dependensi linear atau nominal sederhana, bukan interaksi kompleks. Alternatif berbasis domain knowledge meliputi partial dependence plots, analisis SHAP interaction values, atau pendekatan berbasis fisika/domain yang mempertimbangkan mekanisme kausal dalam data.

5. Oversampling sebelum pembagian train-test menyebabkan data leakage karena kasus yang digandakan bisa muncul di kedua set, membuat model "menghafal" contoh spesifik alih-alih generalisasi pola. Ini menjelaskan kesenjangan besar antara AUC-ROC validasi (0.95) dan testing (0.65).

Temporal split lebih aman untuk fraud detection karena menghormati urutan kronologis data, mencegah model "melihat masa depan". Fraud patterns berevolusi seiring waktu, dan split berdasarkan waktu lebih realistis dalam mengevaluasi kemampuan prediktif model terhadap pola baru.

Stratified sampling dapat memperparah masalah karena memaksa distribusi kelas yang sama antara training dan testing, padahal distribusi fraud sesungguhnya bisa berubah seiring waktu, menghasilkan evaluasi yang terlalu optimistis.

Pipeline preprocessing yang benar: (1) Split dulu data (sebaiknya temporal untuk fraud detection), (2) Lakukan normalisasi/transformasi fitur menggunakan statistik dari training set saja, (3) Terapkan oversampling/undersampling pada training set saja, (4) Evaluasi pada test set asli (tanpa oversampling) untuk mendapatkan estimasi performa realistis.